https://doi.org/10.1142/9789811271151_0032

Chapter 32

Computable Isomorphism Problem*

Valentina Harizanov
Department of Mathematics,
George Washington University,
Washington, DC, USA
harizanv@gwu.edu

Isomorphism is one of the most important equivalence relations between two structures of the same kind. We present some recent results on the maximal complexity of the isomorphism problem for computable structures from a class K closed under isomorphism. Formally, the computable isomorphism problem for K is the set of pairs of computable indices for computable structures in K, which are isomorphic. For many familiar classes, the computable isomorphism problem is complete at some level of arithmetical hierarchy, such as for vector spaces over a fixed infinite computable field or for algebraically closed fields of a fixed characteristic, as established by Calvert. On the other hand, there are classes of structures with Σ^1_1-complete computable isomorphism problem, which is of maximal complexity. These classes contain isomorphic computable structures that are not even hyperarithmetically isomorphic. For example, Friedman and Stanley established that fields of a fixed characteristic have Σ^1_1-complete computable isomorphism problem. Recent examples of such classes include 2-step nilpotent groups, distributive lattices, and nilpotent rings. The method we use to establish Σ^1_1-completeness is based on a uniform effective interpretation of computable structures from a certain class into computable structures in K.

*This chapter is dedicated to the memory of Zbyszek Oziewicz, a great friend with an inquisite mind and passion for mathematics and science.

1. Structures, Language, and Isomorphisms

While early computable structure theory focused on decidability or undecidability of various important mathematical problems, modern computable structure theory delves into fine computability-theoretic classification of properties in mathematical structures. We apply sophisticated and often unique methods of computability theory, including various encoding techniques and syntactic descriptions using computable language, to study various problems on mathematical structures. Isomorphism is one of the most important relations between mathematical structures for the same language.

A *structure* $\mathcal{A}$ consists of a nonempty set A, called the domain or the universe of the structure, with certain relations and operations (functions) on the domain, and possibly constants, which are specific elements of the domain. Every relation and every function has assigned length or arity. The names for relations, functions, and constants are called relation symbols, function symbols, and constant symbols, and they form the *language* L of the structure. The language can be finite or infinite. First-order formulas are obtained using the symbols in L, variables for elements, equality, negation ($\neg$), disjunction ($\vee$), conjunction ($\wedge$), existential quantifier ($\exists$) and universal quantifier ($\forall$) over element variables. Formulas in which all variables are bound by quantifiers are called *sentences*. A sentence is either true or false in a structure for the same language according to natural interpretation of symbols. A *theory* is a consistent set of sentences.

For example, $(A, \cdot, e)$ is a group if $\cdot$ is a binary operation and e is a constant so that the group axioms are satisfied, i.e., sentences expressing that $\cdot$ is associative, e is the identity element, and every element has an inverse:

$$(\forall x)(\forall y)(\forall z)[(x \cdot y) \cdot z = x \cdot (y \cdot z)],$$
$$(\forall x)[x \cdot e = e \cdot x = x],$$
$$(\forall x)(\exists y)[x \cdot y = y \cdot x = e].$$

Let $\mathcal{A}$ and $\mathcal{B}$ be structures for the same language L, with domains A and B, respectively. An *isomorphism* from $\mathcal{A}$ to $\mathcal{B}$ is

a bijection (one-to-one and onto function) $f : A \to B$ such that f "preserves structure". Say, $*$ is a binary operation symbol, R is a binary relation symbol, and c is a constant symbol. Then for any $a, d \in A$, we have

$$f(a *^{\mathcal{A}} d) = f(a) *^{\mathcal{B}} f(d),$$
$$R^{\mathcal{A}}(a, d) \Leftrightarrow R^{\mathcal{B}}(f(a), f(d)),$$
$$f(c^{\mathcal{A}}) = c^{\mathcal{B}}.$$

We have similar definitions for n-ary functions and relations.

A sequence of variables displayed after a formula contains a subsequence of all free variables occurring in the formula. If f is an isomorphism from $\mathcal{A}$ to $\mathcal{B}$, then for every first-order formula $\theta(x_1, \ldots, x_n)$ and every n-tuple $a_1, \ldots, a_n \in A$, we have that $a_1, \ldots, a_n$ satisfy θ in $\mathcal{A}$ if and only if $f(a_1), \ldots, f(a_n)$ satisfy θ in $\mathcal{B}$. Hence, if two structures are isomorphic, then they satisfy the same first-order sentences, that is, they have the same first-order theory. The converse is not true for infinite structures. By $\mathcal{A} \cong \mathcal{B}$ we denote that $\mathcal{A}$ and $\mathcal{B}$ are isomorphic. An automorphism of $\mathcal{A}$ is an isomorphism between $\mathcal{A}$ and itself. For more on isomorphism of structures, see Ref. 7.

2. Computable Sets and Structures: Turing Degrees

We will focus on countable structures, that is, ones with countable domains. If infinite, we can identify them with the set N of natural numbers. A set A of natural numbers is *computable* if there is a decision procedure for identifying its elements. That is, there is a Turing machine that on every input a halts and outputs "yes" if $a \in A$ and outputs "no" if $a \notin A$.

The *characteristic function* of A is defined as

$$c_A(a) = \begin{cases} 1, & \text{if } a \in A; \\ 0, & \text{if } a \notin A. \end{cases}$$

A function $f : N^n \to N$, where $n \geq 1$, is *computable* if there is a Turing machine that on every input $a_1, \ldots, a_n$ halts and outputs its value $f(a_1, \ldots, a_n)$. Hence, a set A is computable if the

characteristic function of A is computable. Clearly, the complement of a computable set is computable. Sets are identified with unary relations. Computable n-ary relations for $n > 1$ are defined similarly. For example, the set of prime numbers is computable. All finite sets are computable. All co-finite sets are computable. Addition and multiplication on natural numbers are computable functions (operations). It can be shown that a function $f : N^2 \to N$ defined by $f(a, b) = \frac{1}{2}(a^2 + 2ab + b^2 + 3a + b)$ is a computable bijection. Hence, it algorithmically codes pairs of natural numbers by natural numbers. Decidable problems are encoded by computable sets.

Since each Turing machine is a finite list of instructions, Turing machines can be algorithmically enumerated, even without repetition (i.e., one-to-one), as

$$M_0, M_1, M_2, \ldots$$

Hence, there are countably many computable functions and countably many computable sets. Since there are uncountably many subsets of natural numbers, there are uncountably many noncomputable sets.

We consider structures for computable languages. A computable language is a countable language with algorithmically presented set of symbols and their arities. A structure $\mathcal{A}$ with domain A for a computable language L is a *computable* structure if A is a computable set and its operations, relations, and constants are uniformly computable. For a finite language L, we can just say that its operations and relations are computable. For example, a group $(G, *)$ is *computable* if G is a computable set and $*$ is a computable operation. A structure (A, R) with a relation R is *computable* if A is a computable set and R is a computable relation. Examples of familiar computable structures include the ordered sets of natural numbers, $(N, <)$, the ordered sets of integers, $(Z, <)$, the ordered sets of rational numbers, $(Q, <)$, the additive group of integers, $(Z, +)$, and the field of rational numbers, $(Q, +, \cdot)$.

An ordinal is *computable* if it is the order type of some computable well-ordering. Computable ordinals form a countable initial

segment of the ordinals. There exists a least non-computable ordinal, and it is easy to see that this must be a countable ordinal, i.e., the order type of a countable well-ordering. We call this ordinal ω_1^{CK}, in analogy to ω_1, the least uncountable ordinal. The CK stands for Alonzo Church and Stephen Kleene.

The standard model of arithmetic, $\mathcal{N} = (N, +, \cdot, S, 0)$, the natural numbers with addition, multiplication, successor function, and zero, is computable. In particular, $\mathcal{N}$ satisfies the axioms of Peano Arithmetic. A model not isomorphic to $\mathcal{N}$, which satisfies all axioms of Peano Arithmetic, is called a *nonstandard model* of Peano Arithmetic. Nonstandard models have numbers that are larger than any natural number. Skolem constructed a countable nonstandard model of Peano Arithmetic. Tennenbaum showed that if $\mathcal{M}$ is a nonstandard model of Peano Arithmetic, then $\mathcal{M}$ is not computable (see Ref. 1).

Turing machines augmented with oracles were introduced by Turing. An oracle is a set that supplies membership information on demand and the corresponding Turing machine computations perform finitely many non-mechanical steps since the oracle may not be computable. Turing degrees of complexity were introduced by Post. Let A and B be subsets of N. We say that A is Turing computable from B if A can be computed by a Turing machine with oracle B. We denote this by

$$A \leq_T B \quad \text{or} \quad \deg(A) \leq \deg(B).$$

For example, if B is the complement of A, $B = \overline{A}$, we can compute whether $n \in A$ by asking B whether $n \in B$. Hence, $A \leq_T \overline{A}$. Sets A and B have the same Turing degree, in symbols,

$$A \equiv_T B \quad \text{or} \quad \deg(A) = \deg(B),$$

if $A \leq_T B$ and $B \leq_T A$. Thus, A and its complement $\overline{A}$ have the same Turing degree. We say that A has a strictly smaller Turing degree than B, in symbols $A <_T B$ or $\deg(A) < \deg(B)$, if A can be computed by a Turing machine with oracle B, but not vice versa. Two Turing degrees may be incomparable. There are uncountably

many Turing degrees since each Turing degree has only countably many sets.

We can show that two Turing degrees have the least upper bound, i.e., supremum:

$$\sup\{\deg(A), \deg(B)\} = \deg(A \oplus B) \text{ where}$$
$$A \oplus B = \{2k : k \in A\} \cup \{2k+1 : k \in B\}.$$

For example, we can show that $A \leq_T A \oplus B$: for a given a we compute $2a$ and then ask oracle $A \oplus B$ whether $2a$ belongs to it. The greatest lower bound, i.e., infimum, may not exist. Hence, Turing degrees are partially ordered, forming an upper semi-lattice. Computable sets have Turing degree $\mathbf{0}$, which is the least Turing degree. For more on Turing degrees, see Refs. 24 and 23.

In general, when measuring complexity of a structure $\mathcal{A}$, we identify $\mathcal{A}$ with its *atomic diagram* $D(\mathcal{A})$, the set of all atomic and negations of atomic sentences allowing additional constants for elements of the domain. More precisely, if L is the language of $\mathcal{A}$, then L_A is the language L expanded by adding a constant symbol for every $a \in A$, and $\mathcal{A}_A = (\mathcal{A}, a)_{a \in A}$ is the corresponding expansion of $\mathcal{A}$ to L_A. For example,

$$a_i * a_j = a_k,\ a_i * a_j \neq a_l,\ a_m < a_n, \ldots$$

are sentences in the atomic diagrams of the corresponding structures.

Formulas are effectively encoded with natural numbers, so $D(\mathcal{A})$ has a Turing degree, which is also considered to be the Turing degree of $\mathcal{A}$. A structure $\mathcal{A}$ is computable if its Turing degree is $\mathbf{0}$. Hence, $\mathcal{A}$ is computable if the characteristic function of $D(\mathcal{A})$ is computable. For more on computable structures, see Ref. 17.

3. Computably Enumerable Sets: The Halting Set

A set A of natural numbers is *computably enumerable* (abbreviated by c.e.) if A is empty or there is a computable unary function $f : N \to N$ such that A is the range of f. Hence, a nonempty c.e. set A

can be enumerated as

$$a_0 = f(0), a_1 = f(1), a_2 = f(2), \ldots$$

Clearly, every computable set is c.e. since a decision algorithm for identifying its elements can be transformed into an enumeration algorithm. For an infinite c.e. set, a computable enumeration $a_0, a_1, a_2, \ldots$ can be modified by crossing repetitions to be one-to-one, i.e., without repetitions. However, a computable enumeration may not be in the increasing (natural) order, so we cannot know at any stage in advance whether a number that has not been enumerated in A will later be enumerated or not. Computable infinite sets are exactly those c.e. sets for which there is a computable enumeration in the increasing order:

$$a_0 < a_1 < a_2 < \cdots$$

It can be shown that a set A is c.e. if and only if there is a computable binary relation R such that for every a,

$$a \in A \Leftrightarrow (\exists x) R(a, x).$$

A set A is c.e. if and only if A is the domain of some partial computable function.

For each Turing machine M_e (in the enumeration in the previous section), we will denote the unary partial function it computes by φ_e, and for $n > 1$, an n-ary partial function it computes by $\varphi_e^{(n)}$. Hence,

$$\varphi_0, \varphi_1, \varphi_2, \ldots$$

is a computable enumeration of all unary partial computable functions. For each partial computable function ψ there are infinitely many indices e such that $\psi = \varphi_e$. Two partial functions are equal if they have the same domains and for each input in the domain they output the same value. For φ_e, we denote its domain by W_e. Hence, it follows that a set A is c.e. if and only if $A = W_e$ for some e. Hence,

$$W_0, W_1, W_2, \ldots$$

is a computable enumeration of all c.e. sets.

For an oracle X, we can enumerate all partial X-computable unary functions:

$$\varphi_0^X, \varphi_1^X, \varphi_2^X, \ldots,$$

as well as the corresponding X-c.e. sets.

There are c.e. sets with complements that are not c.e. These are exactly the noncomputable c.e. sets. For example, the halting set H is c.e., but not computable. The set H consists of all inputs e on which the Turing machine with index e halts. That is,

$$\begin{aligned} H = \{e : e \in W_e\} &= \{e : \varphi_e(e) \text{ halts (is defined)}\} \\ &= \{e : M_e \text{ halts on input } e\}. \end{aligned}$$

The set H is c.e. because it can be algorithmically enumerated by the procedure that simultaneously runs

$$\varphi_0(0), \varphi_1(1), \ldots, \varphi_e(e), \ldots$$

and enumerates those e for which $\varphi_e(e)$ halts (converges), as soon as the halting occurs. Here, simultaneously means that at each step we add a new Turing machine and also run all activated machines for an additional computational step. If the complement $\overline{H}$ were c.e., then for some e_0, we would have $\overline{H} = W_{e_0}$. Then

$$e_0 \in H \Leftrightarrow e_0 \in W_{e_0} \quad \Leftrightarrow \quad e_0 \in \overline{H},$$

which is a contradiction.

We also denote H by $\varnothing'$ and call it the first Turing jump of $\varnothing$. For a set X of natural numbers, the first jump of X is

$$X' = \{e : \varphi_e^X(e) \text{ halts}\}.$$

The second jump of $\varnothing$ is defined as $H' = \varnothing'' = \{e : \varphi_e^{\varnothing'}(e) \text{ halts}\}$. This process can be iterated to obtain iterated halting sets $\varnothing^{(n)}$ for any natural number $n \geq 1$. Turing jumps can also be iterated through the computable ordinals.

An example of a non-computable structure is the linear order $\mathcal{A} = (N, \prec)$ where

$2n \prec 2n+1$ if n belongs to the halting set H, while

$2n+1 \prec 2n$ if n does not belong to H.

If this order were computable, the halting set H would be computable, which is a contradiction. Instead of H we can use any other non-computable set. Note that $\mathcal{A}$ is isomorphic to the ordered set of natural numbers, $(N, <)$. For more on c.e. sets and iterated halting sets, see Refs. 23 and 24.

If a structure $\mathcal{A}$ is computable, then the characteristic function of its atomic diagram $D(\mathcal{A})$ is computable, hence it is, φ_e for some e. We call e a *computable index* for $\mathcal{A}$ and also denote $\mathcal{A}$ by $\mathcal{A}_e$. Clearly, not all numbers e are indices for structures.

Definition 1. Let $\mathbb{K}$ be a family of structures closed under isomorphism. The *index set* of $\mathbb{K}$, denoted by $I(\mathbb{K})$, is the set of all computable indices for members of $\mathbb{K}$.

4. Arithmetical Hierarchy

We have the following classification of first-order formulas. The upper subscript 0 signifies that the quantification is only over variables that stand for individual elements. A formula is a $\Sigma_0^0 = \Pi_0^0$ formula if it is quantifier-free. For $n > 0$, a formula is a Σ_n^0 (Π_n^0, respectively) formula if it is equivalent to a formula in the prenex normal form which begins with an existential (universal, respectively) quantifier and has n alternations of quantifiers.

We also have a similar classification of sets of natural numbers where Π_n^0 and Σ_n^0 sets (or relations) are levels in the *arithmetical hierarchy* obtained from computable relations by applying existential and universal quantifiers. Equivalently, a set A is arithmetical if it can be obtained from a computable relation by finitely many projections and complementations.

A set A is $\Sigma_0^0 = \Pi_0^0$ if it is computable. Let $n > 0$. A set A is Σ_n^0 if there is a computable $(n+1)$-ary relation R such that for every $a \in N$,

$$a \in A \Leftrightarrow (\exists x_1)(\forall x_2) \cdots (Qx_n)R(a, x_1, x_2, \ldots, x_n)$$

where Q is $\exists$ if n is an odd number, and Q is $\forall$ if n is an even number. For example, a binary relation $E(i, j)$ is Σ_3^0 if it is equivalent to

$$\exists x \forall y \exists z C(i, j, x, y, z), \text{ where } C \text{ is a computable relation.}$$

Π^0_n sets are defined similarly starting with the universal quantifier. The existential quantifier here can be viewed as a projection. The negation corresponds to complementation, and the universal quantifier $\forall$ can be expressed using the existential quantifier and negation, as $\neg\exists\neg$. The complement of a Σ^0_n set is a Π^0_n set and vice versa. It follows that Σ^0_1 sets coincide with c.e. sets and Π^0_1 sets coincide with co-c.e. sets.

This hierarchy of sets of natural numbers is a proper hierarchy since Σ^0_n sets are strictly contained in Σ^0_{n+1} sets, and Π^0_n sets are strictly contained in Π^0_{n+1} sets. Clearly, every Σ^0_n set is also Π^0_{n+1}, and every Π^0_n set is also Σ^0_{n+1}. It then follows that the union of all Σ^0_n sets is the same as the union of all Π^0_n sets. A set is called *arithmetical* if it belongs to this union; i.e., it is Π^0_m for some m (or Σ^0_l for some l). We say that a set is Δ^0_n if it is both Σ^0_n and Π^0_n. We have that Δ^0_n sets are strictly contained in Π^0_n (or Σ^0_n) sets. It can be shown that Δ^0_1 sets are exactly the computable sets.

It can be proved that a relation is arithmetical if and only if it is definable in the standard model of arithmetic $\mathcal{N}$ by a first-order formula in the language of arithmetic, $L = \{+, \cdot, S, 0\}$. We say that a set C of natural numbers is definable in $\mathcal{N}$ if there is a formula $\theta(x)$ such that C consists of all natural numbers n that satisfy θ in $\mathcal{N}$. It is not hard to show that a relation definable in $\mathcal{N}$ by a quantifier-free formula is computable. It follows that a relation definable in $\mathcal{N}$ by a Σ^0_n formula (Π^0_n formula, respectively) is a Σ^0_n relation (Π^0_n relation, respectively). To prove his incompleteness theorem, Gödel established that all computable relations are definable in $\mathcal{N}$, and hence all arithmetical relations are definable in $\mathcal{N}$. For any computable relation there are two natural defining formulas: one with a block of existential quantifiers followed by a formula with only bounded quantifiers, $\forall x < y$ and $\exists x < y$, and the other one with a block of universal quantifiers followed by a formula with only bounded quantifiers. A block of existential (universal) quantifiers can be replaced by a single existential (universal) quantifier by coding tuples of natural numbers by a single natural number. It follows from a lemma by Y. Matiyasevich in his proof of Hilbert's Tenth Problem that bounded quantifiers can be eliminated from the above

formulas, so a computable set is definable in $\mathcal{N}$ both by a Σ_1^0 and a Π_1^0 formula (see Appendix A.7 in Ref. 1). Moreover, a set is Σ_n^0 (Π_n^0, respectively) if and only if it can be defined in $\mathcal{N}$ by a Σ_n^0 (Π_n^0, respectively) formula.

Furthermore, a set is Σ_n^0 for $n \geq 1$ if and only if it is c.e. relative to the iterated halting set $H^{(n-1)}$. Here, $H^{(0)} =_{def} \varnothing$. A set is Δ_n^0 if and only if it is computable in $H^{(n-1)}$. For more on arithmetical hierarchy, see Refs. 23 and 24.

5. Index Sets of Structures and the Isomorphism Problem

Given a set complexity class $\mathfrak{C}$, such as Π_2^0 or Σ_3^0, we say that a set X is $\mathfrak{C}$-*complete* if X is in $\mathfrak{C}$, and X is $\mathfrak{C}$-hard in the sense that every set Y in $\mathfrak{C}$ can be reduced to X. The reduction means that there is a computable function $f : N \to N$ such that for every $n \in N$, we have

$$n \in Y \qquad \Leftrightarrow \qquad f(n) \in X.$$

The reduction function f does not have to be one–one, it can be many–one, so the above reduction is called m-reduction. For example, we can show that the set $H^{(n)}$ is Σ_n^0-complete.

Recall that the *index set* of $\mathbb{K}$, denoted by $I(\mathbb{K})$, is the set of all computable indices for members of $\mathbb{K}$.

Theorem 1 (Ref. 11). *For the following classes $\mathbb{K}$, the index set $I(\mathbb{K})$ is Π_2^0: linear orderings, Boolean algebras, equivalence structures, abelian p-groups (for prime numbers p), vector spaces over the field of rational numbers $\mathbb{Q}$.*

An equivalence structure $(A, \sim)$ consists of a set A with a binary relation $\sim$ that is an equivalence relation, i.e., reflexive, symmetric, and transitive.

An abelian p-group, where p is a prime number, is an abelian group in which every non-zero element has order p^m for some $m \geq 1$. Examples of such groups are $\mathbb{Z}(p^n)$, the cyclic group of order p^n, and $\mathbb{Z}(p^\infty)$, the quasicyclic p-group, as well as their direct sums. We can think of the group $\mathbb{Z}(p^\infty)$ as the set of rationals in $[0, 1)$ of the

form $\frac{i}{p^k}$ with addition modulo 1. Countable abelian p-groups were classified up to isomorphism by Ulm.

The *elementary diagram* of $\mathcal{A}$, denoted by $D^c(\mathcal{A})$, is the set of all first-order sentences of L_A that are true in $\mathcal{A}_A$. Henkin's construction of a model for a complete and decidable theory is effective and produces a structure $\mathcal{A}$ with a computable domain such that the elementary diagram of $\mathcal{A}$ is computable. A structure $\mathcal{A}$ is called *decidable* if its elementary diagram $D^c(\mathcal{A})$ is computable. Clearly, every decidable structure is computable. Not every computable structure is decidable. For example, the standard model of arithmetic $\mathcal{N} = (N, +, \cdot, S, 0)$ is computable but not decidable. For more on computable and decidable structures, see Ref. 17.

Theorem 2 (Ref. 12). *The index set of decidable structures is* Σ^0_3*-complete.*

Definition 2. The *computable isomorphism problem* for $\mathbb{K}$ is the set $E(\mathbb{K})$ of pairs $(i, j) \in I(\mathbb{K}) \times I(\mathbb{K})$ such that $\mathcal{A}_i \cong \mathcal{A}_j$.

Theorem 3 (Ref. 2). *For* $\mathbb{Q}$*-vector spaces, where* $\mathbb{Q}$ *is the field of rational numbers,* $E(\mathbb{K})$ *is* Π^0_3*-complete.*

In this result, $\mathbb{Q}$ can be replaced by any infinite computable field.

Theorem 4 (Ref. 4). *For equivalence structures,* $E(\mathbb{K})$ *is* Π^0_4*-complete.*

A field $(F, +, \cdot)$ is algebraically closed if for every $n \geq 1$, we have for every $x_0, x_1, \ldots, x_n$:

$$(\exists y)(x_n \cdot y^n + x_{n-1} \cdot y^{n-1} + \cdots + x_1 \cdot y + x_0 = 0) \vee x_n = 0.$$

Each field has a unique characteristic, either some prime p or 0. Let $\mathcal{F} = (F, +, \cdot)$ be a field. Then we use the notation:

$$p1 =_{def} \underbrace{1 + \cdots + 1}, \text{ added } p \text{ times.}$$

A field $\mathcal{F}$ has characteristic p if we have $p1 = 0$. A field $\mathcal{F}$ has characteristic 0 if for all primes p, we have $p1 \neq 0$.

Theorem 5 (Ref. 2). *For algebraically closed fields of a fixed characteristic, $E(\mathbb{K})$ is Π^0_3-complete.*

A group is torsion-free if all elements have infinite order, i.e., for every $n \geq 1$, we have

$$x \neq 0 \qquad \Rightarrow \qquad nx \neq 0.$$

Torsion-free abelian groups of finite rank n can be viewed as subgroups of $\mathbb{Q}^n$.

Theorem 6 (Ref. 3). *For torsion-free abelian groups of a fixed finite rank, $E(\mathbb{K})$ is Σ^0_3-complete.*

For more on index sets of structures, see Refs. 5,6,11,13,16 and 21, and on the computable isomorphism problem with complexity in the arithmetical hierarchy, see Refs. 11 and 14.

6. Computable Formulas

Scott Isomorphism Theorem states that for any countable structure $\mathcal{A}$, there is a $L_{\omega_1\omega}$ sentence the countable models of which are exactly the isomorphic copies of $\mathcal{A}$. While $L_{\omega_1\omega}$ formulas allow arbitrary countable disjunctions and conjunctions, computable formulas allow only c.e. conjunctions and disjunctions. Computable structures in many familiar classes $\mathbb{K}$ can be described by a computable (infinitary) formula. We can use computable ordinals to classify computable infinitary formulas in a fixed language.

The computable Σ_0 and Π_0 formulas are the finitary quantifier-free formulas.

A *computable* Π_α *formula*, for a computable ordinal α, is a c.e. conjunction of formulas

$$\forall \overline{u} \psi(\overline{x}, \overline{u}),$$

where ψ is a computable Σ_β formula for some ordinal $\beta < \alpha$.

A *computable* Σ_β *formula* is a c.e. disjunction of formulas

$$\exists \overline{v} \theta(\overline{y}, \overline{v}),$$

where θ is a computable Π_γ formula for some $\gamma < \beta$.

Thus, a computable Σ_1 formula is of the form

$$\bigvee_{i \in I} \exists \overline{v}_i \theta_i(\overline{x}, \overline{v}_i),$$

where the index set I is c.e. and θ_i's are quantifier-free formulas.

For example, a computable Π_2 sentence describing an abelian p-group is the conjunction of the axioms for abelian groups and

$$\forall x \bigvee_{m} (\underbrace{x + x + \cdots + x}_{p^m \text{ times}} = 0).$$

Computable structures satisfying the same computable sentences are isomorphic. For more on computable formulas, see Ref. 1.

7. Hyperarithmetical Hierarchy

Arithmetical hierarchy can be extended to hyperarithmetical hiearchy by extending the definition of Σ_n^0 and Π_n^0 sets to Π_α^0 and Σ_α^0 sets for all computable ordinals α. A set is called *hyperarithmetical* if it can be defined in the standard model of arithmetic by a computable formula. A set is Π_α^0 (Σ_α^0, respectively) if it is definable in the standard model of arithmetic by a computable Π_α formula (Σ_α formula, respectively). Ash established that in a computable structure a computable infinitary Σ_α formula defines a Σ_α^0 relation, and similarly a computable infinitary Π_α formula defines a Π_α^0 relation, and this holds uniformly.

Hyperarithmetical sets are the same as Δ_1^1 sets. Here, the superscript 1 indicates that there are function variables in addition to element variables. A set or a unary relation $X(x)$ is Δ_1^1 if it can be expressed both in a Σ_1^1 form and in a Π_1^1 form.

A Σ_1^1 form for $X(x)$ is

$$\exists f \forall y R(f, x, y), \quad \text{where } R(f, x, y) \text{ is a computable relation.}$$

Equivalently, this form allows only existential function quantifiers. Here, a computable relation R is allowed to access f as an oracle.

A Π_1^1 form for $X(x)$ is

$$\forall g \exists z S(g, x, z), \quad \text{where } S(g, x, y) \text{ is a computable relation.}$$

Equivalently, this form allows only universal function quantifiers.

There is a version of compactness theorem due to Kreisel and Barwise, which states that if Γ is a Π^1_1 set of computable sentences such that every Δ^1_1 subset of Γ has a model, then Γ has a model. As a corollary we obtain that if Γ is a Π^1_1 set of computable sentences, and if every Δ^1_1 subset of Γ has a computable model, then Γ has a computable model. For more on hyperarithmetical sets, see Ref. 1.

Theorem 7 (Ref. 11). *The following are equivalent for a class of structures $\mathbb{K}$ closed under isomorphism:*

(i) *There is a computable sentence the computable models of which are exactly the computable structures in $\mathbb{K}$;*
(ii) *$I(\mathbb{K})$ is hyperarithmetical.*

The proof of (i) $\Rightarrow$ (ii) uses that a relation defined in a computable structure $\mathcal{A}$ by a computable Σ_α (Π_α, respectively) formula is Σ^0_α (Π^0_α, respectively), with uniformity. Proof of (ii) $\Rightarrow$ (i) uses Barwise–Kreisel compactness theorem.

Theorem 8 (Kleene, Spector). *For the following classes $\mathbb{K}$, the index set $I(\mathbb{K})$ is Π^1_1-complete (hence not hyperarithmetical): well-orderings, reduced abelian p-groups.*

An abelian p-group is reduced if it has no divisible subgroup.

A computable structure $\mathcal{A}$ is *computably categorical* if for all computable structures $\mathcal{B}$ isomorphic to $\mathcal{A}$, there is a computable isomorphism f from $\mathcal{A}$ onto $\mathcal{B}$.

Theorem 9 (Ref. 9). *The index set of computably categorical structures is Π^1_1-complete.*

A computable structure $\mathcal{A}$ is *relatively computably categorical* if for all structures $\mathcal{B}$ isomorphic to $\mathcal{A}$, there is an isomorphism f from $\mathcal{A}$ onto $\mathcal{B}$, which is computable relative to the atomic diagram of $\mathcal{B}$.

Theorem 10 (Ref. 8). *The index set of relatively computably categorical structures is Σ^0_3-complete.*

Friedberg enumeration of computable structures in $\mathbb{K}$ modulo isomorphism relation $\cong$, in symbols $\mathbb{K}^{\text{comp}}/\cong$, is a sequence

$$\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots$$

of computable structures in $\mathbb{K}$ representing each isomorphism type exactly once. That is, for every computable $\mathcal{A}$ in $\mathbb{K}$, there is unique n such that $\mathcal{A} \cong \mathcal{C}_n$. The complexity of the enumeration is the complexity of the sequence of computable indices for the structures $(\mathcal{C}_n)^{n \geq 0}$.

Theorem 11 (Ref. 11). *Assume that $I(\mathbb{K})$ is hyperarithmetical. The following are equivalent:*

(i) *The computable isomorphism problem $E(\mathbb{K})$ is hyperarithmetical;*
(ii) *There is a hyperarithmetical Friedberg enumeration of $\mathbb{K}^{\text{comp}}/\cong$;*
(iii) *There is a computable ordinal α such that any two computable structures in $\mathbb{K}$ satisfying the same computable Π_α sentences are isomorphic.*

8. Σ^1_1-Complete Computable Isomorphism Problem

If $I(\mathbb{K})$ is (no more complex than) hyperarithmetical, then the computable isomorphism problem $E(\mathbb{K})$ is Σ^1_1 at worst. In this section, we will focus on those classes of structures for which we have Σ^1_1-completeness of the computable isomorphism problem.

Theorem 12 (Ref. 11). *For graphs, abelian p-groups, and arbitrary structures with at least one relation of arity at least 2, $E(\mathbb{K})$ is Σ^1_1-complete.*

Theorem 13 (Ref. 15). *For linear orderings, Boolean algebras, trees, and fields of a fixed characteristic 0 or p, $E(\mathbb{K})$ is Σ^1_1-complete.*

Theorem 14 (Ref. 10). *For torsion-free abelian groups, $E(\mathbb{K})$ is Σ^1_1-complete.*

Theorem 15 (Ref. 18). *For distributive lattices, nilpotent rings, nilpotent semigroups, and 2-step nilpotent groups, $E(\mathbb{K})$ is Σ_1^1-complete.*

By $\mathcal{A} \cong_{\text{hyp}} \mathcal{B}$, we denote that there is a hyperarithmetical isomorphism from $\mathcal{A}$ to $\mathcal{B}$.

Theorem 16 (Ref. 18). *In each of the following classes $\mathbb{K}$ with Σ_1^1-complete computable isomorphism problem, we have the following.*

(i) *There are computable structures $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \cong \mathcal{B}$ but $\mathcal{A} \ncong_{hyp} \mathcal{B}$.*
(ii) *There is a computable structure $\mathcal{M}$ with two finite tuples of elements in the domain of the same length, $\overline{a}$ and $\overline{b}$, such that $(\mathcal{M}, \overline{a}) \cong (\mathcal{M}, \overline{b})$ but $(\mathcal{M}, \overline{a}) \ncong_{hyp} (\mathcal{M}, \overline{b})$.*
(iii) *There exists a computable structure of Scott rank $SR \geq \omega_1^{CK}$ (i.e., of high Scott rank).*

The Scott rank of $\mathcal{A}$ is $< \omega_1^{CK}$ if there is a computable ordinal α such that the orbits (under automorphisms) of all tuples are defined by computable Π_α formulas. Having Scott rank $\geq \omega_1^{CK}$ is the negation of the previous statement: there is some tuple the orbit of which is not definable by any computable formula; or the orbits of all tuples are definable by computable formulas, but there is no computable bound on the complexity of these formulas.

The method we use in Ref. 18 for nilpotent groups, nilpotent rings, and nilpotent semigroups, as well as for distributive lattices, is based on uniform effective interpretations of computable binary relations or of fields into computable structures from the corresponding algebraic classes. In general, when certain structures with particularly interesting computability-theoretic properties are found, we ask whether similar examples can be found in other classes of structures. One approach is to encode the original structures into structures in a given class in a way that is algorithmic enough to preserve desired properties. This method was used by Hirschfeldt *et al.* in Ref. 19 to transfer Turing degree spectra of structures and relations, as well as computable dimensions and some other properties.

To show that the computable isomorphism problem for nilpotent groups is Σ_1^1, we use transformation from fields to Heisenberg groups. Mal'cev in Ref. 20 introduced such transformation from rings to groups of 3×3 matrices. We assign to K, a countable infinite ring with 1, the Heisenberg group G_K of 3×3 matrices:

$$h(a,b,c) = \begin{pmatrix} 1 & a & c \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{where } a, b, c \in K.$$

We can show closure under matrix multiplication:

$$h(a_0, b_0, c_0) \cdot h(a_1, b_1, c_1) = h(a_0 + a_1, b_0 + b_1, c_0 + c_1 + a_0 b_1).$$

The identity is $\mathbf{1} = h(0,0,0)$, and the inverse is

$$(h(a,b,c))^{-1} = h(-a, -b, ab - c).$$

The commutator of two matrices, $[h_0, h_1] = h_0 \cdot h_1 \cdot h_0^{-1} \cdot h_1^{-1}$, is

$$[h(a,b,c),\ h(a_1,b_1,c_1) = h(0,0,D),$$

where $D = \begin{vmatrix} a_0 & a_1 \\ b_0 & b_1 \end{vmatrix}$.

The center, $Z(G_K) = \{g \in G_K : (\forall h \in G_K)(g \cdot h = h \cdot g)\}$, is

$$\left\{ h(0,0,c) = \begin{pmatrix} 1 & 0 & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} : c \in K \right\}.$$

Furthermore, $[G_K, G_K] = Z(G_K)$ and $[Z(G_K), G_K] = \{\mathbf{1}\}$, so G is a 2-step nilpotent group.

We will further assume that K is a field and denote it by F. Then if

$$\begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \text{ we have in } G_F$$

$$[h(a_0,b_0,c_0), h(a_1,b_1,c_1)] = \mathbf{1} \Leftrightarrow \exists \alpha \left(\begin{pmatrix} a_1 \\ b_1 \end{pmatrix} = \alpha \cdot \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \right).$$

Now, we choose two non-commuting matrices in G_F; for example,

$$w_0 = h(1,0,0) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}; \quad w_1 = h(0,1,0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

for which $D = 1$. We can show that F can be interpreted into G_F. Namely, we can define in $Z(G_F)$ field operations of addition and multiplication, $\oplus$ and $\odot$, so that

$$\delta \to h(0,0,\delta) \quad \text{is a field isomorphism between } F \text{ and } Z(G_F).$$

Since the definition of $\odot$ depends on the pair w_0, w_1, we will also denote it by $\odot_{w_0,w_1}$. Let $x = h(0,0,\alpha)$, $y = h(0,0,\beta)$. Then define the addition as

$$x \oplus y = x \cdot y, \quad \text{since } x \cdot y = h(0,0,\alpha+\beta).$$

Let $z = h(0,0,\gamma)$. We define multiplication by the formula

$$x \odot_{w_0,w_1} y = z \Leftrightarrow \exists x' \exists y' ([x', w_0] = [y', w_1] = \mathbf{1} \wedge [x', w_1]$$
$$= x \wedge [w_0, y'] = y \wedge z = [x', y']) \ (*)$$

It can be shown that $\alpha\beta = \gamma$ if and only if formula (*) holds. (We can also use any other non-commuting pair as w_0, w_1.)

It follows that if F_0 and F_1 are fields such that $G_{F_0} \cong G_{F_1}$, then $F_0 \cong F_1$. Since the computable isomorphism problem for fields is Σ^1_1-complete, it follows that the computable isomorphism problem for Heisenberg groups is Σ^1_1-complete.

We will now sketch the proof that the computable isomorphism problem for distributive lattices is Σ^1_1-complete. The result also follows from Σ^1_1-completeness of the computable isomorphism of Boolean algebras, but here we give a substantially different proof using an idea of Rabin and Scott in Ref. 22.

We reduce the structures with a single binary relation (N, R) to distributive lattices $\mathcal{L}_R$. Let R be a given infinite binary relation on N, which is irreflexive. Assume that the domain of $R \geq 3$ elements. Fix disjoint sets $A = \{a_i : i \in N\}$ and $B = \{b_i : i \in N\}$. Partition B into uniformly computable infinite sets $B_{x,y}$ for $x \neq y$, such that

$$\{x,y\} = \{z,t\} \quad \Rightarrow \quad B_{x,y} = B_{z,t}$$
$$\{x,y\} \neq \{z,t\} \quad \Rightarrow \quad B_{x,y} \cap B_{z,t} = \varnothing.$$

Define a lattice that is a subset of the power set of $A \cup B$, $\mathcal{L}_R \subseteq \mathcal{P}(A \cup B)$, with respect to the set-theoretic $\subseteq, \cap, \cup$ (hence a distributive one) such that it is generated by the following elements:

all one-element sets $\{x\}$ for $x \in A \cup B$,

set A,

sets $U_{x,y} = {}_{der}\{a_x, a_y\} \cup B_{x,y}$ for all x, y with $R(x,y)$.

It is the sets $U_{x,y}$ that encode the relation R. Every element of $\mathcal{L}_R$ can be represented as the union of finitely many one-element sets, finitely many $U_{x,y}$'s, and possibly A. By omitting one-element sets that are contained in other sets, we obtain the canonical representation of the elements.

From a computable R, we can build a computable lattice $\mathcal{L}_R$. To show that $\mathcal{L}_{R_0} \cong \mathcal{L}_{R_1} \Rightarrow (N, R_0) \cong (N, R_1)$, we establish that R can be defined in $\mathcal{L}_R$ by some formula $\psi_R(x,y)$. Since the computable isomorphism problem for structures with a single binary relation is Σ^1_1-complete, it follows that the computable isomorphism problem for distributive lattices is Σ^1_1-complete.

Next, effectively transform a structure with a computable binary relation into a structure with another computable binary relation R satisfying a certain combinatorial condition so that the isomorphism is preserved. We will then transform R into a commutative ring $\mathcal{A}_R$. Fix distinct elements $a, b, c_0, c_1, c_2, \ldots$. The ring elements will be formal linear expressions of the following form:

$ma + nb + \sum_{i \in I} z_i c_i$ for $m, n, z_i \in Z$, where the index set I is a finite set of natural numbers, and Z is the set of integers.
We set

$$(\forall x)(a \cdot x = 0 \wedge b \cdot x = 0)$$

and

$$c_i \cdot c_j = \begin{cases} b & \text{if } R(i,j) \\ a & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Thus, we encode R into the multiplication $\cdot$ in the ring. It is not hard to show that for all x, y, z, we have

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) = 0.$$

This implies associativity of $\cdot$ and nilpotency of the ring $\mathcal{A}_R$. We show how we can interpret R in $\mathcal{A}_R$, so that

$$\mathcal{A}_{R_0} \cong \mathcal{A}_{R_1} \quad \Rightarrow \quad (N, R_0) \cong (N, R_1).$$

This implies that the computable isomorphism problem for nilpotent rings is Σ^1_1-complete.

We have presented several natural classes of structures for which the computable isomorphism problem is Σ^1_1-complete. This is the most complicated situation when there is no simpler way to determine whether two computable structures are isomorphic than by searching through a set of functions between the domains of two structures, which are bijections and homomorphisms. As a corollary, we obtain that in these classes there are pairs of isomorphic structures with no easy isomorphisms from the computability-theoretic perspective.

Acknowledgements

The author gratefully acknowledges support of the Simons Foundation grant #853762 and NSF FRG grant #2152095.

References

1. C. Ash and J. Knight, *Computable Structures and the Hyperarithmetical Hierarchy*. Elsevier, Amsterdam (2000).
2. W. Calvert, The isomorphism problem for classes of computable fields, *Archive for Mathematical Logic* **43**, 327–336 (2004).
3. W.C. Calvert, *Algebraic Structure and Computable Structure*, PhD dissertation, University of Notre Dame (2005).
4. W. Calvert, D. Cenzer, V. Harizanov, and A. Morozov, Effective categoricity of equivalence structures, *Ann. Pure Appl. Logic* **141**, 61–78 (2006).
5. W. Calvert, V.S. Harizanov, J.F. Knight, and S. Miller, Index sets of computable structures, *Algebra and Logic* **45**, 306–325 (2006) (English translation).
6. J. Carson, V. Harizanov, J. Knight, K. Lange, C. McCoy, A. Morozov, S. Quinn, C. Safranski, and J. Wallbaum, Describing free groups, *Trans. Am. Math. Soc.* **364**, 5715–5728 (2012).

7. C.C. Chang and H.J. Keisler, *Model Theory*, 3rd edn., North-Holland, Amsterdam (1990).
8. R.G. Downey, A.M. Kach, S. Lempp, and D.D. Turetsky, Computable categoricity versus relative computable categoricity, *Fundamenta Mathematicae* **221**, 129–159 (2013).
9. R.G. Downey, A.M. Kach, S. Lempp, A.E.M. Lewis-Pye, A. Montalbán, and D.D. Turetsky, The complexity of computable categoricity, *Adv. Math.* **268**, 423–466 (2015).
10. R. Downey and A. Montalbán, The isomorphism problem for torsion-free abelian groups is analytic complete, *J. Algebra* **320**, 2291–2300 (2008).
11. S.S. Goncharov, J.F. Knight, Computable structure and antistructure theorems, *Algebra and Logic* **41**, 351–373 (2002) (English translation).
12. E.B. Fokina, Index sets for some classes of structures, *Ann. Pure Appl. Logic* **157**, 139–147 (2009).
13. E.B. Fokina, S.S. Goncharov, V. Harizanov, O.V. Kudinov, and D. Turetsky, Index sets for n-decidable structures categorical relative to m-decidable presentations, *Algebra Logic* **54**, 336–341 (2015) (English translation).
14. E.B. Fokina, V. Harizanov, and A. Melnikov, Computable model theory, in: R. Downey (ed.), *Turing's Legacy: Developments from Turing Ideas in Logic*, Cambridge University Press/ASL (2014), pp. 124–194.
15. H. Friedman and L. Stanley, A Borel reducibility theory for classes of countable structures, *J. Symbol. Logic* **54**, 894–914 (1989).
16. T. Ha, V. Harizanov, L. Marshall and H. Walker, Computability and definability, in: D. Cenzer, C. Porter and J. Zapletal (eds.), *Structure and Randomness in Computability and Set Theory*, World Scientific, Singapore (2020), pp. 285–355.
17. V.S. Harizanov, Pure computable model theory, in: Yu.L. Ershov, S.S. Goncharov, A. Nerode, and J.B. Remmel (eds.), *Handbook of Recursive Mathematics*, vol. 1, Elsevier, Amsterdam (1998), pp. 3–114.
18. V.S. Harizanov, S. Lempp, C.F.D. McCoy, A.S. Morozov and R. Solomon, On the isomorphism problem for some classes of computable algebraic structures, *Arch. Math. Logic* **61**, 813–825 (2022).
19. D.R. Hirschfeldt, B. Khoussainov, R.A. Shore, and A.M. Slinko, Degree spectra and computable dimension in algebraic structures, *Ann. Pure Appl. Logic* **115**, 71–113 (2002).
20. A.I. Mal'cev, On a correspondence between rings and groups, *Am. Math. Soc. Transl.* (Series 2), **45**, 221–232 (1965).
21. C. McCoy and J. Wallbaum, Describing free groups, Part II: Π^0_4 hardness and no Σ^0_2 basis, *Trans. Am. Math. Soc.* **364**, 5729–5734 (2012).
22. M.O. Rabin and D.S. Scott, The undecidability of some simple theories, unpublished manuscript.
23. R.I. Soare, Turing computability, *Theory and Applications*, Springer-Verlag, Berlin (2016).
24. R.I. Soare, *Recursively Enumerable Sets and Degrees. A Study of Computable Functions and Computably Generated Sets.* Springer-Verlag, Berlin (1987).