The Tree of Tuples of a Structure

Matthew Harrison-Trainor

Victoria University of Wellington & Massey University

May 12, 2020

I will talk about trees and where they fit with respect to various computability-theoretic reductions between classes of structures. I will touch on:

1. universality,
2. bi-interpretations,
3. Borel reductions, and
4. coding sets and families.

We will also learn something about the back-and-forth information of a structure.

By a tree, I mean a tree as a graph with a root.

The main theorem I want to talk about is joint work with Montalbán, and I will also talk about work with Melnikov and R. Miller along the way.

All of the structures in this talk will be countable with domain ω. By a computable structure, we mean that the atomic diagram is computable.

We will talk about a couple of computability-theoretic properties as examples, and in particular:

Definition

$\mathrm{DgSp}(\mathcal{A})$ is the set of all degrees $\mathbf{d}$ such that $\mathbf{d}$ computes a copy of $\mathcal{A}$.

Definition

The computable dimension of $\mathcal{A}$ is the number of computable copies of $\mathcal{A}$ up to computable isomorphism.

It is easy to construct structures of computable dimension 1 or ω.

Theorem (Goncharov; Goncharov and Dzgoev; Metakides and Nerode; Nurtazin; LaRoche; Remmel; HT, Melnikov, and Montalbán)

Every computable structure in the following classes has computable dimension 1 *or* ω*:*

- *algebraically closed fields,*
- *real closed fields,*
- *Abelian groups,*
- *ordered Abelian groups,*
- *linear orders,*
- *Boolean algebras,*
- *differentially closed fields,*
- *and many more.*

Theorem (Goncharov)

For each $n > 0$ there is a computable structure of computable dimension n.

Theorem (Goncharov; Goncharov, Molokov, and Romanovskii; Kudinov)

There are such examples in each of the following classes:

- *graphs,*
- *lattices,*
- *partial orders,*
- *2-step nilpotent groups, and*
- *integral domains.*

The same thing happened over and over again with examples of structures with other properties, and it was noticed that for some examples—graphs, partial orders, integral domains, etc.—examples could always be found. Informally, we call such structures *universal*.

Every time we builds a new example, e.g. of a graph, we could transform that example into a lattice, a partial order, etc.

But it would be better to come up with a general construction that transfers many properties at once.

Definition (Hirschfeldt, Khoussainov, Shore, Slinko)

Let $\mathcal{C}$ be a class of structures. $\mathcal{C}$ is *complete with respect to degree spectra of nontrivial structures, effective dimensions, expansion by constants, and degree spectra of relations* if for every nontrivial countable graph $\mathcal{G}$ there is a nontrivial $\mathcal{A} \in \mathcal{C}$ with the following properties:

- $\mathrm{DgSp}(\mathcal{A}) = \mathrm{DgSp}(\mathcal{G})$;
- If G is computably presentable then:
  - For any degree $\mathbf{d}$, $\mathcal{A}$ has the same $\mathbf{d}$-computable dimension as $\mathcal{G}$;
  - If $\bar{x} \in \mathcal{G}$, then there is an $\bar{a} \in \mathcal{A}$ such that $(\mathcal{A}, \bar{a})$ has the same computable dimension as $(\mathcal{G}, \bar{x})$;
  - If $S \subseteq \mathcal{G}$ then there is $U \subseteq \mathcal{A}$ such that $\mathrm{DgSp}(S) = \mathrm{DgSp}(U)$; and if S is intrinsically c.e. then so is U.

Theorem (Hirschfeldt, Khoussainov, Shore, Slinko)

The following classes are complete with respect to degree spectra of nontrivial structures, effective dimensions, expansion by constants, and degree spectra of relations:

- *undirected graphs,*
- *partial orderings, and*
- *lattices,*

and, after naming finitely many constants,

- *integral domains,*
- *commutative semigroups, and*
- *2-step nilpotent groups.*

In fact they are complete with respect to every other reasonable computability-theoretic property we can think of.

There are other similar ideas in other areas of logic where we code structures of one kind into structures of another kind. From descriptive set theory, we have Borel reductions.

Definition

Let $\mathcal{C}$ and $\mathcal{D}$ be classes of structures. We say that $\mathcal{C}$ is Borel-reducible to $\mathcal{D}$, and write $\mathcal{C} \leq_B \mathcal{D}$, if there is a Borel operator $\Phi : \mathcal{C} \to \mathcal{D}$ such that

$$\mathcal{A} \cong \mathcal{B} \iff \Phi(\mathcal{A}) \cong \Phi(\mathcal{B}).$$

Definition

We say that a class $\mathcal{C}$ is Borel complete if the class of graphs is Borel reducible to $\mathcal{C}$.

The same proofs given by Hirschfeldt, Khoussainov, Shore, and Slinko also show that all the classes previously mentioned are Borel complete.

Theorem (Friedman, Stanley)

The following classes are also Borel complete.

- *trees,*
- *linear orders,*
- *fields of any fixed characteristic.*

Linear orders are not universal. For example, we mentioned previously:

Theorem (Goncharov and Dzgoev; Remmel)

Every computable linear order has dimension 1 *or* ω.

It was recently shown that fields can be added to the list of universal structures.

Theorem (Miller, Park, Poonen, Schoutens, Shlapentokh)

Fields of characteristic zero are complete with respect to degree spectra of nontrivial structures, effective dimensions, expansion by constants, and degree spectra of relations.

What about trees?

Let us first look at why trees are Borel complete. The main idea is:

Definition

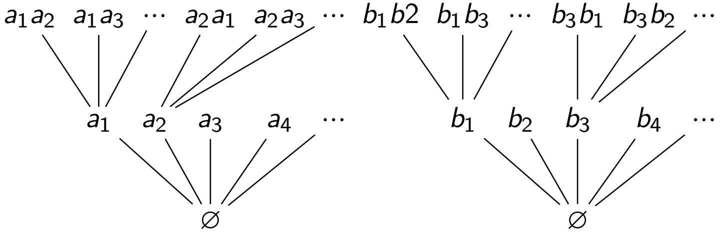
The *tree of tuples* of a structure $\mathcal{A}$, $\mathcal{T}(\mathcal{A})$, is a labeled tree consisting of all the tuples from $\mathcal{A}$ ordered by inclusion where each tuple $\bar{a}$ is labeled by a number coding its finite atomic diagram $D_{\mathcal{A}}(\bar{a})$. We can also define a tree $\mathcal{T}_\infty(\mathcal{A})$ by replicating each branch infinitely often.

The standard back-and-forth argument shows that

$$\mathcal{A} \cong \mathcal{B} \iff \mathcal{T}(\mathcal{A}) \cong \mathcal{T}(\mathcal{B}) \iff \mathcal{T}_\infty(\mathcal{A}) \cong \mathcal{T}_\infty(\mathcal{B}).$$

So $\mathcal{T}$ and $\mathcal{T}_\infty$ give a Borel reduction from graphs to labeled trees.

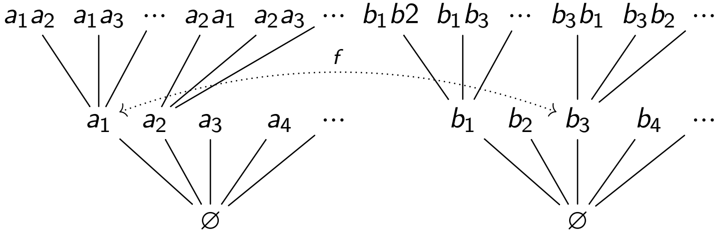
Let $\mathcal{A}$ consist of elements $a_1, a_2, a_3, \ldots$ and let $\mathcal{B}$ consist of elements $b_1, b_2, b_3, \ldots$. Suppose that $\mathcal{T}(\mathcal{A})$ and $\mathcal{T}(\mathcal{B})$ are isomorphic via an isomorphism f.



$a_1 \mapsto ?$

Think of $\mathcal{T}_\infty(\mathcal{A})$ as being the back-and-forth information of the structure $\mathcal{A}$.

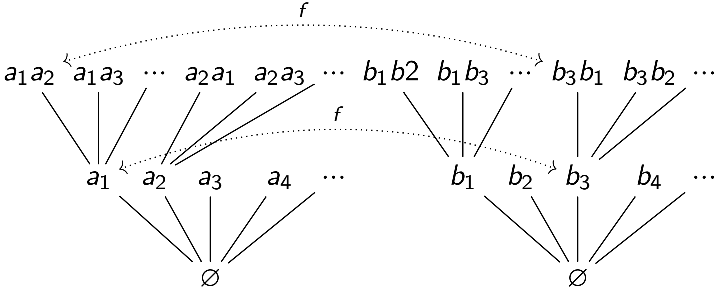
Let $\mathcal{A}$ consist of elements $a_1, a_2, a_3, \ldots$ and let $\mathcal{B}$ consist of elements $b_1, b_2, b_3, \ldots$. Suppose that $\mathcal{T}(\mathcal{A})$ and $\mathcal{T}(\mathcal{B})$ are isomorphic via an isomorphism f.



$$a_1 \mapsto b_3$$
$$? \mapsto b_1$$

Think of $\mathcal{T}_\infty(\mathcal{A})$ as being the back-and-forth information of the structure $\mathcal{A}$.

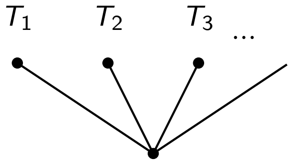
Let $\mathcal{A}$ consist of elements $a_1, a_2, a_3, \ldots$ and let $\mathcal{B}$ consist of elements $b_1, b_2, b_3, \ldots$. Suppose that $\mathcal{T}(\mathcal{A})$ and $\mathcal{T}(\mathcal{B})$ are isomorphic via an isomorphism f.



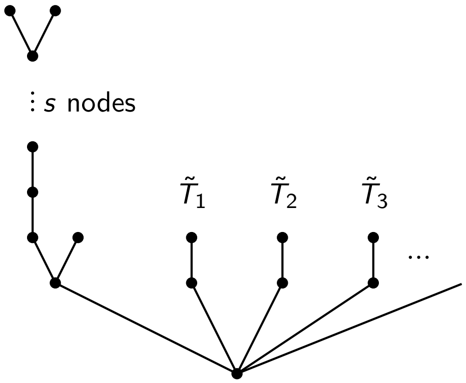
$$a_1 \mapsto b_3$$
$$a_2 \mapsto b_1$$

Think of $\mathcal{T}_\infty(\mathcal{A})$ as being the back-and-forth information of the structure $\mathcal{A}$.

$\mathcal{T}$ and $\mathcal{T}_\infty$ build labeled trees, but it is not hard to turn labeled trees into trees. Recursively replace a subtree T with root node labeled s and subtrees $T_1, T_2, \ldots$ by the tree $\tilde{T}$ as follows:



$\mathcal{T}$ and $\mathcal{T}_\infty$ build labeled trees, but it is not hard to turn labeled trees into trees. Recursively replace a subtree T with root node labeled s and subtrees $T_1, T_2, \ldots$ by the tree $\tilde{T}$ as follows:



Thus the class of trees is Borel complete.

Are trees universal?

Question

Is a structure with some reasonable computability-theoretic property such that there is no tree with the same property?

Let's look at some examples. Many examples are constructed by coding something into a structure.

The most basic kind of coding is:

Definition

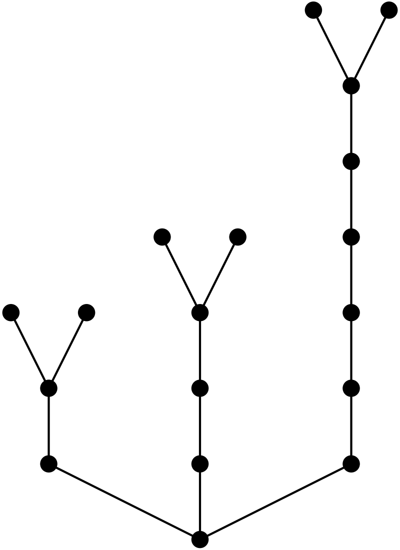
A set $X \subseteq \omega$ is *c.e.-coded* in a structure $\mathcal{A}$ if X is computably enumerable in the atomic diagram $D(\mathcal{A})$ of any presentation of $\mathcal{A}$.

Theorem (Knight)

A set X is c.e.-coded in a structure $\mathcal{A}$ if and only if X is enumeration reducible to the $\exists$-type of some tuple from $\mathcal{A}$, where the $\exists$-type of a tuple $\bar{a}$ in $\mathcal{A}$ is given by

$$\exists\,\mathrm{tp}_{\mathcal{A}}(\bar{a}) = \{\ulcorner\varphi\urcorner : \varphi(\bar{x}) \text{ is an } \exists\text{-formula such that } \mathcal{A} \models \varphi(\bar{a})\}.$$

It is easy to code a set into a tree. For example, we can code the set $W = \{1, 2, 5\}$ into a tree T_W:



The next kind of coding is to code families of sets.

Definition

An enumeration of a countable family $\mathcal{F} \subseteq \mathcal{P}(\omega)$ is a set $W \subseteq \omega^2$ such that $\mathcal{F} = \{W^{[i]} : i \in \omega\}$, where $W^{[i]}$ is the ith column of W.

To avoid issues with multiplicity, we may always assume that in an enumeration of a family each column is replicated infinitely many times.

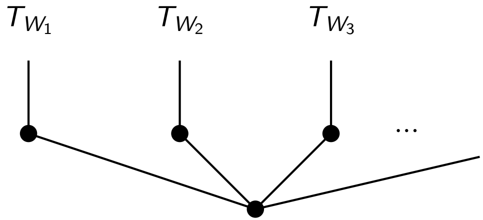
Definition

A family $\mathcal{F}$ is *computably enumerable* in a set X if there is an X-c.e. enumeration of $\mathcal{F}$.

Definition

A family $\mathcal{F}$ is *c.e.-coded* in a structure $\mathcal{A}$ if it is computably enumerable in the atomic diagrams of all copies of $\mathcal{A}$.

Coding families into trees is easy as well. We can code the family $\mathcal{F} = \{W_1, W_2, \ldots\}$ into a tree as:



where T_W is the tree define previously that codes the set W.

This immediately lets us build many interesting trees.

Theorem (Slaman; Wehner)

There is a family of finite sets which can be enumerated by every non-computable degree, but which cannot be computably enumerated.

Corollary

There is a tree whose degree spectrum is exactly the non-computable degrees.

Theorem (Goncharov)

There is a family of sets which has exactly two computable enumerations up to computable equivalence.

Corollary

There is a computable tree with computable dimension two.

Theorem (Badaev; Selivanov; Goncharov)

There is a family $\mathcal{F}$ that has one enumeration up to computable equivalence, but for some X, multiple non-X-equivalent X-enumerations.

Corollary

There is a tree which is computably categorical but not relatively computably categorical.

So when considering some computability-theoretic property that a structure might have, trees always seem to be on the side of the universal structures. And yet they did not show up in the list of structures complete for degree spectra, etc. Why?

We must first ask: What is it about the transformations used by Hirschfeldt, Khoussainov, Shore, and Slinko that causes them to maintain degree spectra, computable dimension, etc.?

When Miller, Park, Poonen, Schoutens, Shlapentokh added fields to the list of universal structures, they gave a formulation using computable functors.

Around the same time, Montalbán was thinking about effective bi-interpretations.

It turns out that these are really the same!

Definition

A relation $R \subseteq \mathcal{A}^{<\omega}$ is uniformly relatively intrinsically computable (u.r.i. computable) if it is definable by a computable Σ_1^c formula without parameters, or equivalently, if it is uniformly computable in and relative to every copy of $\mathcal{B}$.

Let $\mathcal{A} = (A; P_0^{\mathcal{A}}, P_1^{\mathcal{A}}, \dots)$ where $P_i^{\mathcal{A}} \subseteq A^{a(i)}$.

Definition

$\mathcal{A}$ is *effectively interpretable* in $\mathcal{B}$ if there exist a u.r.i. computable sequence of relations $(\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}}, \sim, R_0, R_1, \dots)$ such that

(1) $\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}} \subseteq \mathcal{B}^{<\omega}$,

(2) $\sim$ is an equivalence relation on $\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}}$,

(3) $R_i \subseteq (B^{<\omega})^{a(i)}$ is closed under $\sim$ within $\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}}$,

and a function $f_{\mathcal{A}}^{\mathcal{B}}: \mathcal{D}om_{\mathcal{A}}^{\mathcal{B}} \to \mathcal{A}$ which induces an isomorphism:

$$(\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}}/\sim; R_0/\sim, R_1/\sim, \dots) \cong (A; P_0^{\mathcal{A}}, P_1^{\mathcal{A}}, \dots).$$

This is equivalent to Σ-reducibility without parameters.

Definition

$\mathcal{A}$ and $\mathcal{B}$ are *effectively bi-interpretable* if there are effective interpretations of each in the other, and u.r.i. computable isomorphisms $\mathcal{D}om_{\mathcal{A}}^{(\mathcal{D}om_{\mathcal{B}}^{\mathcal{A}})} \to \mathcal{A}$ and $\mathcal{D}om_{\mathcal{B}}^{(\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}})} \to \mathcal{B}$.

$$\begin{array}{ccc} & & \mathcal{B} \\ & & \cong \\ \mathcal{A} & \longrightarrow & \mathcal{D}om_{\mathcal{A}}^{\mathcal{B}} \end{array}$$

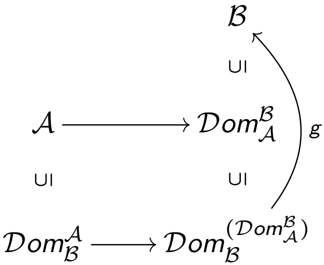
Definition

$\mathcal{A}$ and $\mathcal{B}$ are *effectively bi-interpretable* if there are effective interpretations of each in the other, and u.r.i. computable isomorphisms $\mathcal{D}om_{\mathcal{A}}^{(\mathcal{D}om_{\mathcal{B}}^{\mathcal{A}})} \to \mathcal{A}$ and $\mathcal{D}om_{\mathcal{B}}^{(\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}})} \to \mathcal{B}$.

$$
\begin{array}{ccc}
 & & \mathcal{B} \\
 & & \wr\!\!\cong \\
\mathcal{A} & \longrightarrow & \mathcal{D}om_{\mathcal{A}}^{\mathcal{B}} \\
\wr\!\!\cong & & \wr\!\!\cong \\
\mathcal{D}om_{\mathcal{B}}^{\mathcal{A}} & \longrightarrow & \mathcal{D}om_{\mathcal{B}}^{(\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}})}
\end{array}
$$

Definition

$\mathcal{A}$ and $\mathcal{B}$ are *effectively bi-interpretable* if there are effective interpretations of each in the other, and u.r.i. computable isomorphisms $\mathcal{D}om_{\mathcal{A}}^{(\mathcal{D}om_{\mathcal{B}}^{\mathcal{A}})} \to \mathcal{A}$ and $\mathcal{D}om_{\mathcal{B}}^{(\mathcal{D}om_{\mathcal{A}}^{\mathcal{B}})} \to \mathcal{B}$.



Definition

$\text{Iso}\,\mathcal{A}$ is the category of copies of $\mathcal{A}$ with domain ω. The morphisms are isomorphisms between copies of $\mathcal{A}$.

Recall: a functor F from $\text{Iso}\,\mathcal{A}$ to $\text{Iso}\,\mathcal{B}$

(1) assigns to each copy $\widehat{\mathcal{A}}$ in $\text{Iso}\,A$ a structure $F(\widehat{\mathcal{A}})$ in $\text{Iso}\,\mathcal{B}$,

(2) assigns to each isomorphism $f\colon \widehat{\mathcal{A}} \to \widetilde{\mathcal{A}}$ in $\text{Iso}\,\mathcal{A}$ an isomorphism $F(f)\colon F(\widehat{\mathcal{A}}) \to F(\widetilde{\mathcal{A}})$ in $\text{Iso}\,\mathcal{B}$.

Definition

F is *computable* if there are computable operators Φ and Φ_* such that

(1) for every $\widehat{\mathcal{A}} \in \text{Iso}\,\mathcal{A}$, $\Phi^{D(\widehat{\mathcal{A}})}$ is the atomic diagram of $F(\mathcal{A})$,

(2) for every isomorphism $f : \widehat{\mathcal{A}} \to \widetilde{\mathcal{A}}$, $F(f) = \Phi_*^{D(\widehat{\mathcal{A}}) \oplus f \oplus D(\widetilde{\mathcal{A}})}$.

Let $F, G\colon \operatorname{Iso}\mathcal{B} \to \operatorname{Iso}\mathcal{A}$ be computable functors.

Definition

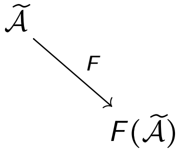
F is *effectively isomorphic* to G if there is a computable Turing functional Λ such that for any $\widetilde{\mathcal{B}} \in \operatorname{Iso}\mathcal{B}$, $\Lambda^{\widetilde{\mathcal{B}}}$ is an isomorphism from $F(\widetilde{\mathcal{B}})$ to $G(\widetilde{\mathcal{B}})$, and the following diagram commutes:

$\widetilde{\mathcal{A}}$

Let $F, G\colon \operatorname{Iso}\mathcal{B} \to \operatorname{Iso}\mathcal{A}$ be computable functors.

Definition

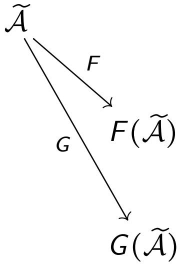
F is *effectively isomorphic* to G if there is a computable Turing functional Λ such that for any $\widetilde{\mathcal{B}} \in \operatorname{Iso}\mathcal{B}$, $\Lambda^{\widetilde{\mathcal{B}}}$ is an isomorphism from $F(\widetilde{\mathcal{B}})$ to $G(\widetilde{\mathcal{B}})$, and the following diagram commutes:



Let $F, G\colon \mathrm{Iso}\,\mathcal{B} \to \mathrm{Iso}\,\mathcal{A}$ be computable functors.

Definition

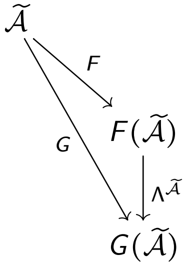
F is *effectively isomorphic* to G if there is a computable Turing functional Λ such that for any $\widetilde{\mathcal{B}} \in \mathrm{Iso}\,\mathcal{B}$, $\Lambda^{\widetilde{\mathcal{B}}}$ is an isomorphism from $F(\widetilde{\mathcal{B}})$ to $G(\widetilde{\mathcal{B}})$, and the following diagram commutes:



Let $F, G\colon \mathrm{Iso}\,\mathcal{B} \to \mathrm{Iso}\,\mathcal{A}$ be computable functors.

Definition

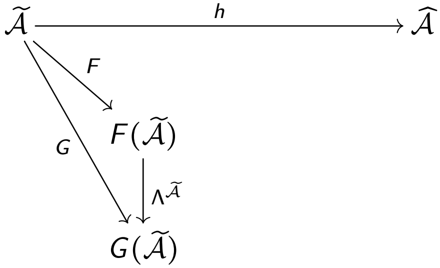
F is *effectively isomorphic* to G if there is a computable Turing functional Λ such that for any $\widetilde{\mathcal{B}} \in \mathrm{Iso}\,\mathcal{B}$, $\Lambda^{\widetilde{\mathcal{B}}}$ is an isomorphism from $F(\widetilde{\mathcal{B}})$ to $G(\widetilde{\mathcal{B}})$, and the following diagram commutes:



Let $F, G\colon \operatorname{Iso}\mathcal{B} \to \operatorname{Iso}\mathcal{A}$ be computable functors.

Definition

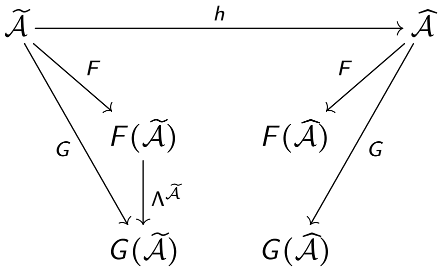
F is *effectively isomorphic* to G if there is a computable Turing functional Λ such that for any $\widetilde{\mathcal{B}} \in \operatorname{Iso}\mathcal{B}$, $\Lambda^{\widetilde{\mathcal{B}}}$ is an isomorphism from $F(\widetilde{\mathcal{B}})$ to $G(\widetilde{\mathcal{B}})$, and the following diagram commutes:



Let $F, G \colon \operatorname{Iso} \mathcal{B} \to \operatorname{Iso} \mathcal{A}$ be computable functors.

Definition

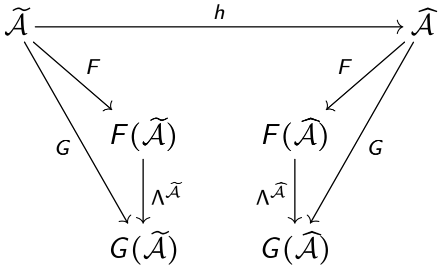
F is *effectively isomorphic* to G if there is a computable Turing functional Λ such that for any $\widetilde{\mathcal{B}} \in \operatorname{Iso} \mathcal{B}$, $\Lambda^{\widetilde{\mathcal{B}}}$ is an isomorphism from $F(\widetilde{\mathcal{B}})$ to $G(\widetilde{\mathcal{B}})$, and the following diagram commutes:



Let $F, G\colon \mathrm{Iso}\,\mathcal{B} \to \mathrm{Iso}\,\mathcal{A}$ be computable functors.

Definition

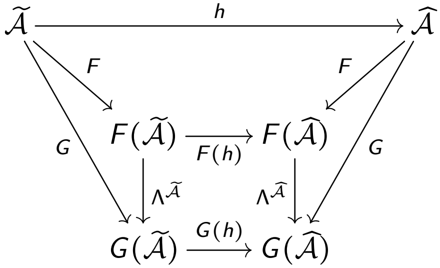
F is *effectively isomorphic* to G if there is a computable Turing functional Λ such that for any $\widetilde{\mathcal{B}} \in \mathrm{Iso}\,\mathcal{B}$, $\Lambda^{\widetilde{\mathcal{B}}}$ is an isomorphism from $F(\widetilde{\mathcal{B}})$ to $G(\widetilde{\mathcal{B}})$, and the following diagram commutes:



Let $F, G \colon \mathrm{Iso}\,\mathcal{B} \to \mathrm{Iso}\,\mathcal{A}$ be computable functors.

Definition

F is *effectively isomorphic* to G if there is a computable Turing functional Λ such that for any $\widetilde{\mathcal{B}} \in \mathrm{Iso}\,\mathcal{B}$, $\Lambda^{\widetilde{\mathcal{B}}}$ is an isomorphism from $F(\widetilde{\mathcal{B}})$ to $G(\widetilde{\mathcal{B}})$, and the following diagram commutes:



Definition

$\mathcal{A}$ and $\mathcal{B}$ are *computably bi-transformable* if there are computable functors $F\colon \mathrm{Iso}\,\mathcal{A} \to \mathrm{Iso}\,\mathcal{B}$ and $G\colon \mathrm{Iso}\,\mathcal{B} \to \mathrm{Iso}\,\mathcal{A}$ such that both $F \circ G\colon \mathrm{Iso}\,\mathcal{B} \to \mathrm{Iso}\,\mathcal{B}$ and $G \circ F\colon \mathrm{Iso}\,\mathcal{A} \to \mathrm{Iso}\,\mathcal{A}$ are effectively isomorphic to the identity functor.

So if $\widehat{\mathcal{B}}$ is a copy of $\mathcal{B}$, then $F(G(\widehat{\mathcal{B}})) \cong \widehat{\mathcal{B}}$ and the isomorphism can be computed uniformly in $\widehat{\mathcal{B}}$.

Theorem (HT, Melnikov, Miller, Montalbán)

$\mathcal{A}$ and $\mathcal{B}$ are effectively bi-interpretable

$$\Updownarrow$$

$\mathcal{A}$ and $\mathcal{B}$ are computably bi-transformable.

$$\Downarrow$$

$\mathcal{A}$ and $\mathcal{B}$ have the same automorphism group.

Theorem

If two structures are effectively bi-interpretable, then they have the same degree spectrum, Scott rank, computable dimension, and much more.

Definition

A class $\mathcal{C}$ is universal if every graph is effectively bi-interpretable with a structure in $\mathcal{C}$.

Theorem (Hirschfeldt, Khoussainov, Shore, Slinko)

The following classes are universal:

1. *undirected graphs,*
2. *partial orderings, and*
3. *lattices,*

and, after naming finitely many constants,

1. *integral domains,*
2. *commutative semigroups, and*
3. *2-step nilpotent groups.*

Theorem (Miller, Park, Poonen, Schoutens, Shlapentokh)

Fields of characteristic zero are universal.

The class of trees was Borel complete, but it is easy to see that it is not universal: There are simple structures, e.g. the integers $(\mathbb{Z}, s)$ with the successor operator, which are not effectively bi-interpretable with any tree.

The automorphism group of $(\mathbb{Z}, s)$ is $\mathbb{Z}$, but this is not the automorphism group of any tree. For example, every tree with a non-trivial automorphism has an automorphism of order two. So $(\mathbb{Z}, s)$ is not effectively bi-interpretable with any tree.

This argument is unsatisfying. What we really want is some computability-theoretic property that cannot be realised within the class of trees.

A natural candidate would be to find a degree spectrum which is not the degree spectrum of a tree.

Question

Is every degree spectrum the degree spectrum of a tree?

Question

Do $\mathcal{A}$ and $\mathcal{T}_\infty(\mathcal{A})$ always have the same degree spectrum?

If the answer was yes, then every degree spectrum would be the degree spectrum of a tree.

Theorem (HT and Montalbán)

There is a structure $\mathcal{A}$ with no computable copies but for which $\mathcal{T}(\mathcal{A})$ has a computable copy.

Think of this as saying that a structure cannot be computably recovered from its back-and-forth information.

Paths through $\mathcal{T}(\mathcal{A})$ correspond to substructures of $\mathcal{A}$. We could recover $\mathcal{A}$ from $\mathcal{T}(\mathcal{A})$ together with a sufficiently generic path.

$\mathcal{A}$ and $\mathcal{T}_\infty(\mathcal{A})$ do not always have the same degree spectrum. We have *not* shown that there is a degree spectrum which is not the degree spectrum of a tree, since there might be some other tree T with $\mathrm{DgSp}(\mathcal{A}) = \mathrm{DgSp}(T)$.

Using Marker extensions, we get a strengthening.

Corollary

For each computable ordinal α, there is a structure $\mathcal{A}$ with no Δ^0_α-computable copies but for which $\mathcal{T}(\mathcal{A})$ has a computable copy.

Knight, Soskova, and Vatev have independently shown that there are no $\mathcal{L}_{\omega_1\omega}$ formulas that uniformly (in the choice of signature) interpret a structure $\mathcal{A}$ in the tree $\mathcal{T}(\mathcal{A})$. This follows from our theorem as if $\mathcal{A}$ is interpreted in $\mathcal{T}(\mathcal{A})$ using Σ^X_α formulas, then any copy of $\mathcal{T}(\mathcal{A})$ $\Delta^{0,X}_\alpha$-computes a copy of $\mathcal{A}$.

Suppose that a set X is c.e.-coded by $\mathcal{A}$.

Theorem (Knight)

A set X is c.e.-coded in a structure $\mathcal{A}$ if and only if X is enumeration reducible to the $\exists$-type of some tuple from $\mathcal{A}$, where the $\exists$-type of a tuple $\bar{a}$ in $\mathcal{A}$ is given by

$$\exists\,\mathrm{tp}_{\mathcal{A}}(\bar{a}) = \{\ulcorner\varphi\urcorner : \varphi(\bar{x}) \text{ is an } \exists\text{-formula such that } \mathcal{A} \models \varphi(\bar{a})\}.$$

So X is enumeration reducible to the $\exists$-type of a tuple from $\mathcal{A}$, and so it is c.e.-coded by $\mathcal{T}_\infty(\mathcal{A})$.

Let $\{\Theta_e : e \in \omega\}$ be the standard computable enumeration of all enumeration operators.

A natural way for a family $\mathcal{F}$ to be (uniformly) coded by a structure is for there to be a uniformly computable list of Σ_1^c formulas $\varphi_\ell(\bar{y})$ such that

$$\mathcal{F} = \{\Theta_\ell(\exists\text{-tp}_{\mathcal{A}}(\bar{b})) : \ell \in \omega,\ \bar{b} \in A^{<\omega},\ \mathcal{A} \models \varphi_\ell(\bar{b})\}.$$

Theorem (Montalbán)

If a family $\mathcal{F}$ is uniformly c.e.-coded in a structure $\mathcal{A}$, then there exist a uniformly computable list of Σ_3^c formulas $\varphi_\ell(\bar{y})$ such that

$$\mathcal{F} = \{\Theta_\ell(\exists\text{-}tp_{\mathcal{A}}(\bar{b})) : \ell \in \omega,\ \bar{b} \in A^{<\omega},\ \mathcal{A} \models \varphi_\ell(\bar{b})\}.$$

Can Σ_3^c be strengthened to Σ_1^c?

Let $\mathcal{A}$ be the structure with no computable copies but such that $\mathcal{T}_\infty(\mathcal{A})$ has a computable copy.

Let $\mathcal{F}$ be the Slaman-Wehner family, which has an enumeration in every non-computable degree, but no computable enumeration.

Then $\mathcal{F}$ is c.e.-coded in $\mathcal{A}$, but not in $\mathcal{T}_\infty(\mathcal{A})$.

If $\mathcal{F}$ was coded by $\mathcal{A}$ using a Σ_1^c formula $\varphi_\ell(\bar{y})$ such that

$$\mathcal{F} = \{\Theta_\ell(\exists\text{-tp}_{\mathcal{A}}(\bar{b})) : \ell \in \omega,\ \bar{b} \in A^{<\omega},\ \mathcal{A} \models \varphi_\ell(\bar{b})\}$$

then $\mathcal{F}$ would be c.e. coded by $\mathcal{T}_\infty(\mathcal{A})$.

Theorem (HT and Montalbán)

There is a structure $\mathcal{A}$ and a family $\mathcal{F}$ uniformly c.e.-coded in structure $\mathcal{A}$, such that there is no uniformly computable list of Σ_1^c formulas $\varphi_\ell(\bar{y})$ such that

$$\mathcal{F} = \{\Theta_\ell(\exists\text{-}tp_{\mathcal{A}}(\bar{b})) : \ell \in \omega,\ \bar{b} \in A^{<\omega},\ \mathcal{A} \models \varphi_\ell(\bar{b})\}.$$

To make everything work nicely, we have to strengthen the way we code.

Definition

A family $\mathcal{F}$ is *functorially c.e.-coded* if also there is a computable operator that, given two copies $\widehat{\mathcal{A}}$ and $\widetilde{\mathcal{A}}$ of $\mathcal{A}$ and an isomorphism f between them, produces a permutation of ω matching the columns of $W^{D(\widehat{\mathcal{A}})}$ and $W^{D(\widetilde{\mathcal{A}})}$ in a *functorial way*, meaning that this operator maps the identity to the identity and preserves composition of isomorphisms.

Proposition (HT and Montalbán)

A family $\mathcal{F}$ is functorially c.e.-coded in a structure $\mathcal{A}$ if and only if there exists a uniformly computable list of Σ^c_1 formulas $\varphi_\ell(\bar{x}, \bar{y})$ such that

$$\mathcal{F} = \{\Theta_\ell(\exists\text{-}tp_{\mathcal{A}}(\bar{b})) : \ell \in \omega,\ \bar{b} \in A^{<\omega},\ \mathcal{A} \models \varphi_\ell(\bar{b})\}.$$

There is no reason to stop at sets and families of sets. Families of families of sets and so on often arise naturally, e.g. when dealing with Marker extensions. We can index the depth of the families by ordinals, or even have the families be non-well-founded.

When talking about families of families of families... what we are really talking about is (replicated) labeled trees. By replicated we mean that each child is replicated infinitely many times.

$\mathcal{T}_\infty(\mathcal{A})$ is the maximal replicated labeled tree coded functorially by $\mathcal{A}$.

Theorem (HT and Montalbán)

Given a structure $\mathcal{A}$, $\mathcal{T}_\infty(\mathcal{A})$ is functorially reducible to $\mathcal{A}$ and every other replicated labeled tree that is functorially reducible to $\mathcal{A}$ is functorially reducible to $\mathcal{T}_\infty(\mathcal{A})$.

A structure can code information in ways that are not captured by its back-and-forth structure.

Conjecture

There is a degree spectrum which is not the degree spectrum of a tree.

I believe this should be true because I feel that:

1. If every degree spectrum is the degree spectrum of a tree, this will be witnessed by a reasonably nice transformation Φ from graphs to trees such that for every graph G, G and $\Phi(G)$ have the same degree spectrum.
2. $\mathcal{T}_\infty$ has a special place as the "strongest" transformation from graphs to trees.
3. $\mathcal{A}$ can code things that $\mathcal{T}_\infty(\mathcal{A})$ does not.

Thanks!