Equivalence Relations That Are Σ^0_3 Complete for Computable Reducibility*
(Extended Abstract)

Ekaterina Fokina[1], Sy Friedman[1], and André Nies[2]

[1] Kurt Gödel Research Center, University of Vienna, Austria
{efokina,sdf}@logic.univie.ac.at
[2] Department of Computer Science, University of Auckland, Auckland, New Zealand
andre@cs.auckland.ac.nz

Abstract. Let E, F be equivalence relations on $\mathbb{N}$. We say that E is computably reducible to F, written $E \leq F$, if there is a computable function $p\colon \mathbb{N} \to \mathbb{N}$ such that $xEy \leftrightarrow p(x)Fp(y)$. We show that several natural Σ^0_3 equivalence relations are in fact Σ^0_3 complete for this reducibility. Firstly, we show that one-one equivalence of computably enumerable sets, as an equivalence relation on indices, is Σ^0_3 complete. Thereafter, we show that this equivalence relation is below the computable isomorphism relation on computable structures from classes including predecessor trees, Boolean algebras, and metric spaces. This establishes the Σ^0_3 completeness of these isomorphism relations.

1 Introduction

Invariant descriptive set theory studies the complexity of equivalence relations on the reals via Borel reductions (see [6]). An analog for equivalence relations on natural numbers, where the reductions are computable functions, was already introduced in [1], and has received considerable attention in recent years [7,3].

The isomorphism relation on a class of structures is a natural example of an equivalence relation. A countable structure in a countable signature can be encoded by a real. The complexity of the isomorphism relation on (reals encoding) countable structures has been studied in invariant descriptive set theory beginning with H. Friedman and Stanley [5]. For instance, they showed that isomorphism of countable graphs is not Borel complete for analytic equivalence relations.

We may assume that the domain of a countable structure is an initial segment of $\mathbb{N}$. Then the quantifier free statements involving elements of the structure can also be encoded by natural numbers. Suppose the signature is computable. We say that a presentation of a countable structure is *computable* if its atomic

* The first and the second authors acknowledge the generous support of the FWF through projects Elise-Richter V206, and P22430-N13. The third author is partially supported by the Marsden Fund of New Zealand under grant 09-UOA-187.

L. Ong and R. de Queiroz (Eds.): WoLLIC 2012, LNCS 7456, pp. 26–33, 2012.

diagram, that is, all the quantifier free facts about the structure, is a computable set. A computable index for the atomic diagram is also called a *computable index* for the structure. As a general rule, familiar countable structures all have computable presentations. Examples include $(\mathbb{Z}, +)$ and $(\mathbb{Q}, <)$.

Following Fokina et al. [4], for a class $\mathcal{K}$ of structures, we denote by $I(\mathcal{K})$ the set of computable indices for structures in $\mathcal{K}$. For common classes, this will be an arithmetical set. Isomorphism can now be viewed as an equivalence relation on $I(\mathcal{K})$, and clearly is Σ_1^1. Fokina et al. [4] studied possible analogs of some results in [5] for isomorphism on computable structures. Their reduction, denoted $\leq_{FF}$, was a slight extension of computable reducibility which allows for partial computable functions as reductions as long as their domain contains the relevant set $I(\mathcal{K})$. In contrast to the above-mentioned result of [5], they proved as a main result that isomorphism of computable graphs is $\leq_{FF}$ complete for Σ_1^1 equivalence relations. Coding graphs into structures, they then obtained the similar result for other classes, such as torsion free abelian groups, and linear orders. Boolean algebras were notably absent.

In this paper, we go one step further in effectivizing the setting of [5]: we also require that the isomorphisms are computable. For computable presentations C, D of structures in the same computable signature, we write

$$C \cong_{comp} D$$

if there is a partial computable bijection between the domains of C, D (initial segments of $\mathbb{N}$) which induces an isomorphism of the structures. Clearly, if $I(\mathcal{K})$ is Σ_3^0, then computable isomorphism on $I(\mathcal{K})$ is also Σ_3^0.

We will show that for several classes of structures, the computable isomorphism relation is a Σ_3^0-complete equivalence relation under computable reducibility: computable trees and graphs, computable Boolean algebras, and (with some adjustment of terminology) metric spaces. Note that for some classes, however, the computable isomorphism problem may be less complex than Σ_3^0. For instance, consider the class $\mathcal{K}$ of computable permutations of order 2. Then $I(\mathcal{K})$ is Π_2^0. The computable isomorphism relation on $I(\mathcal{K})$ is also Π_2^0. This is so because we only need to figure out whether for two given permutations, both have the same number of 1-cycles, and the same number of 2-cycles.

Our completeness results rely on a recursion theoretic fact of interest by itself. As usual let $(W_e)_{e \in \mathbb{N}}$ be an effective listing of the computably enumerable sets. Recall that sets $A, B \subseteq \mathbb{N}$ are *1-equivalent*, $A \equiv_1 B$, if there is a computable permutation h of $\mathbb{N}$ such that $h(A) = B$.

Theorem 1. *For each Σ_3^0 equivalence relation S, there is a computable function g such that*

$$ySz \Rightarrow W_{g(y)} \equiv_1 W_{g(z)}, \textit{ and}$$
$$\neg ySz \Rightarrow W_{g(y)}, W_{g(z)} \textit{ are Turing incomparable.}$$

The proof will be given in Section 3. As an immediate consequence, we have:

Corollary 2. *Many-one equivalence and 1-equivalence on indices of c.e. sets are Σ^0_3 complete for equivalence relations under computable reducibility.*

Note that this is significantly stronger than the mere Σ^0_3 completeness of $\equiv_m$ as a set of pairs of c.e. indices, which follows for instance because the m-complete c.e. set have a Σ^0_3 complete index set.

As a further consequence, Turing equivalence on indices of c.e. sets is a Σ^0_3 hard equivalence relation for computable reducibility. However, this equivalence relation is only Σ^0_4. We conjecture that it is in fact Σ^0_4 complete in our sense.

In the following Section 2, we will encode 1-equivalence on indices of c.e. sets into computable isomorphism for the relevant classes. We then use Corollary 2 to conclude these isomorphism relations are Σ^0_3 complete.

2 Computable Isomorphism of Computable Structures

2.1 Computable Trees and Computable Equivalence Relations

We use the terminology of Fokina, Friedman et al. [4]. In particular, a tree is a structure in the language containing the predecessor function as a single unary function symbol. The root is its own predecessor. A countable tree can be represented by a nonempty subset B of $\omega^{<\omega}$ closed under prefixes. The unary predecessor function takes off the last entry of a non-empty tuple of natural numbers, and maps the empty tuple to itself.

A tree has a computable presentation iff we can choose B c.e. For in that case B is the range of a partial computable 1-1 function ϕ with domain an initial segment of ω; the preimage of the predecessor function under ϕ is the required computable atomic diagram.

We let

$$T_e = \{\sigma\colon\ \exists \tau \succeq \sigma\, [\tau \in W_e]\},$$

where the e-th c.e. set W_e is now viewed as a subset of $\omega^{<\omega}$. Then $(T_e)_{e\in\mathbb{N}}$ is a uniform listing of all computable trees.

We say a tree has height k if every leaf has length at most k.

Proposition 3. *Computable isomorphism of computable trees of height* 2 *where every node at level* 1 *has out-degree at most* 1 *is a complete* Σ^0_3 *equivalence relation.*

Proof. Let h be a computable function such for each e, $T_{h(e)}$ is the tree

$$\{\emptyset\} \cup \{\langle x\rangle\colon\ x \in \omega\} \cup \{\langle x, 0\rangle\colon\ x \in W_e\}.$$

Clearly, $W_y \equiv_1 W_z$ iff $T_{h(y)}$ is computably isomorphic to $T_{h(z)}$. Now we apply Corollary 2.

A similar argument shows:

Proposition 4. *Computable isomorphism of computable equivalence relations where every class has at most 2 members is a complete* Σ^0_3 *equivalence relation.*

2.2 Boolean Algebras

For a linear order L with least element, *Intalg* L denotes the subalgebra of the Boolean algebra $\mathcal{P}(L)$ generated by intervals $[a, b)$ of L where $a \in L$ and $b \in L \cup \{\infty\}$. Here ∞ is a new element greater than any element of L, and $[a, \infty)$ is short for $\{x \in L\colon\ x \geq a\}$. Note that *Intalg* L consists of all sets S of the form

$$S = \bigcup_{r=1}^{n} [a_r, b_r)$$

where $a_0 < b_0 < a_1 \ldots < b_n \leq \infty$. From a computable presentation of L as a as a linear order, we may canonically obtain a computable presentation of the Boolean algebra *Intalg* L.

Theorem 5. *Computable isomorphism of computable Boolean algebras is complete for Σ_3^0 equivalence relations.*

Proof. Let $(V^e)_{e\in\mathbb{N}}$ be an effective listing of the c.e. sets containing the even numbers. The relation of 1-equivalence $\equiv_1$ of c.e. sets V^e is Σ_3^0 complete by Theorem 1 and its proof below. We will computably reduce it to computable isomorphism of computable Boolean algebras. We define the Boolean algebra C^e to be the interval algebra of a computable linear order L^e. Informally, to define L^e, we begin with the order type ω. For each $x \in \omega$, when x enters V^k we replace x by a computable copy of $[0, 1)_{\mathbb{Q}}$. More formally,

$$L^e = \bigoplus\nolimits_{x\in\omega} M_x^e,$$

where M_x^e has one element $m_x^k = 2x$, until x enters V^e; if and when that happens, we expand M_x^e to a computable copy of $[0, 1)_{\mathbb{Q}}$, using the odd numbers, while ensuring that $m_x^k = \min M_x^k$ holds in L^k. Also note that the domain of L^k is $\mathbb{N}$ because $0 \in V^k$.

Claim. $V^e \equiv_1 V^i \Leftrightarrow C^e \cong_{comp} C^i$.

$\Rightarrow$: Suppose $V^e \equiv_1 V^i$ via a computable permutation π. We define a computable isomorphism $\Phi : C^e \cong C^i$.
(a) Let $\Phi(m_x^e) = m_{\pi(x)}^i$. Once x enters V^e, we know that $\pi(x) \in V^i$. So we may always ensure that Φ restricts to a computable isomorphism of linear orders $M_x^e \cong M_{\pi(x)}^i$.
(b) Consider an element S of C^e. It is given in the form $S = \bigcup_{r=1}^{n} [a_r, b_r)$ where $a_0 < b_0 < a_1 \ldots < b_n$ for $a_r, b_r \in L^e \cup \{\infty\}$ as above. If $b_n < \infty$, we can compute the maximal $x \in \omega$ such that $M_x^e \cap S \neq \emptyset$. Define

$$\Phi(S) = \bigcup_{y\leq x} \Phi(S \cap M_y^e).$$

Note that the set $\Phi(S \cap M_y^e)$ can be determined by (a).

If $b_n = \infty$, then let $\Phi(S)$ be the complement in L^i of $\Phi(L^e \setminus S)$.

$\Leftarrow$: Now suppose that $C^e \cong_{comp} C^i$ via some computable isomorphism Φ. We show that $V^e \leq_1 V^i$ via some computable function f. Suppose we have defined $f(y)$ for $y < x$. We have $\Phi(M^e_x) = \bigcup_{r=1}^n [a_r, b_r)$ where $a_r, b_r \in L^i \cup \{\infty\}$ as above.

If $n > 1$ then M^e_x is not an atom in C^e, whence $x \in V^e$. Thus let $f(x)$ be the least even number that does not equal $f(y)$ for any $y < x$.

Now suppose $n = 1$. If $a_1 = m^i_y, b_1 = m^i_{y+1}$ then let $f(x) = y$. Otherwise, again we know M^e_x is not an atom in C^e, and define $f(x)$ as before.

By symmetry, we also have $V^i \leq_1 V^e$, and hence $V^i \equiv_1 V^e$ by Myhill's theorem.

2.3 Metric Spaces

Let (M, d) be a metric space, and let $(\alpha_i)_{i\in\mathbb{N}}$ be a dense sequence in M without repetitions. We say that $\mathcal{M} = (M, d, (\alpha_i)_{i\in\mathbb{N}})$ is a *computable metric space* if $d(\alpha_i, \alpha_k)$ is a computable real uniformly in i, k. We call the elements of the sequence $(\alpha_i)_{i\in\mathbb{N}}$ the *special points*. For background on computable metric spaces, see [2].

A computable metric space is *discrete* if every point is isolated. For such a space, necessarily every point is a special point.

Corollary 6. *Computable isometry of discrete computable metric spaces is complete for Σ^0_3 equivalence relations.*

Proof. Given a computable tree B, create a discrete computable metric space M_B as follows: if a string $\langle x\rangle$ enters B, add a point p_x. If later $\langle x, i\rangle$ enters B for the first i, add a further point q_x. Declare $d(p_x, q_x) = 1/4$. Declare $d(p_x, p_y) = 1$ and $d(q_x, p_y) = 1$ (if q_x exists). Clearly for trees B, C as in Cor. 3, B is computably isomorphic to C iff M_B is computably isometric to M_C.

3 Proof of Theorem 1

Since S is Σ^0_3, there is a uniformly c.e. triple sequence

$$(V_{y,z,i})_{y,z,i\in\omega, y<z}$$

of initial segments of $\mathbb{N}$ such that for each $y < z$,

$$ySz \Leftrightarrow \exists i\, V_{y,z,i} = \omega.$$

We build a uniformly c.e. sequence of sets $A_x = W_{g(x)}$ $(x \in \omega)$, g computable. We meet the following coding requirements for all $y < z$ and $i \in \omega$.

$G_{y,z,i}$: $V_{y,z,i} = \omega \Rightarrow A_y \equiv_1 A_z$.

We meet diagonalization requirements for $u \neq v$,

$N_{u,v,e}$: $u = \min[u]_S \wedge v = \min[v]_S \Rightarrow A_u \neq \Phi_e(A_v)$.

where Φ_e is the e-th Turing functional, and $[x]_S$ denotes the S-equivalence class of x. Meeting these requirements suffices to establish the theorem.

The basic strategies to meet the requirements are as follows. If $V_{y,z,i} = \omega$, a strategy for $G_{y,z,i}$ "finds out" that z is S-related to the smaller y. Hence it builds a computable permutation h such that $A_y \equiv_1 A_z$ via h.

A strategy for $N_{u,v,e}$ picks a witness n, and waits for $\Phi_e(A_v; n)$ to converge. Thereafter, it ensures that this computation is stable and $A_u(n)$ does not equal its output $\Phi_e(A_v; n)$ by enumerating n into A_u if this output is 0.

The tree of strategies. To avoid conflicts between strategies that enumerate into the same set A_z, we need to provide the strategies with a guess at whether z is least in its S-equivalence class $[z]_S$. An N-type strategy will only enumerates into A_z if according to its guess, z is least in its $[z]_S$; a G-type strategy only enumerates into A_z if according to its guess, z is not least.

Fix an effective priority ordering of all requirements. We define a tree T of strategies, which is a computable subtree T of $2^{<\omega}$. We write $\alpha : R$ if strategy α is associated with the requirement R. By recursion on $|\alpha|$, we define whether $\alpha \in T$, and which is the requirement associated with α. We also define a function L mapping $\alpha \in T$ to a cofinite set $L(\alpha)$ consisting of the numbers x such that according to α's guesses, x is least in its equivalence class.

Let $L(\emptyset) = \omega$. Assign to α the highest priority requirement R not yet assigned to a proper prefix of α such that either (a) or (b) hold.

(a) R is $G_{y,z,i}$ and $z \in L(\alpha)$; in this case put both $\alpha 0$ and $\alpha 1$ on T, and define $L(\alpha 0) = L(\alpha) - \{z\}$ while $L(\alpha 1) = L(\alpha)$ (along $\alpha 0$ we know that x is no longer the least in its equivalence class)
(b) R is $N_{u,v,e}$ and $u, v \in L(\alpha)$; in this case put only $\alpha 0$ on T, and define $L(\alpha 0) = L(\alpha)$.

For strings $\alpha, \beta \in 2^{<\omega}$, we write $\alpha <_L \beta$ if there is i such that $\alpha \restriction_i = \beta \restriction_i$, $\alpha(i) = 0$ and $\beta(i) = 1$. We let $\alpha \preceq \beta$ denote that α is a prefix of β. We define a linear ordering on strings by

$$\alpha \leq \beta \text{ if } \alpha <_L \beta \text{ or } \alpha \preceq \beta.$$

Construction of a u.c.e. sequence of sets $(A_x)_{x \in \mathbb{N}}$. We declare in advance that $A_x(4m + 1) = 0$ and $A_x(4m + 3) = 1$ for each x, m. The construction then only determines membership of even numbers in the A_x.

We define a computable sequence $(\delta_s)_{s \in \mathbb{N}}$ of strings on T of length s. Suppose inductively that δ_t has been defined for $t < s$. Suppose $k < s$ and that $\eta = \delta_s \restriction_k$ has been defined. If $\eta : N_{u,v,e}$ let $\delta_s(k) = 0$. Otherwise $\eta : G_{y,z,i}$. Let $t < s$ be the largest stage such that $t = 0$ or $\eta \preceq \delta_t$. Let $\delta_s(k) = 0$ if $V_{y,z,i,s} \neq V_{y,z,i,t}$, and otherwise $\delta_s(k) = 1$.

The *true path* TP is the lexicographically leftmost path $f \in 2^\omega$ such that $\forall n \, \exists^\infty s \geq n \, [\delta_s \restriction_n \prec f]$. To *initialize* a strategy α means to return it to its first instruction. If $\alpha : G_{y,z,i}$ we also make the partial computable function h_α built by the strategy α undefined on all inputs. At stage s, let $\mathrm{init}(\alpha, s)$ denote the largest stage $\leq s$ at which α was initialized.

An $N_{u,v,e}$ *strategy* α. At stages s:

(a) Appoint an unused even number $n > \mathrm{init}(\alpha, s)$ as a witness for diagonalization. Initialize all the strategies $\beta \succ \alpha$.
(b) Wait for $\Phi_e(A_v; n)[s]$ to converge with output r. If $r = 0$ then put n into A_u. Initialize all the strategies $\beta \succ \alpha$.

A $G_{y,z,i}$ strategy α. If $\alpha 0$ is on the true path then this strategy builds a computable increasing map h_α from even numbers to even numbers such that $A_y(k) = A_z(h_\alpha(k))$ for each k. Furthermore, $A_z - \mathrm{range}(h_\alpha)$ is computable. By our definitions of A_y and A_z on the odd numbers, this implies that h_α can be extended to a computable permutation showing that $A_y \equiv_1 A_z$, as required.

At stages s, if $\alpha 0 \subseteq \delta_s$, let $t < s$ be greatest such that $t = 0$ or $\alpha 0 \subseteq \delta_t$, and do the following.

(a) For each even $k < s$ such that $k \notin \mathrm{dom}(h_{\alpha,t})$ pick an unused even value $m = h_{\alpha,s}(k) > \mathrm{init}(\alpha, s)$ in such a way that h_α remains increasing.
(b) From now on, unless α is initialized, ensure that $A_z(m) = A_y(k)$. (We will verify that this is possible.)

The stage-by-stage construction is as follows. At stage $s > 0$ initialize all strategies $\alpha >_L \delta_s$. Go through substages $i \leq s$. Let $\alpha = \delta_s \restriction_i$. Carry out the strategy α at stage s.

Verification. To show the requirements are met, we first check that there is no conflict between different strategies that enumerate into the same set A_z.

Claim. Let $\alpha\colon G_{y,z,i}$. Then (b) in the strategy for α can be maintained as long as α is not initialized.

To prove the claim, suppose a strategy $\beta \neq \alpha$ also enumerates numbers into A_z. If $\alpha 0 <_L \beta$ then β is initialized when α extends its map h_α, so the numbers enumerated by β are not in the range of h_α. If $\beta <_L \alpha 0$ then α is initialized when β is active, so again the numbers enumerated by β are not in the range of h_α. Now suppose neither hypothesis holds, so $\alpha 0 \preceq \beta$ or $\beta \prec \alpha$.

Case β: $N_{z,v,e}$. In this case $\alpha 0 \preceq \beta$ is not possible because $z \notin L(\alpha 0)$. If $\beta \prec \alpha$ then α is initialized when β appoints a new diagonalization witness.
Case β: $G_{y',z,i'}$. In this case $\alpha 0 \preceq \beta$ is not possible because $z \notin L(\alpha 0)$. If $\beta 1 \preceq \alpha$ then α is initialized each time β extends its map h_β. Finally, $\beta 0 \preceq \alpha$ is not possible because $z \notin L(\beta 0)$. This proves the claim.

Claim. Let α be the $N_{u,v,e}$ strategy on the true path. Suppose α is not initialized after stage s. Then α only acts finitely often, and meets its requirement.

At some stage $\geq \mathrm{init}(\alpha, s)$ the strategy α picks a permanent witness n. No strategy $\beta \prec \alpha$ can put n into A_u because $u \in L(\alpha)$. No other strategy can put n into A_u because of the initialization α carries out when it picks n. Suppose now that at a later stage t, a computation $\Phi_e(A_v; n)[t]$ converges. Since $v \in L(\alpha)$, no G-type strategy $\beta \prec \alpha$ enumerates into A_v. Thus the initialization of strategies $\gamma \succ \alpha$ carried out by α at that stage t will ensure that this computation is preserved with value different from $A_u(n)$. This proves the claim.

It is now clear by induction that each strategy α on the true path is initialized only finitely often. Thus the N-type requirements are met. Now suppose $\alpha\colon G_{y,z,i}$ and $\alpha 0$ is on the true path. Then no strategy $\beta \succeq \alpha 0$ enumerates into A_z. Thus by the initialization at stages s such that $\alpha 0 \preceq \delta_s$, the set $A_z - \mathrm{range}(h_\alpha)$ is computable. As noted earlier, this implies that h_α can be extended to a computable permutation showing that $A_y \equiv_1 A_z$. There is a computable bijection q between the set of odd numbers and the set of numbers that are odd, or even but not in the range of h_α, so that $m \in A_y \leftrightarrow q(m) \in A_z$. Now let the permutation be $q \cup h_\alpha$.

References

1. Bernardi, C., Sorbi, A.: Classifying positive equivalence relations. J. Symb. Log. 48(3), 529–538 (1983)
2. Brattka, V., Hertling, P., Weihrauch, K.: A tutorial on computable analysis. In: Barry Cooper, S., Löwe, B., Sorbi, A. (eds.) New Computational Paradigms: Changing Conceptions of What is Computable, pp. 425–491. Springer, New York (2008)
3. Coskey, S., Hamkins, J.D., Miller, R.: The hierarchy of equivalence relations on the natural numbers under computable reducibility, pp. 1–36, `http://arxiv.org/abs/1109.3375` (submitted)
4. Fokina, E.B., Friedman, S.-D., Harizanov, V.S., Knight, J.F., McCoy, C.F.D., Montalbán, A.: Isomorphism relations on computable structures. J. Symb. Log. 77(1), 122–132 (2012)
5. Friedman, H., Stanley, L.: A Borel reducibility theory for classes of countable structures. Journal of Symbolic Logic 54, 894–914 (1989)
6. Gao, S.: Invariant descriptive set theory. Pure and Applied Mathematics (Boca Raton), vol. 293. CRC Press, Boca Raton (2009)
7. Gao, S., Gerdes, P.: Computably enumerable equivalence relations. Studia Logica 67(1), 27–59 (2001)