Computable Π^0_2 Scott sentences

Charlie McCoy, with Julia Knight and Karen Lange

November 14, 2024

Preliminary machinery

- $L_{\omega_1,\omega}$ formulas
  - finite tuples of variables
  - countable infinitary conjunctions and disjunctions
  - bulit up recursively over countable ordinals: $\Sigma_0, \Pi_0, \Sigma_\alpha, \Pi_\alpha$

Preliminary machinery

- $L_{\omega_1,\omega}$ formulas
  - finite tuples of variables
  - countable infinitary conjunctions and disjunctions
  - bulit up recursively over countable ordinals: $\Sigma_0, \Pi_0, \Sigma_\alpha, \Pi_\alpha$
- Computable $L_{\omega_1,\omega}$ formulas
  - formally, coding of formulas for $\alpha < \omega_1^{CK}$
  - countable infinitary conjunctions and disjunctions over c.e. sets of formulas

Scott sentences

Theorem 1 (Scott)

Let $\mathcal{A}$ be a countable structure for a countable language L

- *Then there is a sentence Φ in $L_{\omega_1\omega}$ that defines $\mathcal{A}$ up to isomorphism among countable structures*
- *That is, for $\mathcal{B}$ a countable L-structure, $\mathcal{A} \cong \mathcal{B}$ iff $\mathcal{B} \models \Phi$*

Observation 1

At face value, it does not appear that Φ being a Scott sentence for $\mathcal{A}$ would be known "internally" by $\mathcal{A}$; must consider all other structures $\mathcal{B}$ and assert the existence of isomorphisms

Example 1

Consider the structure $(\mathbb{N}, +, \times, S, 0, 1)$. Then it has a Scott sentence formed as the infinitary conjunction of all basic addition and multiplication facts and the infinitary sentence
$\forall x \bigvee_n (x = S^n(0))$

Montalbán's characterization

Theorem 2 (Montalbán)

Let $\alpha \geq 1$ be a countable ordinal, and $\mathcal{A}$ be a countable structure for a countable language L. Then the following are equivalent:

1. *$\mathcal{A}$ has a $\Pi_{\alpha+1}$ Scott sentence*
2. *for each $\vec{a} \in \mathcal{A}$, the automorphism orbit of $\vec{a}$ is defined by a Σ_α formula*

- The proof of 2) $\Rightarrow$ 1) goes back to Scott's own proof of his Isomorphism Theorem using Scott families

Montalbán's characterization

Theorem 2 (Montalbán)

Let $\alpha \geq 1$ be a countable ordinal, and $\mathcal{A}$ be a countable structure for a countable language L. Then the following are equivalent:

1. *$\mathcal{A}$ has a $\Pi_{\alpha+1}$ Scott sentence*
2. *for each $\vec{a} \in \mathcal{A}$, the automorphism orbit of $\vec{a}$ is defined by a Σ_α formula*

- The proof of 2) $\Rightarrow$ 1) goes back to Scott's own proof of his Isomorphism Theorem using Scott families
- Scott's proof shows this characterization is internal to $\mathcal{A}$ (automorphism replaced with back-and-forth families)

Montalbán's characterization

Theorem 2 (Montalbán)

Let $\alpha \geq 1$ be a countable ordinal, and $\mathcal{A}$ be a countable structure for a countable language L. Then the following are equivalent:

1. *$\mathcal{A}$ has a $\Pi_{\alpha+1}$ Scott sentence*
2. *for each $\vec{a} \in \mathcal{A}$, the automorphism orbit of $\vec{a}$ is defined by a Σ_α formula*

- The proof of 2) $\Rightarrow$ 1) goes back to Scott's own proof of his Isomorphism Theorem using Scott families
- Scott's proof shows this characterization is internal to $\mathcal{A}$ (automorphism replaced with back-and-forth families)
- When β is a limit ordinal, 1) $\Rightarrow$ 2) holds: that is, having a Π_β Scott sentence implies having $\Sigma_{<\beta}$ orbit-defining formulas

Montalbán's characterization

Theorem 2 (Montalbán)

Let $\alpha \geq 1$ be a countable ordinal, and $\mathcal{A}$ be a countable structure for a countable language L. Then the following are equivalent:

1. *$\mathcal{A}$ has a $\Pi_{\alpha+1}$ Scott sentence*
2. *for each $\vec{a} \in \mathcal{A}$, the automorphism orbit of $\vec{a}$ is defined by a Σ_α formula*

- The proof of 2) $\Rightarrow$ 1) goes back to Scott's own proof of his Isomorphism Theorem using Scott families
- Scott's proof shows this characterization is internal to $\mathcal{A}$ (automorphism replaced with back-and-forth families)
- When β is a limit ordinal, 1) $\Rightarrow$ 2) holds: that is, having a Π_β Scott sentence implies having $\Sigma_{<\beta}$ orbit-defining formulas
- 2) $\Rightarrow$ 1) is not true in general for limit ordinals

Effective version

Theorem 3 (Alvir, Knight, M)

For $\alpha \geq 2$ a computable ordinal, if $\mathcal{A}$ has a computable Π_α Scott sentence, then each tuple has an orbit defined by computable $\Sigma_{<\alpha}$ formula

Some natural examples:

- $(\mathbb{N}, +, \times, S, 0, 1)$
- $\langle \mathbb{Q}, + \rangle$
- Any computable subfield K of the algebraic numbers

What about the converse?

Proposition 4 (Alvir, Knight, M)

The converse of Theorem 3 is false for $\alpha = 2$

Tree out of which the counter-example is formed

The structure is built from a computable tree $T \subset 2^{<\omega}$ (constructed previously by Badaev) with the following features:

- No terminal nodes
- One non-computable path from which we can enumerate the elements of the Halting Set in order

The theory for the counter-example

The theory $\mathscr{T}$ is formed from the tree

- Language has a unary predicate U_n for each level n of the tree
- Each $\sigma \in T$ corresponds to a conjunction of $U_n(x)$ and $\neg U_n(x)$ for $n < |\sigma|$; call it $\sigma(x)$
- $\mathscr{T}$ is axiomatized by saying:
  - for each element a and each level n, there is some σ of length n with $\sigma(a)$
  - for any σ and any m, there are m elements satisfying σ

Models of the theory

- The elements of a model $\mathcal{A}$ of $\mathscr{T}$ correspond to paths through the tree

Models of the theory

- The elements of a model $\mathcal{A}$ of $\mathscr{T}$ correspond to paths through the tree
- Every model of $\mathscr{T}$ has infinitely many elements corresponding to each isolated path

Models of the theory

- The elements of a model $\mathcal{A}$ of $\mathscr{T}$ correspond to paths through the tree
- Every model of $\mathscr{T}$ has infinitely many elements corresponding to each isolated path
- The prime model has only these elements; there is a computable copy of the prime model

Models of the theory

- The elements of a model $\mathcal{A}$ of $\mathscr{T}$ correspond to paths through the tree
- Every model of $\mathscr{T}$ has infinitely many elements corresponding to each isolated path
- The prime model has only these elements; there is a computable copy of the prime model
- Non-prime models have one or more elements corresponding to the non-isolated path

The actual counter-example

The prime model $\mathcal{A}$ of $\mathscr{T}$ has a Π_2 Scott sentence, but no computable Π_2 Scott sentence

- every tuple of distinct elements $a_1, \ldots, a_k$ has an orbit that is simply the finite conjunction of the q.f. $\sigma(x_i)$ that isolates the path corresponding to a_i

The actual counter-example

The prime model $\mathcal{A}$ of $\mathscr{T}$ has a Π_2 Scott sentence, but no computable Π_2 Scott sentence

- every tuple of distinct elements $a_1, \ldots, a_k$ has an orbit that is simply the finite conjunction of the q.f. $\sigma(x_i)$ that isolates the path corresponding to a_i
- By Theorem 2, $\mathcal{A}$ has a Π_2 Scott sentence

The actual counter-example

The prime model $\mathcal{A}$ of $\mathscr{T}$ has a Π_2 Scott sentence, but no computable Π_2 Scott sentence

- every tuple of distinct elements $a_1, \ldots, a_k$ has an orbit that is simply the finite conjunction of the q.f. $\sigma(x_i)$ that isolates the path corresponding to a_i
- By Theorem 2, $\mathcal{A}$ has a Π_2 Scott sentence
- Consider $\mathcal{B}$, a model of $\mathscr{T}$ with just one element corresponding to the non-isolated, non-computable path f

The actual counter-example

The prime model $\mathcal{A}$ of $\mathscr{T}$ has a Π_2 Scott sentence, but no computable Π_2 Scott sentence

- every tuple of distinct elements $a_1, \ldots, a_k$ has an orbit that is simply the finite conjunction of the q.f. $\sigma(x_i)$ that isolates the path corresponding to a_i
- By Theorem 2, $\mathcal{A}$ has a Π_2 Scott sentence
- Consider $\mathcal{B}$, a model of $\mathscr{T}$ with just one element corresponding to the non-isolated, non-computable path f
- Any *computable* Π_2 Scott sentence true of $\mathcal{A}$ and not true of $\mathcal{B}$ would allow us to compute f

Rest of talk's focus

Question 1

When does a structure $\mathcal{A}$ with a Π_2 Scott sentence have a computable Π_2 Scott sentence?

An important feature?

Note: for the computable counter-example $\mathcal{A}$ from Proposition 4, we cannot computably assign a tuple to its orbit-defining formula

Proposition 5

Let $\mathcal{A}$ be a computable structure. If there is a computable function f so that $f(\vec{a})$ assigns $\vec{a}$ to an orbit-defining existial formula, then $\mathcal{A}$ has a computable Π_2 Scott sentence

Initial speculation: At least among computable structures, is the key to distinguishing structures that have computable Π_2 Scott sentences from those that don't somehow bound up in the complexity of a function that assigns each tuple $\vec{a}$ to an orbit-defining existential formula?

NO!!

Finding a tuple's orbit-defining existential formula

- Let $\mathcal{A}$ be a countable L-structure with a Π_2 Scott family. It is $\Pi^0_2(\mathcal{A}, L)$ to determine whether a given existential formula $\varphi(\vec{x})$ defines the orbit of a given tuple $\vec{a}$ in $\mathcal{A}$

Finding a tuple's orbit-defining existential formula

- Let $\mathcal{A}$ be a countable L-structure with a Π_2 Scott family. It is $\Pi^0_2(\mathcal{A}, L)$ to determine whether a given existential formula $\varphi(\vec{x})$ defines the orbit of a given tuple $\vec{a}$ in $\mathcal{A}$
- The computable unary structure $\mathcal{A}$ built from Badaev's tree does *not* have a computable Π_2 Scott sentence, but there is a Δ^0_2 function that takes each tuple in $\mathcal{A}$ to a quantifier free formula that defines its orbit

Finding a tuple's orbit-defining existential formula

- Let $\mathcal{A}$ be a countable L-structure with a Π_2 Scott family. It is $\Pi^0_2(\mathcal{A}, L)$ to determine whether a given existential formula $\varphi(\vec{x})$ defines the orbit of a given tuple $\vec{a}$ in $\mathcal{A}$
- The computable unary structure $\mathcal{A}$ built from Badaev's tree does *not* have a computable Π_2 Scott sentence, but there is a Δ^0_2 function that takes each tuple in $\mathcal{A}$ to a quantifier free formula that defines its orbit
- There is a computable graph $\mathcal{G}$ that *does* have a computable Π_2 Scott sentence, but there is *no* Δ^0_2 function that takes each tuple in $\mathcal{G}$ to an existential formula that defines its orbit

Finding a tuple's orbit-defining existential formula

- Let $\mathcal{A}$ be a countable L-structure with a Π_2 Scott family. It is $\Pi^0_2(\mathcal{A}, L)$ to determine whether a given existential formula $\varphi(\vec{x})$ defines the orbit of a given tuple $\vec{a}$ in $\mathcal{A}$
- The computable unary structure $\mathcal{A}$ built from Badaev's tree does *not* have a computable Π_2 Scott sentence, but there is a Δ^0_2 function that takes each tuple in $\mathcal{A}$ to a quantifier free formula that defines its orbit
- There is a computable graph $\mathcal{G}$ that *does* have a computable Π_2 Scott sentence, but there is *no* Δ^0_2 function that takes each tuple in $\mathcal{G}$ to an existential formula that defines its orbit
- There is a computable field $\mathcal{K}$ that *does* have a computable Π_2 Scott sentence, but there is *no* Δ^0_2 function that takes each tuple in $\mathcal{K}$ to an existential formula that defines its orbit

Finding a "precursor" to a tuple's orbit-defining existential formula

In the last two examples, there *is* a c.e. list of existential formulas $(\varphi_i)_{i\in\omega}$ with:

1. each $\vec{a} \in \mathcal{A}$ satisfies one of the φ_i
2. for each $i \in \omega$, there is a finite list of orbit-defining existential formulas $\gamma_1, \ldots, \gamma_{k_i}$ with $\mathcal{A} \models [\varphi_i \to (\gamma_1 \vee \cdots \vee \gamma_{k_i})]$

Finding a "precursor" to a tuple's orbit-defining existential formula

In the last two examples, there *is* a c.e. list of existential formulas $(\varphi_i)_{i\in\omega}$ with:

1. each $\vec{a} \in \mathcal{A}$ satisfies one of the φ_i
2. for each $i \in \omega$, there is a finite list of orbit-defining existential formulas $\gamma_1, \ldots, \gamma_{k_i}$ with $\mathcal{A} \models [\varphi_i \to (\gamma_1 \vee \cdots \vee \gamma_{k_i})]$

Proposition 6

Assume $\mathcal{A}$ has a Π_2 Scott sentence, and $Th(\mathcal{A})$ has some extra effectiveness. Then a c.e. list as above guarantees $\mathcal{A}$ has a computable Π_2 Scott sentence

Finding a "precursor" to a tuple's orbit-defining existential formula

In the last two examples, there *is* a c.e. list of existential formulas $(\varphi_i)_{i\in\omega}$ with:

1. each $\vec{a} \in \mathcal{A}$ satisfies one of the φ_i
2. for each $i \in \omega$, there is a finite list of orbit-defining existential formulas $\gamma_1, \ldots, \gamma_{k_i}$ with $\mathcal{A} \models [\varphi_i \to (\gamma_1 \vee \cdots \vee \gamma_{k_i})]$

Proposition 6

Assume $\mathcal{A}$ has a Π_2 Scott sentence, and $Th(\mathcal{A})$ has some extra effectiveness. Then a c.e. list as above guarantees $\mathcal{A}$ has a computable Π_2 Scott sentence

Corollary 7

Under the same initial assumptions of Proposition 6, if $Th(\mathcal{A})$ is $\aleph_0$-categorical, then $\mathcal{A}$ has a computable Π_2 Scott sentence

Once again, sufficient but not necessary

Proposition 8

There is a computable structure $\mathcal{A}$ that has a computable Π_2 Scott sentence, but there is no c.e. list of existential formulas satisfying the properties of Proposition 6

- $\mathcal{A}$ is a structure in a unary language
- built from a family of trees T_e so elements of $\mathcal{A}$ represent paths through these trees
- each T_e is constructed to foil W_e from being a list of existential formulas satisfying the properties of Proposition 6

Weak saturation, computable Π_2 Scott sentence, and $\aleph_0$-categoricity

Definition 1

$\mathcal{A}$ has the weak saturation property if any c.e. set $\Gamma(\vec{x})$ of universal formulas finitely satisfied by $\mathcal{A}$ is satisfied by $\mathcal{A}$

Proposition 9

Let $\mathcal{A}$ be a structure whose theory has some extra effectiveness; assume that $\mathcal{A}$ has a Π_2 Scott sentence and the weak saturation property. Then $\mathcal{A}$ has a computable Π^0_2 Scott sentence iff $\mathcal{A}$ is $\aleph_0$-categorical

This, proposition can best be viewed as a "negative result," because it says that, for any structure that *isn't* $\aleph_0$-categorical, weak saturation *precludes* having a computable Π^0_2 Scott sentence

Unary structures from trees

We look at the family of structures formed from trees like the one from Proposition 4

- Computable tree T with no terminal nodes, isolated paths dense
- Computable structure $\mathcal{A}$ with each element corresponding to an isolated path, each isolated path represented by infinitely many elements. So $\mathcal{A}$ has a Π_2 Scott sentence

Proposition 10

Such a structure $\mathcal{A}$ has a computable Π_2 Scott sentence iff there is a computable Π_2 formula $\psi(x)$ such that in all models of $\mathscr{T}$, $\psi(x)$ is satisfied by all elements that represent isolated paths and not by any elements that represent non-isolated paths

Is there a reason for our inability to find necessary and sufficient conditions?

Question 2

Could it be that characterizing the structures with computable Π_2 Scott sentences is just very hard (in the technical sense)?

Index set arguments with these unary structures from trees; e.g., working currently to try to produce a sequence of computable structures $\mathcal{A}_n$ so that $n \in Cof$ iff $\mathcal{A}_n$ has a computable Π_2 Scott sentence

An analogy with categoricity

- There is a nice, internal characterization for being relatively computably categorical: a Σ_1 Scott family (For relative computable categoricity, all isomorphic copies, not just the computable ones, must be considered.)

An analogy with categoricity

- There is a nice, internal characterization for being relatively computably categorical: a Σ_1 Scott family (For relative computable categoricity, all isomorphic copies, not just the computable ones, must be considered.)
- There is no such characterization for being computably categorical, because this property is Π^1_1 hard (Downey, Kach, Lempp, Lewis-Pye, Montalbán, Turetsky)

An analogy with categoricity

- There is a nice, internal characterization for being relatively computably categorical: a Σ_1 Scott family (For relative computable categoricity, all isomorphic copies, not just the computable ones, must be considered.)
- There is no such characterization for being computably categorical, because this property is Π^1_1 hard (Downey, Kach, Lempp, Lewis-Pye, Montalbán, Turetsky)
- There is a nice, internal characterization for having a Π_2 Scott sentence (The characterization captures all Π_2 Scott sentences, not just the computable ones.)

An analogy with categoricity

- There is a nice, internal characterization for being relatively computably categorical: a Σ_1 Scott family (For relative computable categoricity, all isomorphic copies, not just the computable ones, must be considered.)
- There is no such characterization for being computably categorical, because this property is Π^1_1 hard (Downey, Kach, Lempp, Lewis-Pye, Montalbán, Turetsky)
- There is a nice, internal characterization for having a Π_2 Scott sentence (The characterization captures all Π_2 Scott sentences, not just the computable ones.)
- Is it possible that there is no nice characterization for having a computable Π_2 Scott sentence, because the problem is complete at a very high level?

A possible objection to the analogy

- When comparing categoricity to Scott sentences, the implication between the items in the analogy is actually reversed
- That is, for categoricity,
(relative computable categoricity) $\Rightarrow$ (computable categoricity)
- Whereas for Scott sentences,
(having a *computable* Π_2 Scott sentence) $\Rightarrow$ (having a Scott sentence)

Thank you!

Happy birthday, Julia Knight!