COMPUTABLE Π_2 SCOTT SENTENCES

JULIA KNIGHT, KAREN LANGE, AND CHARLES MCCOY

Abstract. By a result of Scott [16], each countable structure for a countable language L is described up to isomorphism by an $L_{\omega_1\omega}$-sentence, called a *Scott sentence*. We consider structures that are countably infinite. By a result of A. Miller [11], no such structure has a Σ_2 Scott sentence, so having a Π_2 Scott sentence is as simple as possible. A result of Montalbán [12] yields a nice characterization of the structures that (for a fixed countable language) have a Π_2 Scott sentence. *Computable infinitary formulas* involve c.e. disjunctions and conjunctions, so they are in a sense comprehensible. Therefore, we set out to characterize the structures that (for a fixed computable language) have a computable Π_2 Scott sentence. We found some examples and some partial results. However, it turns out that (for most languages) there is no nice characterization of the class. The index set is Π^1_1-complete.

§1. Introduction. Infinitary $L_{\omega_1\omega}$ formulas allow countable conjunctions and disjunctions, but only finite strings of quantifiers. The following result of Scott [14] illustrates the expressive power of $L_{\omega_1\omega}$. (See [4] for more background.)

THEOREM 1.1 (Scott). *For any countable structure $\mathcal{A}$ in a countable language L, there is an $L_{\omega_1\omega}$-sentence φ whose countable models are just the isomorphic copies of $\mathcal{A}$.*

We call φ a *Scott sentence* for $\mathcal{A}$. For a computable language L, *computable* infinitary formulas are the formulas of $L_{\omega_1\omega}$ in which the disjunctions and conjunctions are over c.e. sets. These formulas, despite being infinitely long, seem comprehensible. Many of the $L_{\omega_1\omega}$-formulas that describe interesting properties of structures, and important relations on elements, turn out to be computable infinitary.

For $L_{\omega_1\omega}$, there is no prenex normal form. We cannot always bring the quantifiers outside. However, we can bring the negations inside. This yields a different normal form. We classify formulas in this normal form based on the number of alternations of $\bigvee\exists$ and $\bigwedge\forall$. In particular, a Π_2 formula has the form $\bigwedge_i(\forall\bar{u}_i)\bigvee_j(\exists\bar{v}_{i,j})\alpha_{i,j}(\bar{u}_i,\bar{v}_{i,j})$, where $\alpha_{i,j}(\bar{u}_i,\bar{v}_{i,j})$ is quantifier-free. In [10], A. Miller showed that for relational languages, no infinite structure can have a Σ_2 Scott sentence. Harrison-Trainor [2] showed that the same is true for an arbitrary language.

Montalbán [11] characterized the structures with a $\Pi_{\alpha+1}$ Scott sentence. He found quite a few different-looking conditions. We give just two.

Received February 14, 2025.

2020 *Mathematics Subject Classification.* Primary 03C57, Secondary 03C75.

Key words and phrases. Scott sentences, existentially atomic models.

© The Author(s), 2026. Published by Cambridge University Press on behalf of The Association for Symbolic Logic. This is an Open Access article, distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives licence (https://creativecommons.org/licenses/by-nc-nd/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is unaltered and is properly cited. The written permission of Cambridge University Press must be obtained for commercial re-use or in order to create a derivative work.

0022-4812/00/0000-0000
DOI:10.1017/jsl.2026.10186

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

THEOREM 1.2 (Montalbán). *Let α be a countable ordinal, and let L be a countable language. For a countable L-structure $\mathcal{A}$, the following are equivalent*:

1. *$\mathcal{A}$ has a $\Pi_{\alpha+1}$ Scott sentence*;
2. *for each tuple $\bar{a}$, the orbit is defined by a Σ_α formula*;
3. *for each tuple $\bar{a}$, there is a Σ_α formula that implies (in $\mathcal{A}$) all Π_α formulas true of $\bar{a}$.*

For an infinite structure, the simplest form for a Scott sentence is Π_2. Montalbán's Theorem (with $\alpha = 1$) yields the following.

COROLLARY 1.3. *For a countable structure $\mathcal{A}$, the following are equivalent*:

1. *$\mathcal{A}$ has a Π_2 Scott sentence*;
2. *for each tuple $\bar{a}$, the orbit is defined by an existential formula*[1] ;
3. *for each tuple $\bar{a}$, there is an existential formula that implies the universal type of $\bar{a}$.*

REMARK. A further condition, obviously equivalent to Conditions 2 and 3, says that for each tuple, some existential formula implies the elementary first-order type.

In [3], there is an example of a computable structure $\mathcal{A}$ with a Π_2 Scott sentence but not one that is computable Π_2. The structure is "unary." *Unary* structures will be important for our main results. The language of these structures consists just of unary relation symbols, and the structures satisfy some special axioms (see Section 2.4). Most of this article concerns the following problem, which was stated in [3].

PROBLEM. *When does a countable structure have a computable Π_2 Scott sentence*?

In the remainder of Section 1, we give some complexity results for the class of structures with a $\Pi_{\alpha+1}$ Scott sentence, not necessarily computable. We also give a condition that is necessary and sufficient for a structure to have a computable Π_2 Scott sentence. This condition is not very satisfying—it does not yield a computable infinitary sentence characterizing the class. In Section 2, we give a few examples of structures with and without computable Π_2 Scott sentences. We also provide some background on unary structures. In Section 3, we give some conditions sufficient to guarantee that a structure has a computable Π_2 Scott sentence, together with examples showing that these conditions are not necessary.

In Section 4, we give a condition that is necessary and sufficient for a unary structure to have a computable Π_2 Scott sentence. This condition is also not very satisfying, although we will use it in Section 5, where we show that, for the class KU of unary structures with a computable Π_2 Scott sentence, the index set is Π^1_1-complete. This implies that there is no computable infinitary sentence characterizing the class. Similarly, if L is a "rich"[2] computable language, then for the class K_L of L-structures with a computable Π_2 Scott sentence, the index set is Π^1_1-complete. Alvir, Csima, and Harrison-Trainor [1] have also shown this, independently, and at approximately the same time.

[1] Montalbán [12] refers to the structures that satisfy (2) as *existentially atomic*.
[2] A *rich* language is one that has at least one relation symbol of arity at least 2.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

1.1. Complexity consequences of Montalbán's Theorem. The lemma below says how complicated it is to define the standard back-and-forth relations $\leq_\alpha$. (For a discussion of these relations, see [4].)

LEMMA 1.4. *Let L be a computable language. For each computable ordinal $\alpha \geq 1$, and each pair of tuples of variables $\bar{x}, \bar{y}$ of the same length, there is a computable $\Pi_{2\alpha}$ formula that defines the relation $\bar{x} \leq_\alpha \bar{y}$ in all L-structures.*

PROOF. For $\alpha = 1$, the formula is the conjunction over finitary existential formulas $d(\bar{x})$ of formulas saying that $d(\bar{y}) \to d(\bar{x})$. We can take the conjuncts to be finitary $\forall\exists$, so that the conjunction is computable Π_2. For $\alpha > 1$, the formula is the conjunction over $1 \leq \beta < \alpha$, and further tuples of variables $\bar{u}, \bar{v}$ of the same length, of computable $\Pi_{2\beta}$ formulas saying $(\forall \bar{v})(\exists \bar{u})\bar{y}, \bar{v} \leq_\beta \bar{x}, \bar{u}$. ⊣

Using this lemma, we characterize the structures that have a $\Pi_{\alpha+1}$ Scott sentence.

THEOREM 1.5. *Let L be a countable language. For each countable ordinal α, there is a $\Pi_{2\alpha+2}$-sentence characterizing the class of L-structures that have a $\Pi_{\alpha+1}$ Scott sentence. For a computable language L and a computable ordinal α, the sentence is computable $\Pi_{2\alpha+2}$.*

PROOF. By Theorem 1.2, the structure $\mathcal{A}$ has a $\Pi_{\alpha+1}$ Scott sentence, if and only if for each tuple $\bar{a}$, there is a Σ_α formula that implies (in $\mathcal{A}$) all Π_α formulas true of $\bar{a}$. Then the class is characterized by the computable $\Pi_{2\alpha+2}$-sentence saying that for all tuples $\bar{x}$, there exist some $\beta < \alpha$ and some further tuple $\bar{u}$ such that for all $\bar{y}$ and $\bar{v}$, if $\bar{x}, \bar{u} \leq_\beta \bar{y}, \bar{v}$, then $\bar{x} \leq_\alpha \bar{y}$. ⊣

COROLLARY 1.6. *For a computable language L and a computable ordinal α, the index set for the class of L-structures with a $\Pi_{\alpha+1}$ Scott sentence is $\Pi^0_{2\alpha+2}$. In particular, the index set for the class of L-structures with a Π_2 Scott sentence is Π^0_4.*

Corollary 1.6 is due to Gonzalez [7]. Theorem 1.5 was certainly known to Gonzalez and Rossegger, who, in [8], showed that for a rich computable language L and a computable limit ordinal α, the index set for the class of L-structures with a computable Π_α Scott sentence is Π^1_1-complete.

Toward our main problem, the theorem below gives a necessary and sufficient condition for a structure (possibly in a rich language) to have a computable Π_2 Scott sentence.

THEOREM 1.7. *Let $\mathcal{A}$ be a countable structure for a computable language L. Then $\mathcal{A}$ has a computable Π_2 Scott sentence iff there is a uniformly c.e. family $\Gamma_i(\bar{u}_i)$ of sets of universal formulas such that $\mathcal{A}$ omits all $\Gamma_i(\bar{u}_i)$, and every countable L-structure $\mathcal{B}$ not isomorphic to $\mathcal{A}$ realizes some Γ_i.*

The proof involves just writing what it means to have a computable Π_2 Scott sentence. The characterization does not tell us what the structures look like, and it does not yield anything like Theorem 1.5. Our result in Section 5 explains why, in fact, no nice characterization is possible.

§2. First examples. In this section, we consider some examples of structures with and without a computable Π_2 Scott sentence.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

2.1. Linear orders. The following results are known (see the paper of Gonzalez and Rossegger [8]). Both are easy to prove.

PROPOSITION 2.1. *The usual ordering on the rationals has a computable* Π_2*, even finitary, Scott sentence.*

PROPOSITION 2.2. *The usual ordering on the rationals is, up to isomorphism, the only countably infinite linear ordering with a* Π_2 *Scott sentence.*

2.2. Torsion-free Abelian groups. For groups, the usual language has one binary operation symbol (for the group operation), one unary operation symbol (for the inverse), and one constant (for the identity). We show that among torsion-free Abelian groups of rank 1, the only one with a computable Π_2 Scott sentence is the additive group of rationals $\mathbb{Q}$. The fact that $\mathbb{Q}$ has a computable Π_2 Scott sentence is surely known. We give a proof to illustrate what these sentences can express.

PROPOSITION 2.3. *The group* $\mathbb{Q}$ *has a computable* Π_2 *Scott sentence.*

PROOF. For a computable Π_2 Scott sentence, we may take the conjunction of the finitary (universal) axioms for Abelian groups, a computable Π_1 sentence saying that the group is torsion-free, and computable Π_2 sentences saying that the group is divisible and has rank 1. At some x is not 0 and for all pairs x, y, there are integers m, n, not both zero, such that $mx + ny = 0$. To say that the group is divisible, we say that every element is divisible by all non-zero n. ⊣

PROPOSITION 2.4. *No rank* 1 *torsion-free Abelian group apart from* $\mathbb{Q}$ *has a* Π_2 *Scott sentence.*

PROOF. Let G be a subgroup of $\mathbb{Q}$. By a result of Baer [6], the isomorphism type of G is determined by the divisibility type of a non-zero element a, where this is given by the collection of prime powers that divide a. If, for a particular prime p, a is divisible by p^n for all n, then the same is true for all elements. If G is not isomorphic to $\mathbb{Q}$, then for any non-zero element a, there is some prime p and some n such that p^n does not divide a. Replacing a by a multiple, if necessary, we may suppose that p itself does not divide a. The function $x \to px$ takes G to a copy pG that is a proper substructure of G. Any existential formula true of a in G is true of $b = pa$ in pG. It is also true of b in G. However, no automorphism of G takes a to b. Since the orbit of a is not defined by an existential formula, G cannot have a Π_2 Scott sentence, by Corollary 1.3. ⊣

2.3. Subfields of the algebraic numbers. For fields, the standard language has two binary operation symbols, one for addition and one for multiplication, and two constants, for the additive and multiplicative identities.

PROPOSITION 2.5. *If* F *is a subfield of the algebraic numbers, then it has a* Π_2 *Scott sentence. Moreover, if* F *is computable, then we may take the Scott sentence to be computable* Π_2*.*

PROOF. Let S be the set of polynomials $p(x)$ over $\mathbb{Q}$ (or $\mathbb{Z}$) with roots in F. Then F has a Scott sentence saying that the structure is a field of characteristic 0 in which

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

every element is a root of some polynomial in S and all polynomials in S have roots. This Scott sentence is Π_2. If F is computable, then S is c.e., and the Scott sentence is computable Π_2. ⊣

2.4. Unary structures. Let L be the language consisting of unary relation symbols U_n, for $n \in \omega$. Below, we define the class of unary structures. For $\sigma \in 2^{<\omega}$, we write $\sigma(x)$ for the formula $\bigwedge_{\sigma(n)=1} U_n(x)$ & $\bigwedge_{\sigma(n)=0} \neg U_n(x)$.

DEFINITION 1. A *unary* structure is an L-structure $\mathcal{A}$ such that for all $\sigma \in 2^{<\omega}$, the formula $\sigma(x)$ is satisfied in $\mathcal{A}$ either by infinitely many elements or by none.

Axioms. The class of unary structures is axiomatized by elementary first-order sentences saying, for each $\sigma \in 2^{<\omega}$ and $n \geq 1$,

$$(\exists x)\sigma(x) \rightarrow (\exists^{\geq n} x)\sigma(x).$$

The axioms above are $\forall\exists$-sentences. The conjunction is a computable Π_2 sentence β. In this section, we give some background on completions of this set of axioms—theories of unary structures. We connect these theories to sub-trees of $2^{<\omega}$. We then describe the example from [3] of a unary structure that has a Π_2 Scott sentence but not one that is computable Π_2.

2.4.1. Theories of unary structures. In unary structures, the quantifier-free 1-types in variable x just say $U_n x$ or $\neg U_n x$ for $n \in \omega$. The isomorphism type of a unary structure $\mathcal{A}$ is determined by the number of elements realizing the different quantifier-free 1-types. The elementary first-order theory $Th(\mathcal{A})$ is determined just by the formulas $\sigma(x)$ that are satisfied.

PROPOSITION 2.6. *For a unary structure $\mathcal{A}$, $Th(\mathcal{A})$ is generated by the $\exists \cup \forall$ part. Moreover, the theory admits elimination of quantifiers.*

PROOF. We consider the L_s-reduct $\mathcal{A}^s$ of $\mathcal{A}$, for $s \in \omega$. For each s, $Th(\mathcal{A}^s)$ is $\aleph_0$-categorical, generated by the true sentences saying for each $\sigma \in 2^s$, how many elements satisfy $\sigma(x)$. There is either a single universal sentence $(\forall x)\neg\sigma(x)$ or an infinite family of existential sentences, one for each n, saying $(\exists^{\geq n} x)\sigma(x)$. Note that either possibility implies the sentences $(\exists x)\sigma(x) \rightarrow (\exists^{\geq n} x\sigma(x)$, which were among the axioms for unary structures. We have seen that for each s, $Th(\mathcal{A}^s)$ is generated by its $\exists \cup \forall$ part. Hence, the same is true for $Th(\mathcal{A})$.

Each elementary first-order formula $\varphi(\bar{x})$ in the language L uses only symbols from some finite language L_s. In $\mathcal{A}^s$, the tuples satisfying $\varphi(\bar{x})$ satisfy one of finitely many quantifier-free types. We have a quantifier-free formula $\varphi^*(\bar{x})$ equivalent to $\varphi(\bar{x})$ in $\mathcal{A}^s$, and therefore, in $\mathcal{A}$. ⊣

Let T be a completion of the theory of unary structures. Each model has infinitely many realizations of every isolated quantifier-free 1-type. The models differ in the number of realizations of non-isolated types.

2.4.2. Scott sentences for unary structures. By Theorem 1.5, it is, in general, computable Π_4 to say that a structure has a Π_2 Scott sentence. For unary structures, we save a quantifier.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

PROPOSITION 2.7. *The class of unary structures with a* Π_2 *Scott sentence is characterized by a computable* Π_3 *sentence.*

PROOF. Let φ be the conjunction of the computable Π_2 sentence β characterizing the unary structures and the computable Π_3 sentence saying that for all x, there is some $\sigma \in 2^{<\omega}$ such that x satisfies

$$\sigma(x) \,\&\, (\forall y)[\sigma(y) \to \bigwedge_n (U_n y \leftrightarrow U_n x)].$$

We show that $\mathcal{A}$ is a unary structure with a Π_2 Scott sentence iff $\mathcal{A} \models \varphi$. First, suppose $\mathcal{A}$ is a unary structure with a Π_2 Scott sentence. By Montalbán's Theorem, the orbit of each element is defined by an existential formula. By elimination of quantifiers, we may take the existential formula to be quantifier-free, of the form $\sigma(x)$ for some $\sigma \in 2^{<\omega}$. Thus, $\mathcal{A} \models \varphi$. Now, suppose $\mathcal{A} \models \varphi$. Then $\mathcal{A}$ is a unary structure. Moreover, for each tuple $\bar{a}$, we get a quantifier-free formula $d(\bar{x})$ that defines the orbit—the formula has conjuncts saying that $x_i = x_j$ or $x_i \neq x_j$ to match the equality relation on $\bar{a}$, and further conjuncts $\sigma_i(x_i)$, one for each i, where $\sigma_i(x_i)$ defines the orbit of a_i. By Montalbán's Theorem, $\mathcal{A}$ has a Π_2 Scott sentence. ⊣

COROLLARY 2.8. *The index set for the class of unary structures with a* Π_2 *Scott sentence is* Π^0_3.

2.4.3. Connection to trees. Given a unary structure $\mathcal{A}$, we get a unique tree $T \subseteq 2^{<\omega}$ such that $\sigma \in T$ if and only if some element of $\mathcal{A}$ satisfies $\sigma(x)$. The tree T has no dead ends. Each element a of $\mathcal{A}$ represents a unique path p through T, where for $\sigma \in 2^{<\omega}$, $\mathcal{A} \models \sigma(a)$ iff $\sigma \subseteq p$. The theory of $\mathcal{A}$ depends just on the set of nodes in the tree. The model itself depends on the number of elements representing the different (non-isolated) paths. For any tree $T \subseteq 2^{<\omega}$ with no dead ends, we get a complete theory Th_T generated by the axioms, for $\sigma \in 2^{<\omega}$, saying $(\forall x)\neg\sigma(x)$ if $\sigma \notin T$, and $(\exists^{\geq n} x)\sigma(x)$, for all n, if $\sigma \in T$. The models of Th_T are the unary structures $\mathcal{A}$ such that $Th(\mathcal{A}) = Th_T$. Note that there always exist such models—choose a countable dense set $\mathbb{D}$ of paths through T, and let $\mathcal{A}$ have infinitely many elements representing each path in $\mathbb{D}$.

PROPOSITION 2.9. *Let* $T \subseteq 2^{<\omega}$ *be a tree with no dead ends. Then* Th_T *is decidable iff* T *is computable.*

PROOF. First, suppose that T is computable. We have seen that the theory Th_T is complete, generated by the axioms given in the paragraph above. Since T is computable, the set of axioms is computable. Therefore, Th_T is decidable. Now, suppose that Th_T is decidable. For $\sigma \in 2^{<\omega}$, $\sigma \in T$ iff Th_T contains the sentence $(\exists x)\sigma(x)$. Therefore, T is computable. ⊣

REMARK. Let $T \subseteq 2^{<\omega}$ be a tree with no dead ends. Then Th_T has a model with a Π_2 Scott sentence iff the isolated paths are dense in T.

2.4.4. Example recalled. We recall the example from [3].

THEOREM 2.10. *There is a computable unary structure* $\mathcal{A}$ *with a* Π_2 *Scott sentence but no computable* Π_2 *Scott sentence.*

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

The proof uses ideas of Badaev [5] and Selivanov [15]. The first step is to construct a computable tree $T \subseteq 2^{<\omega}$ with no dead ends and with just one non-isolated path p, which is not computable. The theory Th_T is decidable. The structure $\mathcal{A}$ is a computable model of Th_T. It is atomic, since no element can represent the non-computable path p. In [3], there is a direct argument that $\mathcal{A}$ has no computable Π_2 Scott sentence.

§3. Partial results. In this section, we give some conditions that imply the existence of a computable Π_2 Scott sentence, together with examples showing that the conditions are not necessary. The result below is given in [3].

PROPOSITION 3.1. *Let $\mathcal{A}$ be a computable structure, and suppose there is a c.e. family Φ of $\exists$-formulas such that each tuple in $\mathcal{A}$ satisfies some $d(\bar{x}) \in \Phi$, and any two tuples satisfying the same formula in Φ are automorphic.*[3] *Then $\mathcal{A}$ has a computable Π_2 Scott sentence.*

Under the hypothesis of Proposition 3.1, there is a computable function f that takes each tuple $\bar{a}$ to an $\exists$-formula defining the orbit. There are computable structures, with a computable Π_2 Scott sentence, but with no such function.

PROPOSITION 3.2. *There is a computable graph $\mathcal{A}$ with a computable Π_2 Scott sentence, such that no Δ^0_2 function assigns to each tuple an existential formula that defines the orbit.*

PROOF. We build $\mathcal{A}$ to satisfy the requirements R_e for all e.

R^{e}: φ_e^K does not assign to each tuple $\bar{a}$ an existential formula defining the orbit.

To satisfy R_e, we designate two elements a_e, b_e—these will represent different connected components. We make sure that if $\varphi_e^K(a_e) = d$, where d is an existential formula true of a_e, then d is also true of b_e but the two are not automorphic. To start off, we make a_e and b_e each part of a cycle of size p_e (different values for different e). If we guess that $\varphi_e^K(a_e) \downarrow = d$, where d is an existential formula true of a_e, and also of b_e, then we extend so that a_e, b_e are not automorphic, making a_e part of a cycle of length r and making b_e part of a cycle of size $s \neq r$ (choosing r and s larger than the stage and the size of any other cycles already in the graph).

If, later, we guess that $\varphi_e^K(a_e) \downarrow = d'$, where d' is an existential formula true of a_e, we first extend so that the connected components of a_e and b_e match—d' is then true of both. Then we extend further, making a_e part of a cycle of length r' and b_e part of a cycle of length $s' \neq r'$ as before. We continue in this way. For each new guess that $\varphi_e^K(a_e) \downarrow = d_{e,n}$, where this is an existential formula true of a_e, we first extend the connected components of a_e and b_e to make them match, so that $d_{e,n}$ is also true of b_e. We then make a_e part of a new cycle of length $r_{e,n}$ and we make b_e part of a cycle of length $s_{e,n} \neq r_{e,n}$.

If $\varphi_e^K(a_e)$ converges to d, where d is an existential formula true of a_e, then after finitely many steps, we will computably know the part of K and K^c used in the computation. In this case d is true of both a_e and b_e, but the two are not in the same orbit. The connected components of a_e and b_e are finite and not isomorphic.

[3] The set Φ is a c.e. *Scott family* with no parameters.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

If $\varphi_e^K(a_e)$ actually diverges, but infinitely often we guess that it converges to an existential formula true of a_e, then a_e and b_e are in the same orbit, which is defined by the formula describing the original cycle of size p_e and one extra cycle that each is a part of. The connected components of a_e and b_e are both infinite and isomorphic.

To produce a computable Π_2 Scott sentence for $\mathcal{A}$, we first consider the set S of all existential sentences true in $\mathcal{A}$ and the set T of all universal sentences true in $\mathcal{A}$. Since $\mathcal{A}$ is computable, the set S is c.e., and the set T is co-c.e. Let $\{\psi_n : n \in \omega\}$ be a computable list of all finitary universal sentences in the language of graphs. Then by a standard technique, employed, for instance, in [4], there is a sequence of infinitary computable Π_1 sentences τ_n, each built entirely from the logical constants for truth and falsity, so that τ_n is logically equivalent to truth iff $\psi_n \in T$; and τ_n is logically equivalent to falsity iff $\psi_n \notin T$. Therefore, the conjunction $\left(\bigwedge_{\varphi \in S} \varphi\right) \wedge \left(\bigwedge_n (\tau_n \to \psi_n)\right)$ is a computable Π_2 sentence logically equivalent to the conjunction of the finitary existential and universal sentences true in $\mathcal{A}$.

Next, consider the computable Π_2 sentence γ stating that every finite tuple $\bar{x}$ that forms a finite linear chain of distinct elements is part of one or two cycles; i.e., for each such $\bar{x}$, either there is $\bar{y}$ so that $\bar{x}\bar{y}$ forms a cycle; or there are distinct $x, \bar{x}', \bar{x}'', \bar{y}$, and $\bar{z}$ so that $\bar{x} = \bar{x}' x \bar{x}''$ and $x\bar{x}'\bar{y}$ forms a cycle, and $x\bar{x}''\bar{z}$ forms a cycle. Next, consider the computable Π_2 sentence ρ saying that for each x, there exists y such that for some e, y lies on a cycle of length p_e and there is a finite path from x to y.

Let $\varphi = \left[\left(\bigwedge_{\varphi \in S} \varphi\right) \wedge \left(\bigwedge_n (\tau_n \to \psi_n)\right)\right] \wedge \gamma \wedge \rho$. Clearly, $\mathcal{A} \models \varphi$. Now consider a countable graph $\mathcal{B}$ so that $\mathcal{B} \models \varphi$. The conjuncts γ and ρ imply that each connected component of $\mathcal{A}$ and $\mathcal{B}$ consists entirely of finite cycles, one of which is a cycle of length p_e for some e. The fact that $\mathcal{A}$ and $\mathcal{B}$ satisfy the same existential and universal sentences implies a number of basic facts. First, each connected component contains exactly one element that is a part of multiple cycles. Moreover, since γ says that each finite chain is itself part of a cycle, each connected component actually contains one element that is a "center," i.e., part of every cycle in that component. Second, for each e, there are exactly two cycles of length p_e, and these are not connected to each other; and for $e \neq e'$, a cycle of length p_e is not connected to a cycle of length $p_{e'}$.

By the construction of $\mathcal{A}$, for each e, consider the two connected components of $\mathcal{A}$ where each contains a cycle of length p_e; either they are isomorphic, or each contains a cycle of a length that the other does not. Because $\mathcal{B}$ satisfies the same existential and universal sentences that $\mathcal{A}$ does, the two connected components of $\mathcal{B}$ that contain a cycle of length p_e must look exactly the same as those in $\mathcal{A}$. Because ρ implies that $\mathcal{A}$ and $\mathcal{B}$ consist entirely of these kinds of connected components, we conclude $\mathcal{A} \cong \mathcal{B}$. ⊣

PROPOSITION 3.3. *For a computable structure $\mathcal{A}$ with a Π_2 Scott sentence, there is always a Δ_3^0 function f taking each tuple $\bar{a}$ to an $\exists$-formula that defines the orbit.*

PROOF. To say of a given existential formula $d(\bar{x})$ that it defines an orbit, it is enough to say that it is satisfied, and for all $\bar{y}$ that satisfy $d(\bar{y})$, the $\forall$-formulas true of $\bar{x}$ are true of $\bar{y}$. This is computable Π_2, uniformly in $d(\bar{x})$. From this, it is clear that we have a Δ_3^0 function taking each $\bar{a}$ to the first orbit-defining $\exists$-formula $d(\bar{x})$ that is true of $\bar{a}$. ⊣

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

Our results so far show that, among structures with Π_2 Scott sentences, there is little connection between having a computable Π_2 Scott sentence and the computational difficulty of finding, the orbit-defining formulas (beyond the constraints of Proposition 3.3). The graph in Proposition 3.2 has a computable Π_2 Scott sentence, but there is no Δ^0_2 function taking each tuple to an $\exists$-formula that defines the orbit. On the other hand, the structure in Theorem 2.10 lacks a computable Π_2 Scott sentence, although it *does* have a Δ^0_2 function taking each tuple to a quantifier-free formula that defines the orbit.

In the graph from Proposition 3.2, where we have a computable Π_2 Scott sentence, we also have a computable procedure that assigns to each tuple $\bar{a}$ an existential formula $\varphi_{\bar{a}}(\bar{x})$ (satisfied by $\bar{a}$), where the tuples satisfying this formula belong to one of only finitely many orbits. The structure in Theorem 2.10, with no computable Π_2 Scott sentence, does not have this property. These examples suggested the following result.

PROPOSITION 3.4. *Assume that $\mathcal{A}$ is a structure with the following properties*:

1. *The $\forall\exists$-theory of $\mathcal{A}$, call it S, is Σ^0_2.*
2. *Every tuple $\overline{a} \in \mathcal{A}$ has an orbit in $\mathcal{A}$ defined by a finitary existential formula.*
3. *There is a computable sequence of finitary existential formulas $(\varphi_i)_{i\in\omega}$ such that*:
   - *for every $\overline{a} \in \mathcal{A}$, there is a formula $\varphi_i(\overline{x})$ with $\mathcal{A} \models \varphi_i(\overline{a})$;*
   - *for each $i \in \omega$, there is a finite list of orbit-defining existential formulas $\gamma_1(\overline{x}), \ldots, \gamma_{k_i}(\overline{x})$ with $\mathcal{A} \models (\forall\overline{x})[\varphi_i(\overline{x}) \to (\gamma_1(\overline{x}) \vee \cdots \vee \gamma_{k_i}(\overline{x}))]$.*

Then $\mathcal{A}$ has a computable Π_2 Scott sentence.

The next lemma will be used to prove Proposition 3.4. It will be used again in Section 5. Before stating the lemma, we give a definition and some notation.

DEFINITION 2. We say that an existential formula $d(\bar{x})$ *splits* in a structure $\mathcal{A}$ if there exist tuples $\bar{a}, \bar{b}$ and an existential formula $d'(\bar{x})$ such that $\bar{a}, \bar{b}$ both satisfy $d(\bar{x})$, but just one of the two satisfies $d'(\bar{x})$.

NOTATION. *For a structure $\mathcal{A}$, $NS(\mathcal{A})$ is the set of existential formulas $d(\bar{x})$ that are satisfied in $\mathcal{A}$ and do not split in $\mathcal{A}$.*

LEMMA 3.5. *Let $\mathcal{A}, \mathcal{B}$ be structures for the same computable language, each with a Π_2 Scott sentence. If $NS(\mathcal{A}) = NS(\mathcal{B})$, then $\mathcal{A} \cong \mathcal{B}$.*

PROOF. We write NS for the common set $NS(\mathcal{A}) = NS(\mathcal{B})$. We may suppose that each formula $d(\bar{x}) \in NS$ implies $\pm\alpha(\bar{x})$ for α among the first n atomic formulas in variables in the n-tuple $\bar{x}$—we drop from NS, the formulas that do not have this feature. Let $D_\top$ be the set of formulas $d(x) \in NS$ in one variable. For each formula $d(\bar{u}) \in NS$, let $D_{d(\bar{u})}$ be the set of formulas $d'(\bar{u}, x) \in NS$ such that $d(\bar{u})$ & $d'(\bar{u}, x) \in NS$. In both $\mathcal{A}$ and $\mathcal{B}$, the formulas in $D_\top$ define the orbits of single elements. The formulas in $D_{d(\bar{u})}$ define the orbits of one-element extensions of tuples in the orbit that is defined by $d(\bar{u})$.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

Proceeding much as Scott did, we arrive at a common Scott sentence φ that is the conjunction of the following:

1. the sentence saying $(\forall x)\bigvee_{d\in D_\top} d(x)$ & $\bigwedge_{d\in D_\top}(\exists x)d(x)$, and
2. a sentence for each $d(\bar{u})\in NS$ saying that for all $\bar{u}$ satisfying $d(\bar{u})$, $(\forall x)\bigvee_{d'\in D_{d(\bar{u})}} d'(\bar{u},x)$ & $\bigwedge_{d'\in D_{d(\bar{u})}}(\exists x)d'(\bar{u},x)$.

Scott built a Scott sentence for a structure, focusing on tuples in the structure, and choosing for each tuple a particular formula that defines the orbit. We focus directly on formulas that define the orbits. We use the fact that for all tuples in the orbit defined by a formula $d(\bar{u})$ (in either structure), the formulas $d'(\bar{u},x)$ that define the orbits of 1-element extensions are the same. ⊣

We are now ready to prove Proposition 3.4.

Proof of Proposition 3.4. Recall that S is the $\forall\exists$-theory of $\mathcal{A}$, which is assumed to be Σ^0_2.

Define the sentence $\Theta = [\bigwedge_{\bar{x}}\forall\bar{x}\bigvee_{i\in\omega}\varphi_i(\bar{x})]$ & $\bigwedge_{\psi\in S}\psi$. The first conjunct is computable Π_2, so consider the second. Since the set S is Σ^0_2, we have the standard trick of creating a computable sequence of computable Σ^0_2 sentences τ_n built out of $\top$ and $\bot$ to define membership in S of the nth finitary $\forall\exists$-sentence ψ_n. Therefore, the second conjunct is logically equivalent to the computable Π_2 sentence $\bigwedge_{n\in\omega}(\tau_n\to\psi_n)$. Thus, Θ is computable Π_2. ⊣

Claim: Θ is a Scott sentence for $\mathcal{A}$

Proof of Claim. Let $\mathcal{B}$ be a countable model of Θ. We show that $\mathcal{A}\cong\mathcal{B}$ using the lemma above.

1. $\mathcal{A}$ has a Π_2 Scott sentence.
   Each tuple $\bar{a}$ in $\mathcal{A}$ satisfies some $\varphi_i(\bar{x})$. Then the orbit is defined by one of finitely many $\exists$-formulas $\gamma_j(\bar{x})$.
2. $NS(\mathcal{A})\subseteq NS(\mathcal{B})$.
   Let $d(\bar{x})$ be a formula in $NS(\mathcal{A})$. Since $\mathcal{B}$ satisfies the $\forall\exists$-theory of $\mathcal{A}$, $d(\bar{x})$ is satisfied in $\mathcal{B}$. If $d(\bar{x})$ split in $\mathcal{B}$, we would have an $\exists$-formula $d'(\bar{x})$ and tuples $\bar{b}_1,\bar{b}_2$, such that $\bar{b}_1$ satisfies $d(\bar{x})$ & $d'(\bar{x})$ and $\bar{b}_2$ satisfies $d(\bar{x})$ & $\neg d'(\bar{x})$. However, the $\forall\exists$ sentence saying
   $$(\forall\bar{x},\bar{x}')[(d(\bar{x})\ \&\ d(\bar{x}')\ \&\ d'(\bar{x}))\to d'(\bar{x}')]$$
   is true in $\mathcal{A}$, so it is also true in $\mathcal{B}$.
3. $\mathcal{B}$ has a Π_2 Scott sentence.
   Each tuple $\bar{b}$ in $\mathcal{B}$ satisfies some $\varphi_i(\bar{x})$, so it satisfies one of the $\exists$-formulas $\gamma_j(\bar{x})$. Since $\gamma_j(\bar{x})$ defines the orbit of some tuple in $\mathcal{A}$, it is in $NS(\mathcal{A})$, so it is also in $NS(\mathcal{B})$. Then by Corollary 1.3, $\mathcal{B}$ has a Π_2 Scott sentence.
4. $NS(\mathcal{B})\subseteq NS(\mathcal{A})$.
   Let $d(\bar{x})\in NS(\mathcal{B})$. Then $d(\bar{x})$ is satisfied by some $\bar{a}$ in $\mathcal{A}$. Say that $\gamma(\bar{x})$ is an $\exists$-formula that defines the orbit of $\bar{a}$. Suppose that $d(\bar{x})$ splits in $\mathcal{A}$. We have some $\bar{a}'$, not automorphic to $\bar{a}$, but also satisfying $d(\bar{x})$. Say that the orbit of $\bar{a}'$ is defined by the $\exists$-formula $\gamma'(\bar{x})$. Then the $\exists$-sentence $(\exists\bar{x})[d(\bar{x})\ \&\ \gamma'(\bar{x})]$ is true in $\mathcal{A}$ but false in $\mathcal{B}$, a contradiction.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

We have shown that the hypotheses of Lemma 3.5 are satisfied. Therefore, $\mathcal{A} \cong \mathcal{B}$. This completes the proof that Θ is a computable Π_2 Scott sentence for $\mathcal{A}$. $\dashv$

Once again, however, we see that Proposition 3.4 does not provide a necessary condition for having a computable Π_2 Scott sentence.

PROPOSITION 3.6. *There is a computable structure $\mathcal{A}$ such that*:

1. *$\mathcal{A}$ has a computable Π_2 Scott sentence*;
2. *for any c.e. set S of existential formulas, if each element of $\mathcal{A}$ satisfies some $\alpha \in S$, then there is some $\alpha \in S$ that is satisfied by elements in infinitely many orbits.*

PROOF. The structure $\mathcal{A}$ is, essentially, a countable family of unary structures, based on a uniformly computable family of trees T_e with no dead ends. For each e, T_e will be either the set of finite sequences of 0's, or the set of sequences of form $\sigma_e \tau$, where σ_e is a finite sequence of 0's of a special length and τ has all of its 1's before any of its 0's. In the first case, T_e has just one path. In the second, T_e has infinitely many paths. All but one are isolated. The language of $\mathcal{A}$ consists of unary relations U_e for $e \in \omega$ and $V_{e,n}$ for $e, n \in \omega$. In $\mathcal{A}$, the relations U_e partition the universe into infinite sets, and the relations $V_{e,n}$ are subsets of U_e. The elements of U_e represent paths through T_e via their membership (or not) in the $V_{e,n}$. More specifically, each node $\sigma \in T_e$ corresponds to a quantifier-free formula d_σ saying that $x \in U_e$ and $x \in V_{e,n}$ if $\sigma(n) = 1$ and $x \notin V_{e,n}$ if $\sigma(n) = 0$. In $\mathcal{A}$, each element a is in exactly one U_e, where it represents some isolated path in T_e. Some σ isolates the path and then d_σ defines the orbit. Moreover, each isolated path of T_e is represented in $\mathcal{A}$ by infinitely many elements.

To construct T_e, we start putting in at step s just 0^s. If for some s, when we have put into $T_e 0^n$ for all $n < s$, we see that $W_{e,s}$ has a quantifier-free formula $d(x)$ satisfied by an element of U_e that is not in any $V_{e,n}$, then T_e is the result of attaching to the sequence 0^s the sequences σ of form $1^j 0^k$. We assume that at stage s, $W_{e,s}$ can only contain formulas with symbols $V_{e,n}$ for $n < s$. If there is no such d in W_e, then T_e has just the path 0^*. Thus, either T_e has just the path 0^*, and the elements representing that path do not satisfy any formula $d \in W_e$, or else T_e has a formula d satisfied by elements representing all but the non-isolated path in T_e, infinitely many.

We now describe the explicit construction of $\mathcal{A}$ from the uniformly computable family of trees T_e. The universe of $\mathcal{A}$ is divided into infinitely many infinite, computable sets A_e. For each stage s that T_e just looks like the set of nodes of all 0's, we assign the sth element a of A_e to this path, so $U_e(a)$ and $\neg V_{e,n}(a)$ for all n. If we see at some stage s that T_e is not just the set of nodes of all 0's, then we know exactly what the *paths* in T_e look like: (1) the path of all 0's; (2) paths that begin with 0^s and continue with a 1^j for some j and then have all 0's; and (3) the sole non-isolated path of 0^s followed by all 1's. We divide the remaining elements of A_e into infinitely many infinite, computable sets and assign infinitely many elements to each of the isolated paths in T_e, i.e., those of the form (1) or (2).

For a Scott sentence φ, we take the conjunction of the following:

1. the existential and universal sentences true in $\mathcal{A}$—the conjunction of the existential sentences is computable Π_2, and the conjunction of the universal sentences is also equivalent to a computable Π_2 sentence;

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

2. the computable Π_2 sentence $(\forall x) \bigvee_e U_e x$;
3. the computable Π_2 sentences $(\exists x) V_{e,n} x \to (\forall x) \bigvee_{n'>n} \neg V_{e,n'} x$, for all e and n.

Let $\mathcal{B}$ be some model of φ. By (2), in $\mathcal{B}$, each element is in some U_e. Also, because the existential theories match, if σ is an isolating node, then there are infinitely many elements satisfying d_σ. There are universal sentences $(\forall x)(U_e(x) \to \bigvee_{\sigma \in T_{e,n}} d_\sigma(x))$, saying that each element of U_e in $\mathcal{B}$ satisfies d_σ for some σ at level n in T_e. These sentences guarantee that each element of U_e represents some path in T_e. We must show that no element of $\mathcal{B}$ represents a non-isolated path. We suppose that $W_{e,s}$ has a quantifier-free formula d_e true of elements representing the all-0 path in T_e. We may suppose that d_e mentions only $V_{e,n}$ for $n < s$. The non-isolated path in T_e has s0's, followed by all 1's. By (3), for e, s, this path is not represented in $\mathcal{B}$. For an isomorphism between $\mathcal{A}$ and $\mathcal{B}$, we just map elements of $\mathcal{A}$ in U_e and representing a given path in T_e to elements of $\mathcal{B}$ in U_e and representing the same path.

We need to show that if every element of $\mathcal{A}$ satisfies some existential formula in W_e, then some existential formula $d \in W_e$ is satisfied by elements in infinitely many orbits. We will show something stronger. The theory of $\mathcal{A}$ has effective elimination of quantifiers, so we may replace each existential formula by a quantifier-free formula that is equivalent over the theory. We may suppose that if d is enumerated into W_e at stage s, then d mentions only $V_{e,n}$ for $n < s$, not for $n \geq s$. Let c_e be an element of U_e representing the path p^* in T_e. If W_e has an existential/quantifier-free formula d satisfied by c_e, then T_e has infinitely many paths, and d is satisfied by all elements of U_e—infinitely many representing each of the isolated paths. ⊣

§4. Characterizing KU. In this section, we give a condition that, at least for unary structures $\mathcal{A}$ with a Π_2 Scott sentence and a computable theory, is necessary and sufficient for $\mathcal{A}$ to have a computable Π_2 Scott sentence. This characterization is not very satisfying, although we will use it in Section 5.

THEOREM 4.1. *Let $\mathcal{A}$ be a unary structure with a Π_2 Scott sentence and a computable theory. Then $\mathcal{A}$ has a computable Π_2 Scott sentence iff there is a computable quantifier-free Π_2 formula $\psi(x)$ that, in all models of $Th_{\mathcal{A}}$, is satisfied by just the elements that realize isolated types.*

REMARKS. For a quantifier-free formula $\psi(x)$, satisfaction of $\psi(x)$ by an element b in a unary structure depends just on the quantifier-free type of b. If T is the tree associated with $\mathcal{A}$, then $Th_T = Th(\mathcal{A})$. The quantifier-free type of an element b in a model of Th_T is determined by the path that b represents. Thus, we may say that $\psi(x)$ is satisfied by the *path*. A quantifier-free Π_2 formula $\psi(x)$ is satisfied by the path p if for each conjunct, there is some disjunct $\alpha(x)$ that satisfies p. The quantifier-free formula α, in a finite language L_s, is equivalent to a finite disjunction of $\sigma(x)$ for $\sigma \in 2^s$, and p satisfies $\alpha(x)$ if $\sigma(x) \subseteq p$, for one of these $\sigma(x)$.

PROOF OF THEOREM 4.1. We split the proof into some lemmas. The first lemma gives the easier implication. ⊣

LEMMA 4.2. *Let $\mathcal{A}$ be a unary structure with a Π_2 Scott sentence and a computable theory. Suppose further that there is a computable Π_2 formula $\psi(x)$ that is satisfied, in all models of $Th(\mathcal{A})$, by just the elements that realize isolated types. Then $\mathcal{A}$ has a computable Π_2 Scott sentence.*

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

PROOF. Since $\mathcal{A}$ is unary, $Th(\mathcal{A})$ is generated by its $\forall \cup \exists$-sentences. Let β be a computable Π_2 sentence equivalent to the conjunction of these. (We use that $Th(\mathcal{A})$ is computable here.) Then $\varphi = \beta \,\&\, (\forall x)\psi(x)$ is also computable Π_2. We claim that φ is a Scott sentence for $\mathcal{A}$.

Clearly, $\mathcal{A} \models \beta$. Since $\mathcal{A}$ has a Π^0_2 Scott sentence, $\mathcal{A}$ is atomic. So, $\mathcal{A} \models (\forall x)\psi(x)$ and thus $\mathcal{A} \models \varphi$. Let $\mathcal{B}$ be another model of φ. Satisfying β implies that a model is a unary structure with $Th(\mathcal{A})$; satisfying $(\forall x)\psi(x)$ implies that the model is atomic. (More specifically, the formulas $\sigma(x)$, for $\sigma \in 2^{<\omega}$, that are satisfied and do not split are the same for all models of β. The quantifier-free type generated over $Th(\mathcal{A})$ by such a $\sigma(x)$ is determined by β and σ, and β guarantees that $\sigma(x)$ is satisfied by infinitely many elements.) Since $\mathcal{A}$ and $\mathcal{B}$ are countable atomic models of the same theory, $\mathcal{A} \cong \mathcal{B}$. ⊣

For the harder implication, we suppose that $\mathcal{A}$ has a computable Π_2 Scott sentence φ. We must find a computable quantifier-free Π_2 formula $\psi(x)$ that is satisfied by just the isolated paths. Let T be the tree associated with $\mathcal{A}$, so that $Th(\mathcal{A}) = Th_T$. The model $\mathcal{A}$ has no elements representing non-isolated paths. We consider models of a special form. For a path p through T, let $\mathcal{B}_p = \mathcal{A} \cup \{b\}$, where b is a further element, not in $\mathcal{A}$, representing the path p. The following is clear.

LEMMA 4.3. *$\mathcal{B}_p \models \varphi$ iff p is isolated.*

The next lemma gives a computable Π_2 formula $\psi(x)$ of a special form.

LEMMA 4.4. *Let T be a computable tree with no dead ends and with isolated paths dense, and let $\mathcal{A}$ be the prime model of Th_T. Suppose $\mathcal{A}$ has a computable Π_2 Scott sentence φ. Then there is a computable Π_2 formula $\psi(x)$, a c.e. conjunction of c.e. disjunctions of formulas $\sigma(x)$ for $\sigma \in 2^{<\omega}$ such that for all paths p, $\mathcal{B}_p \models \psi(b)$ iff p is isolated.*

PROOF. We may suppose that $\varphi = \bigwedge_i (\forall \bar{u}_i)\psi_i(\bar{u}_i)$, where for each i, $\psi_i(\bar{u}_i)$ is computable Σ_1. In Th_T, we have quantifier elimination. We may suppose that $\psi_i(\bar{u}_i)$ has the form $\bigvee_j \alpha_{i,j}(\bar{u}_i)$, where $\alpha_{i,j}(\bar{u}_i)$ is quantifier-free. In addition, we may suppose that $\alpha_{i,j}(\bar{u}_i)$ gives the complete quantifier-free type of $\bar{u}_i$ in some finite language $L_s = \{U_n : n < s\}$, saying for each pair of variables $u, v \in \bar{u}_i$ whether $u = v$, and saying for each single variable v, that $\sigma(v)$ holds for some $\sigma \in 2^s$.

For the models $\mathcal{B}_p = \mathcal{A} \cup \{b\}$, we may fix $\mathcal{A}$, say with universe the odd numbers, and we may let $b = 0$. For each i, we consider the possible *assignments* t mapping $\bar{u}_i$ to $\mathcal{A} \cup \{b\}$. For an assignment t taking $\bar{u}_i$ to $\mathcal{A} \cup \{b\}$, let $\bar{x}_{i,t} = t^{-1}(b)$ and let $\bar{y}_{i,t} = t^{-1}(\mathcal{A})$. For the least s such that $\alpha_{i,j}(\bar{u}_i)$ is in the language L_s, we may write $\alpha_{i,j}(\bar{u}_i)$ as $\rho_{i,j}(\bar{u}_i)$ & $\beta_{i,t,j}(\bar{x}_{i,t})$ & $\gamma_{i,t,j}(\bar{y}_{i,t})$, where $\rho_{i,j}(\bar{u}_i)$ gives the equality relation on $\bar{u}_i$, $\beta_{i,t,j}(\bar{x}_{i,t})$ assigns some $\sigma_x \in 2^s$ to each $x \in \bar{x}_{i,t}$, and $\gamma(\bar{y}_{i,t})$ assigns some $\sigma_y \in 2^s$ to each $y \in \bar{y}_{i,t}$. Say that σ_u is the element of T assigned to the variable u in $\alpha_{i,j}(\bar{u}_i)$.

We write $\alpha_{i,j}(t(\bar{u}_i))$ for the result of plugging in $t(u)$ for the variable u in the formula $\alpha_{i,j}(\bar{u}_i)$. The formula is satisfied in $\mathcal{B}_p$ by the assignment just in case all three parts are satisfied. We have $\mathcal{B}_p \models \rho_{i,j}(t(\bar{u}_i))$ when $t(u) = t(v)$ iff $u = v$ is a conjunct of $\rho_{i,j}$. We can check this effectively, just looking at $\rho_{i,j}(\bar{u}_i)$ and t. We have $\mathcal{B}_p \models \gamma_{i,t,j}(t(\bar{y}_{i,t}))$ iff $\mathcal{A} \models \bigwedge_{y \in \bar{y}_{i,t}} \sigma_y(t(y))$. Finally, $\mathcal{B}_p \models \beta(t(\bar{x}_{i,t}))$ iff $\sigma_x \subseteq p$ for all $x \in \bar{x}_{i,t}$.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

For each i and each assignment t mapping $\bar{u}_i$ to the universe of $\mathcal{B}_p$, let $S(i,t) = \{j : \mathcal{B}_p \models \rho_{i,j}(t(\bar{u}_i))$ and $\mathcal{B}_p \models \gamma_{i,t,j}(t(\bar{y}_{i,t}))\}$. Our $\psi(x)$ should say that for all i and all $t : \bar{u}_i \to \mathcal{A} \cup \{b\}$, there is some $j \in S(i,t)$ such that t makes $\beta_{i,t,j}(\bar{x}_{i,t})$ true.

In case $\mathcal{A}$ is computable, then for each i, t, the set $S(i,t)$ is c.e., depending only on $\mathcal{A}$ and the c.e. set of disjuncts from the original φ. We can then let $\psi(x) = \bigwedge_{i,t} \bigvee_{j \in S(i,t)} \beta^*_{i,t,j}(x)$, where $\beta^*_{i,t,j}(x)$ is the result of replacing each variable in $\bar{x}_{i,t}$ by x. This is computable Π_2, with the desired meaning.

In case $\mathcal{A}$ is not computable, the construction of $\psi(x)$ is a little more complicated. We first recall the set $NS(\mathcal{A})$, defined in Section 3, which is actually the same for all models of Th_T, and is easily seen to be a Π^0_1 set, since T is computable. Next, for each i and each assignment t, and each quantifier-free formula $d(\bar{y}_{i,t})$, let $S(i,t,d) = \{j : \mathcal{B}_p \models \rho_{i,j}(t(\bar{u}_i))$ and $Th_T \vdash d(\bar{y}_{i,t}) \to \gamma_{i,t,j}(\bar{y}_{i,t})\}$. Note that $S(i,t,d)$ is computable; it doesn't, in fact, even depend on where the assignment t sends $\bar{y}_{i,t}$ in $\mathcal{A}$ beyond whether the elements are equal or not.

Now we are almost ready to construct our modified $\psi(x)$. First, for each i consider the set C_i consisting of the conjuncts λ_t, one for each assignment t, from the $\psi(x)$ we defined in case $\mathcal{A}$ was computable; i.e., $\lambda_t = \bigvee_{j \in S(i,t)} \beta^*_{i,t,j}(x)$. For each quantifier-free formula $d(\bar{y}_{i,t})$, let $\lambda_{t,d} = \bigvee_{j \in S(i,t,d)} \beta^*_{i,t,j}(x)$. Now let $C^*_i = \{\lambda_{t,d} : d(\bar{y}_{i,t}) \in NS(\mathcal{A})\}$. ⊣

Claim: $C_i = C^*_i$

Proof of Claim. Let $\lambda_t \in C_i$, and let $\bar{a} = t(\bar{y}_{i,t})$. Since $\mathcal{A}$ has a Π_2 Scott sentence, there is $d(\bar{y}_{i,t}) \in NS(\mathcal{A})$ with $\mathcal{A} \models d(\bar{a})$. Thus, for all j, we have $j \in S(i,t)$ iff $j \in S(i,t,d)$. Therefore, $\lambda_t = \lambda_{t,d} \in C^*_i$. Conversely, let $\lambda_{t,d} \in C^*_i$. Let $\bar{a} \in \mathcal{A}$ with $\mathcal{A} \models d(\bar{a})$. Let t' be the assignment with $t'(\bar{x}_{i,t}) = t(\bar{x}_{i,t})$, and $t'(\bar{y}_{i,t}) = \bar{a}$. Thus, for all j, we have $j \in S(i,t')$ iff $j \in S(i,t,d)$. Therefore, $\lambda_{t,d} = \lambda_{t'} \in C_i$. ⊣

Finally, since $NS(\mathcal{A})$ is Π^0_1, we have a computable sequence of computable Π_1 propositional formulas $(\tau_n)_{n \in \omega}$, built up from $\top$ and $\bot$, such that τ_n is logically equivalent to $\top$ if n is (the code for) a formula in $NS(\mathcal{A})$, and $\bot$ otherwise. For our formula $\psi(x)$, we take the conjunction, over i, t and quantifier-free $d(\bar{y}_{i,t})$, of the formulas $(\tau_d \to \bigvee_{j \in S_{(i,t,d)}} \beta^*(x))$. This computable Π_2 formula has the meaning we want.

To complete the proof of the harder implication, we prove the following.

Lemma 4.5. *Let $\psi(x)$ be a quantifier-free computable Π_2 formula such that for all paths p in T, $\mathcal{B}_p \models \psi(b)$ iff p is isolated. Then in all models $\mathcal{B}$ of Th_T, $\psi(x)$ is satisfied by just those elements b that represent an isolated path.*

Proof. Let $\mathcal{B}$ be a model of Th_T, and let b be an element. Since $\mathcal{A}$ is the prime model, we may suppose that $\mathcal{A} \subseteq \mathcal{B}$. Moreover, since $\mathcal{A}$ has infinitely many elements representing any given isolated path, we may suppose that $b \notin \mathcal{A}$. Since the formula $\psi(x)$ is quantifier-free, with just the variable x, $\mathcal{B} \models \psi(b)$ iff $\mathcal{A} \cup \{b\} \models \psi(b)$. ⊣

This completes the proof of Theorem 4.1.

§5. Π^1_1-completeness. For a class K, the *index set*, denoted by $I(K)$, is the set of computable indices of structures in K. Recall that for a fixed computable language

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

L, the set K_L denotes the class of L-structures with a computable Π_2 Scott sentence, and KU denotes the class of unary structures with a computable Π_2 Scott sentence. In this section, we first show that for an arbitrary computable language L, $I(K_L)$ is Π^1_1. From this, we get the fact that $I(KU)$ is Π^1_1. Next, we show, with some effort, that $I(KU)$ is complete Π^1_1. From this, we conclude that for any rich computable language L, $I(K_L)$ is complete Π^1_1.

5.1. $I(K_L)$ and $I(KU)$ are Π^1_1.

PROPOSITION 5.1. *For an arbitrary computable language L, $I(K_L)$ is Π^1_1.*

PROOF. Using Lemma 3.5, we see that $\mathcal{A}$ has a computable Π_2 Scott sentence iff:

1. $\mathcal{A}$ has a Π_2 Scott sentence, and
2. there is a computable Π_2 sentence φ such that:
   (a) $\mathcal{A} \models \varphi$, and
   (b) for all $\mathcal{B}$, if $\mathcal{B} \models \varphi$, then
      i. $NS(\mathcal{A}) = NS(\mathcal{B})$, and
      ii. each tuple in $\mathcal{B}$ satisfies some $d \in NS(\mathcal{A})$.

We calculate separately the complexities for the pieces, and then put them together.

- To say of a structure $\mathcal{A}$ and an existential formula $d(\bar{x})$ that $d(\bar{x}) \in NS(\mathcal{A})$ is computable Π_2.
- To say that $\mathcal{A}$ has a Π_2 Scott sentence, we say that every tuple satisfies some formula in $NS(\mathcal{A})$. This is computable Π_4.
- For a computable Π_2 sentence φ, saying that $\mathcal{A} \models \varphi$ is computable Π_2, uniformly.
- For structures $\mathcal{A}$ and $\mathcal{B}$, to say that $NS(\mathcal{A}) = NS(\mathcal{B})$, we take the conjunction over existential formulas $d(\bar{x})$ of formulas saying that $d(\bar{x})$ is in $NS(\mathcal{A})$ iff it is in $NS(\mathcal{B})$. This is computable Π_3.
- To say of a tuple $\bar{b}$ that it satisfies in $\mathcal{B}$ some $d(\bar{x}) \in NS(\mathcal{A})$, we take the disjunction, over existential formulas $d(\bar{x})$, of formulas saying that $d(\bar{x}) \in NS(\mathcal{A})$ and $\mathcal{B} \models d(\bar{b})$. This is computable Σ_3.
- Saying that every tuple in $\mathcal{B}$ satisfies some formula in $NS(\mathcal{A})$ is computable Π_4.

To say that $\mathcal{A}$ has a computable Π_2 Scott sentence, we say that $\mathcal{A}$ has a Π_2 Scott sentence, and there is a computable Π_2 sentence φ, true in $\mathcal{A}$, such that for every $\mathcal{B}$ satisfying φ, $NS(\mathcal{A}) = NS(\mathcal{B})$ and each tuple in $\mathcal{B}$ satisfies some $d(\bar{x})$ in $NS(\mathcal{A})$. This has the form Π^c_4 & $\bigvee_\varphi(\Pi^c_2$ & $(\forall\mathcal{B})(\Pi^c_2 \to \Pi^c_4))$, where Π^c_2 and Π^c_4 refer to computable Π_2 and computable Π_4 statements, respectively. By results of Kleene ([13, p. 375] or [4, p. 76]), we can bring the number quantifier $\bigvee_\varphi$ inside the set quantifier $(\forall\bar{B})$ and make other transformations to arrive at the fact that $I(K_L)$ is Π^1_1. ⊣

COROLLARY 5.2. *$I(KU)$ is Π^1_1.*

PROOF. If L is the language of unary structures, then $I(KU)$ is the intersection of the Π^1_1 set $I(K_L)$ with the set of indices of unary structures, which, by Lemma 2.7, is Π^0_2. ⊣

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

5.2. Completeness for $I(KU)$. In what follows, we shall use the material from Section 2.4.

THEOREM 5.3. *$I(KU)$ is complete Π^1_1.*

PROOF. To show that $I(KU)$ is complete Π^1_1, we show that $I(WF) \leq_m I(KU)$, where WF is the class of sub-trees of $\omega^{<\omega}$ that have no path. Given an index for a computable tree $S \subseteq \omega^{<\omega}$, we will pass effectively to a triple of indices for a computable tree $T \subseteq 2^{<\omega}$, a computable unary structure $\mathcal{A}$, and a computable Π_2 formula $\Phi(x)$. The tree T has no dead ends, and the isolated paths are dense. The structure $\mathcal{A}$ is the atomic model of Th_T, with a Π_2 Scott sentence. We must arrange that $\mathcal{A}$ has a computable Π_2 Scott sentence iff $S \in WF$. By Theorem 4.1, $\mathcal{A}$ has a computable Π_2 Scott sentence iff there is a quantifier-free computable Π_2 formula $\Psi(x)$, that is satisfied, in models of Th_T, by all elements that represent isolated paths and not by any elements that represent non-isolated paths. If $S \in WF$, then the formula $\Phi(x)$ will have this property. If $S \notin WF$, then we make sure that no such computable Π_2 formula works. The following technical lemma gives a nice list of the formulas $\Psi_e(x)$ that we need to consider. ⊣

LEMMA 5.4. *There is a computable list of computable quantifier-free Π_2 formulas $\Psi_e(x) = \bigwedge_k \Psi^k_e(x)$, such that every computable quantifier-free Π_2 formula $\Psi(x)$ is logically equivalent to some $\Psi_e(x)$, and each $\Psi_e(x)$ has the following properties*:

1. *for each k, the conjunct $\Psi^k_e(x)$ is a c.e. disjunction of formulas $\sigma(x)$ for $\sigma \in 2^{<\omega}$;*
2. *if $k' < k$, then $\Psi^k_e(x)$ logically implies $\Psi^{k'}_e(x)$.*

PROOF. We can massage a computable quantifier-free Π_2 formula to arrive at an equivalent formula with Properties (1) and (2). For the given $\Psi(x)$, the set C of conjuncts is c.e. We may suppose that C is the range of a total computable function $k \to \Psi^k(x)$. If in k steps of enumerating C, we see a new element, then we let $\Psi^k(x)$ be the first, and if there is no new element, then we let $\Psi^k(x)$ be the computable Σ_1 formula with the single disjunct $\top$, or $\sigma(x)$, for $\sigma = \emptyset$. Each $\Psi^k(x)$ is the disjunction of a c.e. set of quantifier-free formulas $\alpha(x)$. For $\alpha(x)$ in the finite language L_s, we can replace $\alpha(x)$ by a finite disjunction of formulas $\sigma(x)$, for $\sigma \in 2^s$. In this way, we arrive at a formula with property (1). For each k, we have a c.e. set D^k of disjuncts of $\Psi^k(x)$. We allow $D^k = \emptyset$. Let C^k be the (possibly empty) set of formulas obtained by choosing one disjunct from each $D^{k'}$ for $k' \leq k$, and taking the conjunction, which is equivalent to a single conjunct. The modified formulas $\Psi^k(x)$ have Property (2). ⊣

The tree T we construct will be the union of subtrees T_e, and our construction will guarantee that if S has a path and $\Psi_e(x)$ is satisfied by all elements representing isolated paths through T_e, then it is also satisfied by elements (in some non-atomic models $\mathcal{B}$ of Th_T) representing some non-isolated path in T_e. In the construction, we will refer to the natural topology on 2^ω, generated by the basic clopen sets $N_\sigma = \{f : f \supseteq \sigma\}$ for $\sigma \in 2^{<\omega}$. For the subspace $[T]$ consisting of paths through T, we have a topology consisting of the sets $U \cap [T]$, where U is open in 2^ω. For each $\Psi^k_e(x)$, the set of paths p such that the elements representing p satisfy $\Psi^k_e(x)$ is open.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

5.2.1. Construction in stages. The construction proceeds in stages. At stage s, we have a finite subtree T^s of T. We label the nodes that are terminal in T^s either fin or fin^*, where the label fin on a node σ indicates that σ already isolates a path, while the label fin^* indicates that σ may at later stages have incomparable extensions. We also label certain non-terminal nodes fin^*. These will acquire a new successor at the next stage. We have a finite set A^s of elements a in $\mathcal{A}$. To each such a, we assign a node σ that is terminal in T^s, putting into the diagram of $\mathcal{A}$ sentences $\pm U_n a$ to make a satisfy $\sigma(x)$. Toward the formula $\Phi(x)$, for each $k \leq s$, we put finitely many disjuncts into $\Phi^k(x)$. In the end, $\mathcal{A}$ should have infinitely many elements satisfying $\sigma(x)$ for each $\sigma \in T$. Each element of $\mathcal{A}$ should represent a path that is isolated. The formula $\Phi(x)$ should be satisfied, in models of Th_T, by all elements representing isolated paths, and, if $S \in WF$, by no elements representing non-isolated paths.

Stage 0. At stage 0, T^0 consists just of $\emptyset$, with the label fin^*, A^s has a single element, and $\Phi^0(x)$ has the single disjunct $\top$.

Stage $s+1$. At stage $s+1$, each node that was terminal in T^s gets at least one extension terminal in T^{s+1}. Non-terminal nodes added at stage s and labeled fin^* may get a further terminal extension in T^{s+1}.

Each element $a \in A^{s+1}$ is assigned some σ that is terminal in T^{s+1}; we include in the atomic diagram the facts given by $\sigma(a)$. Also, for each node σ terminal at stage s, we add to $\mathcal{A}$ one new element satisfying $\sigma(x)$. *This will guarantee that every isolated path in T corresponds to infinitely many elements of* $\mathcal{A}$.

If $\sigma \in T^s$ has just one extension σ' that is terminal in T^{s+1}, then all elements of A^s that satisfied $\sigma(x)$ at stage s satisfy $\sigma'(x)$ at stage $s+1$. (There is also a new element of A^{s+1} that satisfies $\sigma'(x)$.) If a node σ, terminal in T^s and labeled fin^*, has incomparable terminal extensions in T^{s+1}, and if all the terminal extensions are labeled fin^*, then for σ' the least such (in the natural ordering of nodes), all elements of A^s that satisfied $\sigma(x)$ will satisfy $\sigma'(x)$. If a node σ, terminal in T^s and labeled fin^*, has incomparable terminal extensions in T^{s+1}, and if at least one of these is labeled fin, then, for σ' the least such (in the natural ordering), all elements of A^s satisfying $\sigma(x)$ at stage s will satisfy $\sigma'(x)$ (as does one of the new elements). *This will guarantee that the elements of* $\mathcal{A}$ *correspond only to isolated paths through T.*

For our computable Π_2 formula, $\Phi(x) = \bigwedge_k \Phi^k(x)$, at each stage $s \geq k$, we put into $\Phi^k(x)$ disjuncts $\sigma(x)$ for all σ that are terminal in T^s. The tree T will have some paths p that we plan, from the beginning, to make non-isolated. We will call these "routinely non-isolated." We do not want such a p to satisfy our $\Phi(x)$. If we add a node σ of p at stage s, we give it the label fin^*, and we put into T^s a terminal extension of σ that is not on p. We plan, at stage $s+1$, to give σ another extension that is on p.

We have described completely how $\mathcal{A}$ and $\Phi(x)$ are constructed from the tree T, so the rest of the argument, until the final verification, focuses mainly on T.

5.2.2. Base of T_e, and codes for level 1 *of S.* We have a routine non-isolated path 0^* (none of whose nodes is terminal at any stage). Branching off, for each e, we have $\sigma^e = 0^{e+1}1$, the *base* of the sub-tree T_e, in which we will challenge Ψ_e. We extend σ^e to nodes coding level 1 of the tree S, the elements (m) at level 1 of S, if any. We have

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

a routine non-isolated path $q = \sigma^e 0^*$ (again none of whose nodes is terminal at any stage). Branching off, we have $\sigma^e 0^{m+1}1$, regardless of whether $(m) \in S$. If $(m) \in S$, then we add $s = \sigma^e 0^{m+1}11$, with the label fin^*. If $(m) \notin S$, then we add $\sigma^e 0^{m+1}10$, with the label fin. We will extend this to the single path $\sigma^e 0^{m+1}101^*$.

5.2.3. First challenge. Let s_1 code a node (m) at level 1 of S above σ_e. The node s_1 gives us permission to issue a first *challenge node* $v_1 = s_1 1$. To the side of v_1, we begin building a routine non-isolated path $r_1 = s_1 0^*$, with isolated paths $s_1 0^{k+1}1^*$ branching off from it. At the stage when we add v_1 to the tree, we give it the label fin^*. We also add $\tau_1 = s_1 01$, with the label fin. By our convention described above, elements of $\mathcal{A}$ that were introduced earlier and satisfy s_1 will satisfy τ_1, rather than v_1; there will be new elements that satisfy v_1.

The idea of the challenge is as follows. If $\Psi_e(x)$ is threatening to witness that $\mathcal{A}$ has a computable Π_2 Scott sentence, then r_1 should not satisfy $\Psi_e(x)$. We try to trick $\Psi_e(x)$ by extending v_1 to an isolated path p, until we see that it satisfies $\Psi_e(x)$, at which point, we introduce a split. The purpose of the path r_1 is to guarantee that, if this p extending v_1 satisfies $\Psi_e(x)$, then for at least one Ψ_e^k, the nodes that witness this satisfaction must all extend v_1. If this first challenge seems to be met, and we get permission (because (m) has extensions in S), then there will be further challenges.

5.2.4. Seeming to meet the first challenge. Extending v_1, we have a routine non-isolated path $v_1 0^*$, and branching off, we have a family of nodes $\pi_k = v_1 0^{k+1}1$, indexed by k. We will extend π_k, adding further 1's. Let $p^k = v_1 0^{k+1}1^*$. Let $\Psi_e^k(x)$ be the kth conjunct of $\Psi_e(x)$. Let O_k be the set of paths f through T that satisfy $\Psi_e^k(x)$. The sets O_k are open, and by our assumptions on the conjuncts of $\Psi_e^k(x)$, $O_1 \supseteq O_2 \supseteq O_3 \supseteq$ .... If $\Psi_e(x)$ is threatening to witness that $\mathcal{A}$ has a computable Π_2 Scott sentence, then it is not satisfied by r_1. There will be some first k such that $r_1 \notin O_k$. We look for this k. At stage s, we take the first $k \leq s$, if any, such that r_1 fails to satisfy any of the stage s disjuncts of $\Psi_e^k(x)$. Believing that we have the correct k, we watch for a disjunct of $\Psi_e^k(x)$, say $\sigma(x)$, such that $\sigma \subseteq p^k$; if no such disjunct appears, then p^k will be isolated, but the elements corresponding to p^k fail to satisfy $\Psi_e^k(x)$. Moreover, if at stage s, we see a disjunct $\sigma(x)$ of $\Psi_e^k(x)$ such that $\sigma \subseteq p^k$, then since $r_1 \notin O_k$, it must be true that $v_1 \subseteq \sigma$. Assuming that we have k and σ, the challenge v_1 *seems to be met* by the terminal node σ_1 of T_e^s such that $\sigma \subseteq \sigma_1 \subseteq p^k$.

Our guess at k will change if we later see a disjunct of Ψ_e^k that puts r_1 in O_k. The new guess, if any, is some $k' > k$. When our guess changes in this way, we abandon the part of the tree T_e that extends π_k (or any π_j for $j < k'$), canceling all challenges that extend π_k and making all of the currently terminal nodes isolated. For the new guess k', we proceed as above, except that we watch $\Psi_e^{k'}(x)$ and $p^{k'}$ instead of $\Psi_e^k(x)$ and p^k.

If r_1 does not satisfy $\Psi_e(x)$, then our guess will eventually settle on the first k such that $r_1 \notin O_k$. If v_1 is actually met, by some $\sigma_1 \subseteq p^k$, then π_k will have non-isolated extensions, as σ_1 becomes the base for further challenges. For $j \neq k$, the paths that extend π_j are all eventually isolated.

If r_1 actually *satisfies* $\Psi_e(x)$, then each guess at the first k such that $r_1 \notin O_k$ is eventually changed, and the part of the tree extending π_k is eventually

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

abandoned—with all paths becoming isolated. In this case, the only non-isolated path that extends ν_1 is $\nu_1 0^*$, but none of the nodes along this path is ever terminal in T^s, so no element in $\mathcal{A}$ represents this path and this path will not satisfy the Φ we are constructing.

5.2.5. Continuing. Suppose that we have $\sigma^e \subseteq s_1 \subseteq \nu_1 \subseteq \sigma_1$, where σ^e is the base for T_e, s_1 codes a node (m) at level 1 of S, ν_1 is a challenge permitted by s_1, and σ_1 meets the challenge. The node σ_1 serves as a new base, from which we may, if (m) has a successor (m, m') in S, pass to a node s_2 coding (m, m'). This gives us permission to issue a new challenge ν_2. The challenge ν_2 may be met in some σ_2. In general, if we have $\sigma_{i-1} \subseteq s_i \subseteq \nu_i \subseteq \sigma_i$, where s_i codes a node at level i of S, ν_i is the ith in a chain of challenges, and σ_i meets the challenge ν_i, we use σ_i as a base, after which we may, if permitted, issue a subsequent challenge.

5.2.6. Kinds of paths through T and corresponding elements in $\mathcal{A}$. During the construction, we have in mind three kinds of paths for T. (These categories of paths are not necessarily mutually exclusive, as explained below.)

1. *Definitely isolated*: A path with a node labeled fin.
2. *Routine non-isolated*: When we start building one of these paths, we have in mind that it will be non-isolated. These have come up in various parts of the construction:
   - the path 0^* from which all the base nodes σ_e branch;
   - each path of the form $q = \sigma^e 0^*$ or $q = \sigma_j 0^*$, where σ_j is a base node after j challenges have been met; and each branch off of q codes whether or not an element m at level $j+1$ extends the sequence in S coded by $s_j \subset \sigma_j$;
   - each path of the form $r = s_j 0^*$ to the side of a challenge node $\nu_j = s_j 1$;
   - each path of the form $t = \nu_j 0^*$, where ν_j is a challenge node; branching off of t are the paths of the form p^k; a node of one of these p^k may meet the challenge ν_j.
3. *Challenge*: The challenge paths are the paths that extend a challenge node ν_j, are not of the form t or q above, and do not contain a node of the form $s_k 0$ for any $k > j$.

The routine non-isolated paths will remain non-isolated, unless they extend a π_k associated with a challenge ν_j, and the corresponding r associated with this challenge is seen to satisfy Ψ_e^k, in which case all of these paths become definitely isolated.

A challenge path could be isolated, without being definitely isolated, for two reasons. First, if it is a p^k above a challenge ν_j, and the corresponding r associated with this challenge fails to satisfy a $\Psi_e^{k'}$ for some $k' < k$. Second, if it is a p^k above a challenge ν_j, and it fails to satisfy Ψ_e^k. A challenge path will be non-isolated if it is the limit of an infinite sequence of challenge nodes that, together, code a path through S, and all challenges are met by Ψ_e.

5.2.7. Formal construction and verification. Given a computable tree $S \subseteq \omega^{<\omega}$, and our computable sequence of Ψ_e, we construct our computable tree $T \subseteq 2^{<\omega}$, our computable structure $\mathcal{A}$, and our computable Π_2-formula $\Phi(x)$. The construction proceeds in stages. Since in Section 5.2.1, we described how to get the stagewise

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

approximations of $\mathcal{A}$ and $\Phi(x)$ from the stagewise approximation and labeling of T, we focus only on the latter.

We first specify what information is used and what corresponding finite portion of T is constructed and labeled at each stage.

1. Input tree S :
   At stage s, we use finitely much information about $S \subseteq \omega^{<\omega}$; $S \cap s^{<s}$. Note that this is a finite subtree of S.
2. Formulas $\Psi_e(x)$, for $e < s$, on our special list:
   At stage s, for each $e < s$ and $k \leq s$, we see for which $\sigma \in 2^{<\omega}$, $\sigma(x)$ has been enumerated by stage s as a disjunct of $\Psi_e^k(x)$.

Given this information, we construct and label the tree stagewise, with T^s the stage s approximation, as follows:

Stage 0: $T^0 = \emptyset$ with the empty node labeled fin^*.

Stage s: First, for all terminal nodes labeled fin in T^{s-1}, we extend the node uniquely.

Adding base nodes and nodes to code sequences in S:
We add to T^s the base node σ^{s-1}, and for each $m < s$, we add one of two extensions of the node $\sigma^{s-1}0^{m+1}1$, labeled accordingly, as described in Section 5.2.2, depending on whether or not (m) is a level one node in S. For all σ^e with $e < s-1$, we add one of the two extensions of the nodes $\sigma^e 0^s 1$, labeled accordingly. These nodes are terminal at this stage. The non-terminal node 0^s is labeled fin^*; and, similarly, for each $e < s$, the non-terminal node $\sigma^e 0^s$ is labeled fin^*. (These are nodes along the routine non-isolated paths of the form, respectively, 0^* and q, described in Section 5.2.6.)

Similarly, for each node in the tree extending some σ^e, if σ_j was the terminal node seen to meet a jth challenge at a previous stage, where $j < s-1$, then σ_j is the base node for the $(j+1)$th challenge. That is, σ_j is already associated with a j length sequence τ in S, and for each $m < s$, we add (if it was not already added previously) one of two extensions of the node $\sigma_j 0^{m+1}1$, labeled accordingly, depending on whether or not $\tau^\frown m$ is in S. If $\tau^\frown m$ is in S, then this extension is designated as one of the s_{j+1} nodes and labeled fin^*. Each non-terminal node $\sigma_j 0^s$ is labeled fin^*. (These nodes lie on the routine non-isolated path of the form q described in Section 5.2.6.)

Adding challenge nodes and the nodes "to the side," and their extensions:
To each of the uncanceled nodes s_j added at previous stages that code the presence of a sequence τ of length j in S, we add (if it was not already added at an earlier stage) an extension $v_j = s_j 1$, one of the jth challenge nodes; for each $k < s$, we add a node of the form $s_j 0^{k+1}1$ (if it was not already added) and label it fin. We label the non-terminal nodes of the form $s_j 0^s$ with fin^*. (These are nodes along the routine non-isolated paths of the form r described in Section 5.2.6.)

To each of these challenge nodes v_j (whether seemingly met or not met), and for each $k < s$, we add the node (if it was not already added) $\pi_k = v_j 0^{k+1}1$ and label it fin^*. For each $\pi_k = v_j 0^{k+1}1$ that was added at an earlier stage, if the action above it has not already been canceled, and no base node for a later challenge extends π_k, then there must be a unique terminal node above π_k labeled fin^*. Extend this node

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

with a 1 and label it fin^*. Each non-terminal node of the form $v_j 0^s$ is labeled fin^*. (These are nodes along the routine non-isolated paths of the form t described in Section 5.2.6.)

Note that, according to this construction, no challenge node v_j is ever terminal at any stage s, but instead its extensions along the various π_k are the terminal nodes.

Seeming to meet challenges and canceling extensions of challenges:
Based on the information used at this stage, we consider each challenge node v_j from an earlier stage that has not been canceled. There are four possibilities:

1. At a previous stage, the challenge seemed to be met via the open set O_k and a certain node extending v_j, as described in Section 5.2.4, and that still seems to be the case. Then no further action is taken for the sake of this challenge at this stage. (However, as described immediately above, we will continue to add the extensions of v_j, the nodes $\pi_{k'}$ for all k' with $k < k' < t$ at subsequent stages t, as long as v_j is not canceled.)
2. At a previous stage, the challenge seemed to be met according to the open set O_k and a certain node of $v_j 0^{k+1} 1^*$, but at stage s, we see for the first time that the routine non-isolated path r "to the side of" v_j is in O^k. Then for each $k' \leq k$, we cancel all action above $\pi_{k'} = v_j 0^{k'} 1$, uniquely extend all currently terminal nodes above $\pi_{k'} = v_j 0^{k'} 1$, and label all of these terminal nodes fin. (This will cancel any work for higher challenges that extend $v_j 0^{k'} 1$.)
3. Based on stage s information, there is now a least $k < s$ for which the challenge seems to be met via the open set O^k and a certain node extending v_j, as described in Section 5.2.4. Then let σ_j be the current terminal node in T^s with $\sigma_j \subset v_j 0^{k+1} 1^*$. Then this node is designated as the base node for the next challenge and is labeled fin^*.
4. We are not in any of the previous cases, and the challenge by v_j is not met at stage s via any $k < s$. Then no further action is taken. (Note that we already extended the nodes above the π_k extending such a v_j, as described above.)

This concludes the stage s construction and labeling of T^s.

LEMMA 5.5. *The tree T has no dead ends, with isolated paths dense.*

PROOF. We start with $T^0 = \{\emptyset\}$. At stage $s+1$, each terminal node of T_s is properly extended in T_{s+1}. Thus, there are no dead ends.

The construction of T follows the classification of its paths given in Section 5.2.6. If a node σ has an initial segment or extension labeled fin, then clearly it is an initial segment of an isolated path. So we can assume that σ has no such initial segment or extension. In particular, σ is not the initial segment or the extension of a node that is ever canceled, and σ cannot lie on or below a routine non-isolated path of the form 0^*, q, or r described in Section 5.2.6, because any node on such a path has an extension labeled fin. Therefore, there is a greatest j so that σ is an extension of a challenge node v_j and an initial segment or an extension of a π_k associated with this challenge. This π_k does not extend to a challenge node v_{j+1}, because, otherwise σ is an initial segment of the routine non-isolated path r to the side of this v_{j+1}, or an extension of v_{j+1} (but j is maximal). Since this π_k is not subject to a cancellation, either the challenge of v_j is never met, or it is met via some $k' < k$. In this case,

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

π_k itself isolates the unique path on which it lies, so σ is an initial segment of an isolated path. ⊣

LEMMA 5.6. *The structure $\mathcal{A}$ is a model of Th_T, so it is unary, and $\mathcal{A}$ has a Π_2 Scott sentence.*

PROOF. The construction in Section 5.2.1 guarantees that if σ first enters T at stage s, then at each stage $t \geq s$, we add to $\mathcal{A}$ a new element that will represent some path extending σ Therefore, each isolated path must be represented by infinitely many elements of $\mathcal{A}$.

A routine non-isolated path, once we start building it, will not have any further nodes terminal in T^s. Thus, such a path cannot be represented by any element of $\mathcal{A}$. Therefore, any element of $\mathcal{A}$ is initially assigned to a node marked fin; or to a node marked fin^* of the form s_j, π_k, or $\pi_k 1^n$ for some n, where π_k is an extension of some challenge node v_j. (Recall that challenge nodes v_j are never terminal.) Now, a terminal node s_j that is uncanceled at the next stage will extend to incomparable nodes, one of which is marked fin. By construction, any of the elements of $\mathcal{A}$ initially assigned to s_j will be assigned to this incomparable node. Any terminal node of the form π_k or $\pi_k 1^n$ will either never have incomparable nodes above it, or if it does, it is because at some later stage there is a terminal node σ_j, that serves as the next base node for challenges. At the next stage, above this σ_j, there may be a terminal node marked fin, because for the τ in S associated with v_j there may be some m such that $\tau^\frown m$ is not in S. By construction, any element in $\mathcal{A}$ initially assigned to π_k or $\pi_k 1^n$ will then be assigned to this node. If, on the other hand, at the next stage, above this σ_j, there are no such terminal nodes marked fin, it is because all of these nodes are of the form s_{j+1}; and, by construction, all of the elements in $\mathcal{A}$ initially assigned to π_k or $\pi_k 1^n$ will now be assigned to the least of these s_{j+1}. Then, at the very next stage, by construction, they will all be assigned to an extension of this s_{j+1} marked fin. Therefore, the elements of $\mathcal{A}$ all represent isolated paths through T. Therefore, $\mathcal{A}$ has a Π_2 Scott sentence. ⊣

We have shown that $\mathcal{A}$ is a unary structure with a Π_2 Scott sentence. Since the tree T is computable, Th_T is computable, and this is the theory of $\mathcal{A}$. Then by Theorem 4.1, $\mathcal{A}$ has a computable Π_2 Scott sentence iff there is a computable Π_2 formula $\Psi(x)$ that, in models of Th_T, is satisfied just by the elements that represent isolated paths.

Finally, then, we can turn our attention to the relevant matters of satisfaction of computable, quantifier-free Π_2 formulas $\Phi(x)$ and $\Psi_e(x)$ in our models of Th_T. An element b in a model $\mathcal{B}$ of Th_T satisfies $\Psi(x)$ iff it satisfies $\Psi^k(x)$ for all k, and it satisfies $\Psi^k(x)$ iff it satisfies some disjunct $\sigma(x)$ (for $\sigma \in 2^{<\omega}$). As we said in Section 4, satisfaction of these quantifier-free formulas depends only on the path represented by b, and we may just say that p *satisfies* $\Psi(x)$ if for each k, or for each sufficiently large k (since our Ψ_e have the special features), there is some disjunct $\sigma(x)$ of $\Psi^k(x)$ such that $\sigma \subseteq p$.

LEMMA 5.7. *Suppose $S \in WF$. Then the constructed Π_2 formula $\Phi(x)$ is satisfied by all isolated paths and not by any non-isolated paths.*

PROOF. First note that, since $S \in WF$, the only non-isolated paths through T are routine non-isolated. This is because any challenge path p can pass through

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

only finitely many challenge nodes, since a chain of challenge nodes corresponds to a chain of nodes in S, and an infinite chain would be a path through S. For a particular challenge path, p, let v_j be the last challenge node p extends and π_k be the extension above v_j along p. If π_k does not isolate p, then p must extend π_k through a node in T that codes how some $\tau^\frown m \notin S$; otherwise, p, since it is a challenge path, would run through a v_{j+1}.

For all k, the disjuncts of $\Phi^k(x)$ correspond to nodes that are terminal in T^s at some stage $s \geq k$. The construction ensures that the nodes along a routine non-isolated path are never terminal at any stage s, and hence are never included among the disjuncts of any $\Phi^k(x)$. All other paths are isolated, and once we have an extension of an isolating node as a terminal node in T^s, $\Phi^k(x)$ will be satisfied. ⊣

LEMMA 5.8. *Suppose $S \notin WF$. Then for all e, if $\Psi_e(x)$ is satisfied by all isolated paths through T and not by any routine non-isolated path, then it is satisfied by some non-isolated challenge path p in T_e.*

PROOF. The fact that $\Psi_e(x)$ is satisfied by all isolated paths and not by any routine non-isolated paths guarantees that all challenges are eventually met. The fact that $S \notin WF$ means that there is a path in S. We will have $\sigma^e \subseteq s_1 \subseteq v_1 \subseteq \sigma_1 \subseteq s_2 \subseteq v_2 \subseteq \sigma_2 \subseteq s_3 \subseteq v_3 \subseteq \sigma_3 \subseteq \ldots$, where s_i codes the node on this path through S at level i, v_i is a challenge, and σ_i meets the challenge. For each i, let r_i be the non-isolated path to the side of v_i. Let k_i be the first k such that r_i fails to satisfy $\Psi_e^{k_i}(x)$. All paths that extend σ_i satisfy $\Psi_e^{k_i}(x)$, and one of these is r_{i+1}. Therefore, $k_{i+1} > k_i$. Let $p = \cup_i \sigma_i$. Then p is a non-isolated challenge path through T_e. For each k, there is some i such that $k_i \geq k$. Since $\sigma_i \subseteq p$, p satisfies $\Psi_e^{k_i}(x)$. Since $\Psi_e^{k_i}(x)$ logically implies $\Psi_e^k(x)$, p satisfies $\Psi_e^k(x)$. Therefore, p satisfies $\Psi_e(x)$. ⊣

Since the above lemma is true for all e, by Theorem 4.1, Lemma 5.4, and the two preceding lemmas, we have completed the proof of Theorem 5.3.

We have shown that $I(KU)$ is complete Π^1_1. We obtain the following.

COROLLARY 5.9. *For an arbitrary rich computable language L, the index set for the class K of L-structures with a computable Π_2 Scott sentence is complete Π^1_1.*

PROOF SKETCH. We consider the language L with a single binary relation. There is an obvious Turing computable embedding Φ from the class K of unary structures to a class consisting of certain undirected graphs made up of infinitely many *daisies*, where each daisy consists of a center c, with infinitely many petals, one of size $2n+1$ or $2n+2$ for each n. We let K_D be the class of graphs isomorphic to those in the range of Φ. Let $\mathcal{A} \in K$. If $\mathcal{A}$ has a computable Π_2 Scott sentence, then it is easy to see that $\Phi(\mathcal{A})$ does. Conversely, if $\Phi(\mathcal{A})$ has a computable Π_2 Scott sentence φ_D, then we get a computable Π_2 Scott sentence for $\mathcal{A}$ by taking the conjunction of the computable Π_2 sentence characterizing unary structures and the "pullback" of φ_D (see [9]). ⊣

Acknowledgments. We thank the referee for helpful comments, and Rachael Alvir, Barbara Csima, and Matthew Harrison-Trainor for gracefully navigating the inadvertent overlap of this project with their own.

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press

Funding. The second author acknowledges travel and research support provided by Wellesley College.

REFERENCES

[1] R. Alvir, B. Csima, and M. Harrison-Trainor, *On the computability of optimal Scott sentences*. ***The Journal of Symbolic Logic***, (2025), 1–20. https://doi.org/10.1017/jsl.2025.10143.

[2] R. Alvir, N. Greenberg, M. Harrison-Trainor, and D. Turetsky, *Scott complexity of countable structures*. ***The Journal of Symbolic Logic***, vol. 86 (2021), pp. 1706–1720.

[3] R. Alvir, J. F. Knight, and C. McCoy, *Complexity of Scott sentences*. ***Fundamenta Mathematicae***, vol. 251 (2020), pp. 109–129.

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

[5] S. Badaev, *Computable enumerations of families of general recursive functions*. ***Algebra and Logic***, vol. 16 (1977), pp. 83–98.

[6] R. Baer, *Abelian groups without elements of finite order*. ***Duke Mathematical Journal***, vol. 3 (1937), pp. 68–122.

[7] D. Gonzalez, notes. Personal communication, November 7, 2024.

[8] D. Gonzalez and D. Rossegger, *Scott sentence complexities of linear orderings*, ***The Journal of Symbolic Logic*** vol. 90 (2025), no. 4, pp. 1563–92. https://doi.org/10.1017/jsl.2024.59.

[9] J. F. Knight, S. Miller, and M. Van den Boom, *Turing computable embeddings*. ***The Journal of Symbolic Logic***, vol. 72 (2007), pp. 901–918.

[10] A. Miller, *The Borel classification of the isomorphism class of a countable model*. ***Notre Dame Journal of Formal Logic***, vol. 24 (1983), pp. 22–34.

[11] A. Montalbán, *A robuster Scott rank*. ***Proceedings of the American Mathematical Society***, vol. 143 (2015), pp. 5427–5436.

[12] ———, *Effectively existentially-atomic structures*. ***Proceedings of the American Mathematical Society***, vol. 145 (2017), pp. 5363–5379.

[13] H. Rogers, ***Theory of Recursive Functions and Effective Computability***, McGraw-Hill, (later re-printed by MIT Press), Cambridge, Massachusetts, 1967.

[14] D. Scott, *Logic with denumerably long formulas and finite strings of quantifiers*, ***The Theory of Models*** (J. Addison, et al., editors), North-Holland, Amsterdam, 1965, pp. 329–341.

[15] V. Selivanov, *Enumerations of families of general recursive functions*. ***Algebra and Logic***, vol. 15 (1976), pp. 128–141.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF NOTRE DAME
USA
E-mail: knight.1@nd.edu

DEPARTMENT OF MATHEMATICS AND STATISTICS
WELLESLEY COLLEGE
USA
E-mail: karen.lange@wellesley.edu

CONGREGATION OF HOLY CROSS/US PROVINCE
UNIVERSITY OF PORTLAND
USA
E-mail: cmccoy@holycrossusa.org

https://doi.org/10.1017/jsl.2026.10186 Published online by Cambridge University Press