

Infinite Time Decidable Equivalence Relation Theory

Samuel Coskey and Joel David Hamkins

Abstract We introduce an analogue of the theory of Borel equivalence relations in which we study equivalence relations that are decidable by an infinite time Turing machine. The Borel reductions are replaced by the more general class of infinite time computable functions. Many basic aspects of the classical theory remain intact, with the added bonus that it becomes sensible to study some special equivalence relations whose complexity is beyond Borel or even analytic. We also introduce an infinite time generalization of the countable Borel equivalence relations, a key subclass of the Borel equivalence relations, and again show that several key properties carry over to the larger class. Lastly, we collect together several results from the literature regarding Borel reducibility which apply also to absolutely Δ_2^1 reductions, and hence to the infinite time computable reductions.

1 Introduction

The subject of Borel equivalence relation theory—by now a highly developed, successful enterprise—begins with the observation (see Friedman-Stanley [3] and Hjorth-Kechris [11]) that many classification problems arising naturally in mathematics can be regarded as relations, often Borel relations, on a standard Borel space. For example, since groups are determined by their multiplication functions, the isomorphism relation on countable groups can be regarded as a relation on the subspace of $2^{\omega \times \omega \times \omega}$ corresponding to the graphs of such functions. This isomorphism relation is properly analytic, but its restriction to finitely generated groups is Borel. The subject aims to understand these and many other relations by placing them into a hierarchy of relative complexity measured by Borel reducibility. Specifically, an equivalence relation E on a standard Borel space is *Borel reducible* to another, F , if

Received May 6, 2010; accepted September 27, 2010; printed April 11, 2011

2010 Mathematics Subject Classification: Primary, 03D30, 03D65, 03E15

Keywords: set theory, descriptive set theory, infinite time computation, equivalence relations

© 2011 by University of Notre Dame 10.1215/00294527-1306199

there is a Borel function f such that

$$x E y \iff f(x) F f(y)$$

for all x, y in the underlying space. In this case, we write $E \leq_B F$, and we think of this reducibility as asserting that F is at least as complex as E . Indeed, the function $[x]_E \mapsto [f(x)]_F$ is an explicit classification of the E -equivalence classes using F -equivalence classes. More generally, composition with f gives an explicit method of turning any F -invariant classification into an E -invariant classification, and in this sense, the classification problem for F is at least as hard as the classification problem for E .

In this article, a small project, we aim to extend the analysis from the Borel context to a larger context of effectivity. Namely, we shall consider the context of infinite time computability, a realm properly between Borel and Δ_2^1 . Specifically, we shall enlarge the reducibility concept by allowing infinite time computable reduction functions (a class of functions we review in Section 2). This is sensible for several reasons. First, the class of infinite time computable functions properly extends the class of Borel functions—a function is Borel exactly when it is infinite time computable in uniformly bounded countable ordinal time—while retaining much of the effective flavor and content of the Borel context. The infinite time computable functions, determined by the operation of a finite Turing machine program computing in transfinite ordinal time, seem in many ways as “explicit” as the Borel functions are sometimes described to be, but they reach much higher into the descriptive set-theoretic hierarchy, well into the class Δ_2^1 . Second, meanwhile, many natural sets and equivalence relations that lay outside the Borel boundary are captured within the bounds of infinite time computability. For instance, it is infinite time computable, but not Borel, to decide whether a given real codes a well-order. Hence, our framework allows for the study of the isomorphism relation on well-ordered sets. More generally, the isomorphism relation for arbitrary countable structures in arbitrary countable languages is infinite time computable, but not Borel. Third, it will turn out that much (but not all) of the Borel theory carries over to our enlarged context, at least for many of the relations studied by that theory. Positive instances of Borel reducibility, of course, carry over directly because Borel functions are infinite time computable. Conversely, a deep aspect of the Borel theory is that many of the proofs of nonreducibility, that is, instances of equivalence relations E and F for which $E \not\leq_B F$, actually overshoot the Borel context by showing, for example, that there are no measurable reduction functions for a given pair of equivalence relations; since infinite time computable functions are measurable, these arguments also rule out reducibilities in our context. The point is that such nonreduction arguments lay at the center of the Borel theory, and the overshooting phenomenon means that in these instances, the nonreduction results transfer largely intact to the infinite time computable context. Thus, in our project we explore the limits of this phenomenon. In summary, we propose to study the hierarchy of equivalence relations under the concepts of reducibility provided by infinite time computability.

In contrast, recent work of Knight [17] and others aim far in the other direction, by restricting the notion of reducibility from Borel functions down to the class of (ordinary) Turing computable functions. Since this is a very restrictive notion of reduction, it allowed them to separate many complexity classes more finely and to

analyze even classes of finite structures. Our work here can be seen as complementary to theirs, since we consider comparatively generous notions of reducibility.

This paper is organized as follows. In Section 2, we shall describe in detail the infinite time Turing machines, the class of sets which they decide, and the class of functions which they compute. In fact, we shall define several distinct ways in which these machines can accept their input, leading to distinct but closely related notions of effectivity. In Section 3, we give some basic facts about Borel equivalence relations and Borel reductions and compare this situation with the case of infinite time computable equivalence relations and functions. In Section 4, we consider the special case of *countable* Borel equivalence relations (i.e., those with countable classes). We define the class of infinite time enumerable equivalence relations, which is a natural generalization of the class of countable Borel equivalence relations to the infinite time context. In the last section, Section 5, we give a survey of methods of demonstrating *non-reducibility* results in the case of absolutely Δ_2^1 reductions. We use these methods to determine the relationships between the infinite time computable equivalence relations discussed in this paper, and these relationships are summarized in Figure 3 at the conclusion of the paper.

2 The Infinite Time Complexity Classes

Infinite time Turing machines, introduced in [8], generalize the operation of ordinary Turing machines into transfinite ordinal time. An infinite time Turing machine has three one-way infinite tapes (the input tape, the work tape, and the output tape), each with ω many cells exhibiting either 0 or 1, and computes according to a finite program with finitely many states. Successor stages of computation proceed in exactly the classical manner, with the machine reading from and writing to the tape and moving the head left or right according to the program instructions for the current state. At limit time stages, the configuration of the machine is determined by placing the head on the left-most cell, setting the state to a special “Limit” state, and updating each cell of the tape with the lim sup of the previous values exhibited by that cell (which is the limit value, if the value had stabilized, otherwise 1). Computation stops only when the “Halt” state is explicitly attained, and in this case, the machines outputs the contents of the output tape. Since the input and output tapes naturally accommodate infinite binary sequences, that is, elements of Cantor space 2^ω , the machines provide infinitary notions of computability and decidability on Cantor space. The machines can be augmented with additional input tapes to accommodate real parameters or oracles. We denote by $\varphi_e^z(x)$ the output of program e on input x with parameter z , if this computation halts; if it doesn’t halt, then $\varphi_e^z(x)$ is undefined, or diverges. A partial function $f : 2^\omega \rightarrow 2^\omega$ is infinite time *computable* if there exists e and z such that $f = \varphi_e^z$. For a program e operating on a machine without a parameter tape, we denote the output as $\varphi_e(x)$, and say that the corresponding function φ_e is infinite time computable *without parameters*.

A simple cofinality argument [8, Theorem 1.1] shows that if an infinite time computation halts, then it does so in a countable ordinal number of steps. And if a computation does not halt, then it is necessarily due to the fact that it is caught in an infinite loop, in the strong sense that at limits of repetitions of this loop, the computation remains inside the loop. (Looping phenomenon in ordinal time is complicated

by the curious fact that an infinite time computation can exactly repeat a configuration, looping ω many times, but nevertheless escape the loop at the limit of these repetitions.)

A subset $A \subset 2^\omega$ is infinite time *decidable* if its characteristic function is infinite time computable, and infinite time *semidecidable* if it is the domain of an infinite time computable function. These concepts naturally extend to subsets of $(2^\omega)^n$ for $n \leq \omega$ by the use of canonical pairing functions (or by using extra input tapes). We warn the reader that a function can have an infinite time decidable graph, as a subset of the plane, without being an infinite time computable function (see [8, Lost Melody Theorem 4.9]). The reason has to do with the inability of the infinite time machines to search entirely through Cantor space in a computable manner, and so the analogue of the classical algorithm showing this doesn't happen for finite-time computations on ω simply does not work here. We say that a function f is infinite time *semicomputable* if the graph of f is infinite time decidable. Thus, every infinite time computable function is infinite time semicomputable, but not generally conversely.

We let \mathbf{D} denote the class of infinite time decidable subsets of 2^ω . Since we have allowed a real parameter z in the definition of an infinite time decidable set, the class \mathbf{D} fits naturally into the boldface descriptive set theory context. Similarly, let $s\mathbf{D}$ denote the class of semidecidable subsets of 2^ω , and $s\overline{\mathbf{D}}$ the class of co-semidecidable subsets of 2^ω . The classes of infinite time decidable sets and functions subsume the corresponding Borel classes. In fact, we have the following remarkable characterization.

Theorem 2.1 *A set A is Borel if and only if it is decided by a program which, for some countable ordinal α , halts uniformly on all inputs in fewer than α steps. Similarly, a function f is Borel if and only if it is computed by a program which, for some countable ordinal α , halts uniformly in fewer than α steps.*

Thus, Borel effectivity is the entry into the larger hierarchy of computability provided by infinite time Turing machines. Before the proof, let us fix the notation $\omega_{1,ck}$ for the supremum of the ordinals that are computable by a classical Turing machine. If $z \in 2^\omega$, then let $\omega_{1,ck}^z$ denote the relativized version, the supremum of the ordinals that are computable by a classical Turing machine with oracle parameter z . The ordinal $\omega_{1,ck}^z$ is known to be the least ordinal which is admissible in the parameter z . The proof of Theorem 2.1 is then an easy consequence of [8, Theorem 2.7], which states that A is Δ_1^1 if and only if it can be decided by a program which halts uniformly in fewer than $\omega_{1,ck}$ steps.

Proof of Theorem 2.1 To establish the first statement, we need only observe that the proof of [8, Theorem 2.7] relativizes to show that a set A is $\Delta_1^1(z)$ if and only if it can be decided by a program which halts uniformly in fewer than $\omega_{1,ck}^z$ steps. This implies directly that every Borel set is infinite time decidable in uniformly bounded time. Conversely, if A is infinite time decidable from parameter z in time uniformly bounded by α , then simply augment z with a real y coding α , so that $\alpha < \omega_{1,ck}^{z \oplus y}$, and conclude that $A \in \Delta_1^1(z \oplus y)$ and therefore A is Borel.

For the second statement, suppose that f is infinite time computable uniformly in some number of steps which is bounded below ω_1 . It follows that the graph of f is infinite time decidable in some bounded number of steps, and therefore, by the

first paragraph, the graph of f is Borel. Conversely, if f is a Borel function, then for each n , the set

$$A_n := \{x \in 2^\omega : f(x)(n) = 0\}$$

is Borel, and hence it is infinite time decidable by a program which halts uniformly in fewer than α_n steps. It follows that $f(x)$ is infinite time decidable by a program which halts uniformly in fewer than $\sup(\alpha_n) + \omega$ steps. \square

In the following proposition, $\text{Abs}\Delta_2^1$ denotes the class of absolutely Δ_2^1 sets, where a set A is *absolutely* Δ_2^1 when it is defined by a Π_2^1 formula φ and by a Σ_2^1 formula ψ such that the formulas φ, ψ remain equivalent in any forcing extension.

Proposition 2.2 *The classes of infinite time decidable, semidecidable, and co-semidecidable sets lie within the projective hierarchy as follows.*

$$\begin{array}{ccccccc} \Sigma_1^1 & & & & sD & & \\ & \subset & & & \subset & & \text{Abs } \Delta_2^1 \subseteq \Delta_2^1 \\ & & D & & & & \\ & & \subset & & & & \\ \Pi_1^1 & & & & \widetilde{sD} & & \\ & & & & \subset & & \end{array}$$

Moreover, we have that $sD \cap \widetilde{sD} = D$.

Proof That Π_1^1 sets are infinite time decidable follows from the fact that an infinite time Turing machine can detect whether a given relation is wellfounded (see [8, Count-Through Theorem]). That every sD set is Δ_2^1 is shown in [8, Complexity Theorem], but we briefly sketch the argument. The idea is that any run of an infinite time computation can be coded by a real, namely, a code for a well-ordered sequence $\langle r_\alpha \rangle$, where each r_α is just a code for the configuration of the machine at stage α . It is Π_1^1 to check that a given real codes a well-order, and hence it is Π_1^1 to check that a given real codes a computation history.

Now, if A is semidecidable, then for some program e we have that $x \in A$ if and only if there exists a real code for a halting computation history for e on input x , and hence $A \in \Sigma_2^1$. But also $x \in A$ if and only if every real coding a *settled* run of the program e on input x shows that it accepts x . (A snapshot is said to be settled if and only if it shows that the program halts or is caught in a strongly repeating infinite loop and hence cannot halt.) This shows that $A \in \Pi_2^1$, and since our Σ_2^1 and Π_2^1 descriptions of A are absolutely equivalent, we have that $sD \subset \text{Abs}\Delta_2^1$.

Finally, $sD \cap \widetilde{sD} = D$ holds for the usual reason: given a program which halts if and only if $x \in A$ and a program which halts if and only if $x \in 2^\omega \setminus A$, then one can decide if $x \in A$ by running them simultaneously and outputting “yes” or “no” depending on which one halts. \square

Note that the inclusions in Proposition 2.2 are proper (except, consistently, the last), since by the classical diagonalization argument the halting set

$$H := \{(x, p) : p \text{ halts on input } x\}$$

is infinite time semidecidable but not infinite time decidable. It follows that the complement $2^\omega \times \omega \setminus H$ is a set which is absolutely Δ_2^1 but not semidecidable. It follows that the function which maps each $x \in 2^\omega$ to its infinite time (*lightface*) jump

$$x^\nabla := \{p : p \text{ halts on input } x\}$$

is not infinite time computable. Although one might expect that the jump function is semicomputable, we shall see shortly that this is not the case. However, there is a

Finally, suppose that both A and A^c are sE . Let e be a program which eventually stabilizes if and only if the input $x \in A$ and let q be a program which eventually stabilizes if and only if $x \notin A$. Then consider the program r which simulates both e and q , writing 1 whenever q changes its output, and writing 0 whenever e changes its output. r does not change its output until either of these events occurs. Clearly, r will eventually write 1 if and only if $x \in A$ and it will eventually write 0 if and only if $x \notin A$. \square

The relationship between the corresponding classes of functions is slightly different.

Proposition 2.5 *A function f is infinite time computable if and only if it is both infinite time eventually computable and infinite time semicomputable.*

It follows that the jump function $x \mapsto x^\nabla$ is indeed not semicomputable.

Proof If f is infinite time eventually computable by program e and semicomputable by program q , then it can be computed by the program which simulates e , at each step using q to check to see if the value on the output tape for e is correct. \square

We have already observed that even the infinite time semieventually decidable sets lie within the class of absolutely Δ_2^1 sets. When it comes to functions, the absolutely Δ_2^1 property only extends to the infinite time eventually computable functions. Here, a function f is said to be absolutely Δ_2^1 if and only if its *diagram*

$$\text{diag}(f) := \{(x, s) \in 2^\omega \times 2^{<\omega} \mid s \subset f(x)\}$$

is absolutely Δ_2^1 . By Theorem 3.7 in the next section, not every semicomputable function is absolutely Δ_2^1 .

Theorem 2.6 *Every infinite time eventually computable function is absolutely Δ_2^1 .*

Proof Let f be a function which is infinite time eventually computable using the program p , and (x, s) be given. We can eventually decide whether $s \subset f(x)$ by simulating p on input x and checking at each stage whether s is contained in the output. \square

The following result will be of fundamental importance in later sections.

Corollary 2.7 *Every infinite time eventually decidable set is measurable. Every infinite time eventually computable function is a measurable function.*

When we speak of measure, we are of course referring to the natural coin-flipping probability measure on 2^ω , also called the Lebesgue or Haar measure. It is just the ω -fold product of the $\left\{\frac{1}{2}, \frac{1}{2}\right\}$ measure on $\{0, 1\}$.

Proof By [14, Exercise 14.4] every provably Δ_2^1 set is measurable. It is not difficult to check that the proof also shows that every absolutely Δ_2^1 set is measurable as well. It follows that every infinite time eventually decidable set is measurable. Next, if f is an infinite time eventually computable function, it follows from Theorem 2.6 that for every open $U \subset 2^\omega$, $f^{-1}(U)$ is absolutely Δ_2^1 . Hence $f^{-1}(U)$ is measurable, and so f is measurable. \square

3 Infinite Time Effective Reductions

We are now ready to begin our generalization of Borel reducibility to the infinite time computability context. We shall introduce several classical results of Borel equivalence relation theory in turn and inquire how they are transferred or transformed to the infinite time computability context. And we shall also introduce several of the natural equivalence relations that we aim to fit into our new hierarchy. In the following section, we will begin to treat the infinite time analogue of the countable Borel equivalence relations.

Recall that if E, F are equivalence relations on 2^ω , then f is a reduction from E to F if and only if it satisfies

$$x E y \iff f(x) F f(y).$$

We say that E is Borel reducible to F , written $E \leq_B F$, if there is a Borel reduction from E to F . We propose to focus on the following generalizations of the reduction concept to the context of infinite time computability, corresponding to the two notions of computability that we have discussed.

- The relation E is infinite time *computably reducible* to F , written $E \leq_c F$, if there is an infinite time computable reduction from E to F .
- The relation E is infinite time *eventually reducible* to F , written $E \leq_e F$, if there is an infinite time eventually computable reduction from E to F .

To begin with some elementary considerations from the Borel theory, let $\Delta(1), \Delta(2), \dots, \Delta(\omega)$ denote arbitrary but fixed Borel equivalence relations with $1, 2, \dots, \omega$ classes, respectively. Then $\Delta(1) <_B \Delta(2) <_B \dots <_B \Delta(\omega)$, and, moreover, these are the simplest relations in the sense that for any E with infinitely many classes, $\Delta(\omega) \leq_B E$. The next least complex Borel equivalence relation is the *equality relation* on 2^ω , sometimes denoted $\Delta(2^\omega)$ or simply $=$.

Theorem 3.1 (Silver dichotomy) *If E is a Borel (or even Π_1^1) equivalence relation then either E has at most countably many classes or else $= \leq_B E$.*

Equivalence relations E which are Borel reducible to $=$ are called *smooth* or *completely classifiable*, since the corresponding reduction function shows how to concretely compute complete invariants for E . One step further up the hierarchy, one finds the *almost equality* relation E_0 , which is defined by $x E_0 y$ if and only if $x(n) = y(n)$ for almost all n .

We now present a proof that E_0 is not Borel reducible to $=$. This will be the first example of a proof that there cannot be a Borel reduction from E to F which overshoots and shows more. In this case, it shows that there cannot be a measurable reduction from E to F , and hence there cannot be an infinite time decidable or even provably Δ_2^1 such reduction. We shall discuss this phenomenon further in Section 5.

Proposition 3.2 *There is no measurable reduction from E_0 to equality $=$, and hence $= <_c E_0$.*

Proof Suppose that f is a measurable reduction from E_0 to $=$. Then for every $U \subset 2^\omega$, $f^{-1}(U)$ is closed under E_0 equivalence; that is, it is closed under finite modifications. Such a set is called a “tail set,” and a direct argument shows that such sets have measure 0 or 1. Letting U run over the basic sets, we obtain that f is constant on a set of measure 1. But f is countable-to-one, and since the measure is nonatomic, this is a contradiction. \square

We presently discuss a second dichotomy theorem (see [9]).

Theorem 3.3 (Glimm-Effros dichotomy) *If E is any Borel equivalence relation then either E is smooth or else $E_0 \leq_B E$.*

Neither the Silver dichotomy nor the Glimm-Effros dichotomy can hold in the case of infinite time decidable equivalence relations and infinite time computable reductions for the simple reason that there exist infinite time computable equivalence relations which necessarily have \aleph_1 many classes. But it is conceivable that this is the only obstruction, and many questions about E_0 and infinite time computable equivalence relations remain open.

Question 3.4 *Do any useful generalizations of the Silver dichotomy or Glimm-Effros dichotomy hold in the case of infinite time decidable equivalence relations and infinite time computable reductions?*

One might ask if there is any difference whatsoever between the Borel and infinite time computable theories. Of course, not every infinite time computable reduction can be replaced by a Borel reduction. To give a trivial counterexample, consider an infinite time decidable equivalence relation E with just two non-Borel classes: clearly, in this case we have $E \leq_c \Delta(2)$ and $E \not\leq_B \Delta(2)$. Of course, such counterexamples must be pervasive in the hierarchy, for instance, by replacing one of those equivalences classes with another entire equivalence relation. In Proposition 3.6 we shall give a *naturally occurring* pair of equivalence relations such that $E \leq_c F$ and $E \not\leq_B F$. However, our example will be of high descriptive complexity, and so we are left with the following interesting problem.

Question 3.5 *Are there Borel equivalence relations E, F such that $E \leq_c F$ but $E \not\leq_B F$?*

For an example of such E and F of higher complexity, we consider the following two equivalence relations.

- Let $x \cong_{WO} y$ if and only if x and y , thought of as codes for binary relations on ω , code isomorphic well-orders on ω .
- Let $x E_{ck} y$ if and only if $\omega_{1,ck}^x = \omega_{1,ck}^y$, that is, if and only if x and y can write the same ordinals in ω steps.

Note that \cong_{WO} is the first example of an equivalence relation whose domain is not all of 2^ω . Indeed, $x \cong_{WO} y$ only makes sense for those $x, y \in 2^\omega$ which code a well-order. This issue of an equivalence relation that is merely partial doesn't arise in Borel equivalence relations since the domain of any Borel (partial) equivalence relation is a standard Borel space in its own right, and hence can be assumed to be total. For the purposes of this paper, when considering infinite time computable reductions between partial relations, we shall require the domain of the reduction (that is, the domain of the source relation) to be infinite time decidable.

Proposition 3.6 *The equivalence relations E_{ck} and \cong_{WO} are infinite time computably bireducible. On the other hand, E_{ck} is not Borel reducible to \cong_{WO} .*

Since one does not typically consider Borel reductions with non-Borel domain, one might say that E_{ck} is Borel *incomparable* with \cong_{WO} (admittedly in a weak sense). In fact, if we were to think of \cong_{WO} as a relation on all of 2^ω by adding an equivalence class for the non-well-orders, then it is again Borel incomparable with E_{ck} .

Proof of Proposition 3.6 Results in [8] show that for any real x , the ordinal $\omega_{1,ck}^x$ is infinite time writable from parameter x , and this algorithm is uniform in x . So there is an infinite time computable function f such that $f(x)$ is a real coding the ordinal $\omega_{1,ck}^x$. This function is therefore a reduction from E_{ck} to \cong_{WO} .

Next we show that \cong_{WO} is infinite time reducible to E_{ck} . Let y be a code for an ordinal α . We shall compute $x = f(y)$ depending only on α and such that ω_1^x is equal to the α th admissible ordinal δ . First, given a code z for an ordinal β we can always find its admissible successor (the least admissible above β). To see this, note that it must be bounded by $\omega_{1,ck}^z$. So for each $\beta < \gamma \leq \omega_{1,ck}^z$ we simply build L_γ and check to see that it satisfies the KP axioms. Now we can iterate this α times to find the α th admissible ordinal δ . Next build L_δ , and search inside it to find the L -least x such that $\omega_{1,ck}^x = \delta$. Clearly, x depends only on α and not the given code y , and so we have built the desired reduction.

It remains only to show that there can be no Borel reduction from E_{ck} to \cong_{WO} . Suppose that f is such a reduction. It takes values in WO and since E_{ck} has ω_1 many equivalence classes, the range of f must code unboundedly many ordinals. By the boundedness theorem (see [16, Theorem 35.23]), $\text{im}(f)$ is not Σ_1^1 and hence f is not Borel. \square

Gao had noted [4, Section 9.2] that there exists a Δ_2^1 reduction from \cong_{WO} to E_{ck} , but that the study of Δ_2^1 reducibility is problematic. Thus Proposition 3.6 resolves this by showing that the reduction from \cong_{WO} to E_{ck} is infinite time computable. To see that the Δ_2^1 reductions can pose difficulties, we now show that in L the Δ_2^1 functions, and indeed the semicomputable functions, collapse a large portion of the hierarchy of equivalence relations.

Theorem 3.7 *If $V = L$, then whenever E is an infinite time decidable equivalence relation, there exists an infinite time semicomputable function $f: 2^\omega \rightarrow 2^\omega$ such that $x E y$ if and only if $f(x) = f(y)$.*

Proof Following the L -code argument of [5, Theorem 38], given $x \in 2^\omega$ we shall encode its equivalence class by a pair of reals. Let $\alpha < \omega_1$ be least such that L_α contains a member of the E -equivalence class of x , and $L_\alpha \models$ “some (fixed) large fragment of ZFC and ω_1 exists.” The idea is that L_α is large enough that all computations on reals of L_α halt or repeat in fewer than α steps.

Let $\beta > \alpha$ be least such that β is countable in $L_{\beta+1}$ and let $w \in L_{\beta+1}$ be the L -least real coding β . Finally let z be the L -least real which is E -equivalent to x . Then by our choice of α , we have $z \in L_\alpha$. Now, $f(x) := w \oplus z$ is the code we seek.

We clearly have $x E y$ if and only if $f(x) = f(y)$, but we must verify that f is infinite time semicomputable. That is, a machine must recognize given $(x, w_0 \oplus z_0) \in f$, whether $w_0 = w$, $z_0 = z$ as defined above. The machine first checks to see that w_0 codes an ordinal, and using this ordinal as β it constructs L_α and checks to see that $L_\alpha \models z_0 = z$. Lastly, note that L_α is correct about this since it has access to all computations on its reals. \square

On the other hand, we have seen that infinite time effective sets and functions derive many of their properties from the fact that they are *absolutely* Δ_2^1 . It is therefore natural to study absolutely Δ_2^1 reducibility, as we shall do in Section 5. One might ask whether there is any sense in which the absolutely Δ_2^1 sets and functions are

effective. The following result sheds doubt on this by showing first that this question cannot be separated from that of whether there is a sense in which *all* Δ_2^1 sets are effective.

Proposition 3.8 *There is a forcing extension of the universe in which every Δ_2^1 set is absolutely Δ_2^1 , and indeed, in which every equivalent pair of Σ_2^1 and Π_2^1 definitions remains equivalent after any further forcing.*

Proof Let us first show that the desired situation holds under the Maximality Principle, which is the scheme asserting that any forcibly necessary set-theoretic statement is already true (see [6], also [18]). A statement is forcibly necessary if it is forceable in such a way that it remains true in all further forcing extensions. If V satisfies the Maximality Principle and φ and ψ are Σ_2^1 and Π_2^1 assertions, respectively, which could become inequivalent in a forcing extension $V[G]$, then there is a real z in $V[G]$ such that $\varphi(z)$ differs from $\psi(z)$ in $V[G]$. Since these statements are each absolute to all further extensions of $V[G]$, this means that the inequivalence of φ and ψ is forcibly necessary over V and therefore true there by the Maximality Principle. Thus, under the Maximality Principle, any two Σ_2^1 and Π_2^1 assertions that are equivalent in V remain equivalent in all forcing extensions. In particular, every Δ_2^1 set in V is absolutely Δ_2^1 .

This argument makes use of only a small fragment of the Maximality Principle. And although it is proved in [6] that if ZFC is consistent, then there is a model of ZFC plus the Maximality Principle, it is also observed there that some models of ZFC have no forcing extensions with the Maximality Principle. Nevertheless, the main argument of [6] does show that every model of ZFC has a forcing extension with the Maximality Principle restricted to assertions of a given fixed set-theoretic complexity. Since we only used low projective complexity in the previous paragraph, there is a forcing extension of the universe in which every Δ_2^1 set is absolutely Δ_2^1 as described. (The forcing is simply an iteration, where one continues forcing until all possible inequivalences have been exhibited.) \square

On the other hand, there are models with Δ_2^1 sets which are not absolutely Δ_2^1 . For instance in L there is a Δ_2^1 well-ordering of the reals, though no model of ZFC has an absolutely Δ_2^1 well-ordering of the reals.

We close this section by introducing a number of equivalence relations which are of natural interest. The notation used here will be repeatedly referenced throughout the paper.

- Let $x E_{\text{set}} y$ if and only if x and y , thought of as countable sequences of reals, have the same range.
- Let $x \cong_{HC} y$ if and only if x and y , thought of as codes for hereditarily countable sets, are isomorphic. (Here, x is said to code a hereditarily countable set z iff, thinking of x as a binary relation on ω , we have $(\omega, x) \cong (tc(\{z\}), \in)$.)
- Let $x \cong y$ if and only if x and y , thought of as codes for countable structures in a countable language, are isomorphic.
- Let $x E_\lambda y$ if and only if $\lambda^x = \lambda^y$, that is, if and only if x and y can write the same set of ordinals. Similarly, define $x E_\zeta y$ if and only if x and y can eventually write the same ordinals, and $x E_\Sigma y$ if and only if x and y can accidentally write the same ordinals.
- Let $x \equiv_T y$ if and only if x, y lie in the same classical Turing degree.

- Let $x \equiv_{\text{arith}}$ y if and only if x, y lie in the same arithmetic degree.
- Let $x \equiv_{\text{hyp}}$ y if and only if x, y lie in the same hyperarithmetic degree.
- Let $x \equiv_{\infty}$ y if and only if x and y are infinite time computable from one another (that is, lie in the same infinite time *degree*).
- Let $x \equiv_{e\infty}$ y if and only if x and y are infinite time eventually computable from one another (that is, lie in the same infinite time *eventual degree*).
- Let $x J y$ if and only if x, y have equivalent infinite time jumps, that is, $x^{\nabla} \equiv_{\infty} y^{\nabla}$.

Some reductions between these equivalence relations are already apparent. For instance, given a countable sequence of reals $\langle a_n \rangle$, it is not difficult to construct an HC-code for the set $\{a_n\}$, and hence E_{set} is infinite time computably reducible to \cong_{HC} . Next, \cong_{HC} and \cong_{WO} are infinite time computably reducible to \cong , both being restrictions of \cong to an infinite time decidable subset of the domain. On the other hand, \cong is infinite time computably reducible to \cong_{HC} since a countable structure in a countable language can trivially be coded as a hereditarily countable set. Finally, it is easy to see that the function $x \mapsto x^{\nabla}$ is an eventual reduction from J to \equiv_{∞} . Many more details of the interrelationships (with respect to infinite time computable and infinite time eventually computable reducibility) shall be examined as the exposition unfolds.

4 Enumerable Equivalence Relations

The classical Borel equivalence relation theory has placed a major focus on the countable Borel equivalence relations, and the investigation of this natural subhierarchy of the hierarchy of all equivalence relations has led to some of the most fruitful work (see, for instance, [12]). Here, an equivalence relation E is said to be *countable* if every E -equivalence class is countable. Not only do these relations include many of the most natural examples, but some of the most powerful methods in the theory apply only to the countable relations. The situation is rather reminiscent of the focus in computability theory on the c.e. Turing degrees as a subhierarchy of the hierarchy of all Turing degrees.

We shall find the following characterization of the countable Borel equivalence relations useful. Namely, E is countable Borel if and only if it admits a *Borel enumeration*, that is, a Borel function f such that $f(x) = \langle x_0, x_1, \dots \rangle$ codes some enumeration of $[x]_E$, for all x . (This is a consequence of the Lusin-Novikov theorem, [16, Theorem 18.10].) The natural extension of the class of countable Borel equivalence relations to the infinite time computable context simply generalizes this enumerability concept.

Definition 4.1

- A countable equivalence relation E is infinite time *enumerable* if it admits an infinite time computable enumeration function, that is, a function f for which $f(x) = \langle x_0, x_1, \dots \rangle$ enumerates $[x]_E$ for all $x \in 2^{\omega}$.
- Similarly, E is infinite time *eventually enumerable* if it admits an infinite time eventually computable enumeration function.

Recall that if Γ is any group of bijections of 2^{ω} , we can define the corresponding *orbit equivalence relation* E_{Γ} by

$$x E_{\Gamma} y \iff \Gamma x = \Gamma y.$$

By a theorem of Feldman and Moore [2], E is a countable Borel equivalence relation if and only if there exists a countable group Γ of Borel bijections of 2^ω such that $E = E_\Gamma$. Our first observation is that the infinite time enumerable relations enjoy an analogous property.

Theorem 4.2 *An equivalence relation E is infinite time enumerable if and only if there exists a countable group Γ of infinite time computable bijections of 2^ω such that E is precisely the induced orbit equivalence relation E_Γ . The analogous result holds for the infinite time eventually enumerable equivalence relations.*

Proof Suppose first that there exists such a group Γ . Write $\Gamma = \langle \gamma_n \rangle$, and let r be a real code for a sequence $\langle r_n \rangle$ such that each r_n codes a program which computes the function γ_n . We claim that E_Γ is infinite time enumerable in the real r . Indeed, on input x , a program can simply use r to simulate each γ_n on input x , and collect the values $\gamma_n(x)$ into a sequence.

Conversely, suppose that E is infinite time enumerable. By the proof of the classical Feldman-Moore theorem, it suffices to establish the conclusion of the Lusin-Novikov theorem, namely,

(*) E can be expressed as a countable union of graphs of infinite time computable partial functions.

For this, let f be an infinite time computable function which witnesses that E is infinite time enumerable, that is, for every $x \in 2^\omega$, $f(x)$ is a code for the E -class of x . Letting $f_n(x)$ denote the n th element of $f(x)$, we have that $E = \cup f_n$. This completes the proof. □

Proposition 4.3 *The class of infinite time enumerable equivalence relations lies properly between the countable Borel equivalence relations and the countable infinite time decidable equivalence relations.*

Proof That every countable Borel equivalence relation is infinite time enumerable follows from the previous theorem, and it is immediate from the definition that every infinite time enumerable equivalence relation is countable and infinite time decidable.

We now give an example of a countable infinite time decidable equivalence relation which is not infinite time enumerable. For each $x \in 2^\omega$, we let c^x denote the lost melody real relative to x . Recall that c^x is a real such that $\{c^x\}$ is (lightface) infinite time decidable in x and yet c^x is not infinite time writable in x . It follows that the function $f(x) := x \oplus c^x$ is infinite time semicomputable but not infinite time computable, even from a real parameter. Now, we let $x E y$ if and only if there exists n such that $x = f^n(y)$ or $y = f^n(x)$. Since f is injective, E is an equivalence relation. Moreover, it is easy to see that E is countable and infinite time decidable. However, E cannot be infinite time enumerable in the parameter z , for then c^z would be infinite time writable in z , a contradiction. Indeed, E cannot even be accidentally enumerable.

For an example of an infinite time enumerable equivalence relation which is not Borel, we shall use hyperarithmetic equivalence \equiv_{hyp} . Recall that $x \equiv_{\text{hyp}} y$ if and only if $x \in \Delta_1^1(y)$ and $y \in \Delta_1^1(x)$. It follows from the proof of Theorem 2.1 that $x \equiv_{\text{hyp}} y$ if and only if x is infinite time computable from y in fewer than $\omega_{1,ck}^y$ steps

and y is infinite time computable from x in fewer than $\omega_{1,ck}^x$ steps. (Recall that $\omega_{1,ck}^y$ denotes the supremum of the ordinals computable in the ordinary sense from y .)

Since $\omega_{1,ck}^x$ is infinite time computable from x , the equivalence relation \equiv_{hyp} is clearly infinite time enumerable. But suppose, toward a contradiction, that \equiv_{hyp} is Borel. Then since \equiv_{hyp} is also countable, there exists a Borel function f such that for all x , $f(x)$ codes $[x]_{\equiv_{\text{hyp}}}$. By Theorem 2.1, there exists a program e in a parameter z and an ordinal α such that on any input x , e computes $f(x)$ in fewer than α steps. Replacing z with a more complicated real if necessary, we may suppose that $\alpha \leq \omega_{1,ck}^z$. Now, using e it is easy to write a program which first enumerates $[z]_{\equiv_{\text{hyp}}}$, then diagonalizes against this set to write a real $r = z \oplus d$ such that $r \notin [z]_{\equiv_{\text{hyp}}}$. Since z is quickly writable from r , we must have that r isn't writable from z in fewer than $\omega_{1,ck}^z$ steps. This is a contradiction, because we have just described a program which does so. \square

The infinite time eventually enumerable equivalence relations are easily seen to be infinite time semidecidable, but as the next proposition (from [8, Theorem 5.7]) shows, not necessarily infinite time enumerable or even infinite time decidable.

Proposition 4.4 *\equiv_{∞} is infinite time eventually enumerable but not infinite time decidable.*

Proof To show that \equiv_{∞} is infinite time semidecidable one can, on input x, y , just simulate all programs on input x and see if any of them writes y , and vice versa.

Now, suppose toward a contradiction that \equiv_{∞} is infinite time decidable in the real parameter z . We shall use this to decide the halting problem in z , that is, the real $z^{\nabla} = \{e : \varphi_e^z(0) \text{ halts}\}$. Consider the program which attempts to compute this set. It runs all programs simultaneously on input 0, and each time one halts, its output is added to an accumulating set x . Additionally, it checks at each stage whether $x \oplus z \equiv_{\infty} z$ and halts if this does not hold.

Note that this program halts, since after some stage all programs which halt have halted. Moreover, at this moment the approximation is correct and so certainly $x \oplus z \not\equiv_{\infty} z$. It may halt earlier than this, but it must halt with some real x such that $x \oplus z \not\equiv_{\infty} z$. Hence from z it has computed a real strictly more complex than z , a contradiction. \square

In particular, we have the following consequence.

Corollary 4.5 *The relation \equiv_{∞} doesn't computably reduce to any infinite time decidable equivalence relation.*

Proposition 4.6 *The relation $\equiv_{e\infty}$ is infinite time eventually decidable but not infinite time eventually enumerable.*

That the infinite time eventual degree relation $\equiv_{e\infty}$ is infinite time eventually decidable is due to Welch. This result was very surprising to experts in the area; indeed Proposition 4.4 shows that the corresponding infinite time Turing degree relation \equiv_{∞} is not infinite time decidable.

Proof First, to see that the set of infinite time eventually writable reals is not infinite time eventually enumerable, suppose that e is a program with oracle z which on input x , eventually writes a code for the $\equiv_{e\infty}$ -class of x . Then consider the diagonalization program q which simulates e on z , and at each stage of simulation writes

a real which is not in the set coded on the output tape of e . Then q eventually writes a real which is not in the $\equiv_{e\infty}$ -class of z , a contradiction.

Now, to show that $\equiv_{e\infty}$ is infinite time eventually decidable, we shall actually just show that the set of infinite time eventually writable reals is infinite time eventually decidable. The proposition follows, since this argument relativizes to show that given z , the set of reals infinite time eventually writable in z is infinite time eventually decidable. Following the infinite time Turing machine literature, we denote by λ the supremum of the writable ordinals (which is the same as the supremum of the clockable ordinals), by ζ the supremum of the infinite time eventually writable ordinals, and by Σ the supremum of the infinite time accidentally writable ordinals. Results in [8] establish that $\lambda < \zeta < \Sigma$, and Welch ([22], [21], see also [7, Theorem 1.1]) has proved, moreover, that $L_\lambda <_{\Sigma_1} L_\zeta <_{\Sigma_2} L_\Sigma$, and, furthermore, these ordinals are characterized as least having this property. This key result is now known as the λ - ζ - Σ theorem. Welch proved that every infinite time Turing machine computation either halts in time before λ or repeats its stage ζ configuration at Σ . Any computation that eventually stabilizes, reaches its stabilizing configuration before ζ , and the universal computation simulating all programs on trivial input repeats the stage ζ configuration at stage Σ for the first time. Because of this, it is infinite time decidable whether a given real codes the ordinal Σ , since the machine need merely check that it does indeed code a well-order, and that the universal computation, when simulated for that many steps, exhibits this repeating phenomenon exactly at that stage. These facts relativize easily to a real parameter.

Now, on input x , we can eventually decide whether it is infinite time eventually writable by the following algorithm. First, write a preliminary default “No” on the output tape. Next, simulate the universal computation, and search to see if x is ever shown to be accidentally writable. If so, change the answer on the output tape provisionally to “Yes,” and then run the universal program with parameter x to see if there is an x -writable real coding the ordinal Σ . By the remarks in the previous paragraph, any instance of this is infinite time decidable. If Σ is ever found to be x -writable, then change the answer finally back to “No” and halt; otherwise, keep searching.

Let’s argue that this algorithm works. If x is eventually writable, then it will appear on the tape before stage ζ , and so we shall pass the first hurdle, where the answer was changed provisionally to “Yes.” But since x is eventually writable, it follows that Σ cannot be x -writable, since if it were, then Σ would be accidentally writable, contradicting the fact that it is larger than all accidentally writable ordinals. Thus, in this case we shall never pass the second hurdle, and so our final answer will stabilize on “Yes,” as desired. If x is not eventually writable and also not accidentally writable, then the algorithm will never pass the first hurdle, and so the algorithm will stabilize on the first “No,” as desired. Finally, if x is accidentally writable, but not eventually writable, then x appears accidentally on the universal computation, but not before time ζ (since otherwise it would be eventually writable). So it appears at some point between ζ and Σ . In this case, the ordinal ζ is below the supremum of the x -clockable ordinals, and since the supremum of the x -clockable and x -writable ordinals is the same, it follows that ζ is x -writable. From this, it follows that there are ordinals above Σ that are x -clockable, since with oracle x we can run the universal computation, look exactly at the stage ζ configuration, and then wait until stage Σ ,

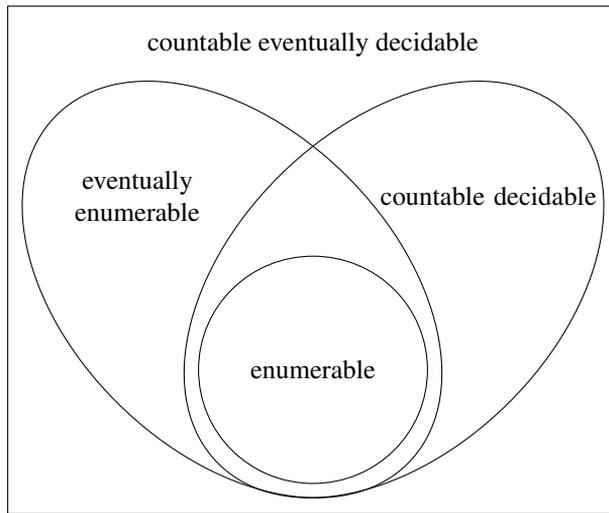


Figure 1 The relationships between various classes of countable equivalence relations.

when this configuration first repeats. Thus, Σ is also x -writable, and so the algorithm will pass the final hurdle, changing the answer to “No,” and halting, as desired. \square

The relationships between various classes of countable equivalence relations are shown in Figure 1. Each of the inclusions is proper; to establish this fully, it remains only to show the following.

Proposition 4.7 *There exists an infinite time eventually enumerable, infinite time decidable equivalence relation which is not infinite time enumerable.*

Proof The equivalence relation we give will not be defined on all of 2^ω , but rather on an infinite time decidable subset. Let A be the set of codes for triples (x, p, y) satisfying $x \in 2^\omega$, p is a program, y is a real coding the computation history of $\varphi_p(x)$, showing that it halts, and y is the first x -accidentally writable such code encountered in the standard enumeration of the x -accidentally writable reals by a universal machine. This set is infinite time decidable, because given (x, p, y) , we can first check if y does code the computation of $\varphi_p(x)$, showing it to halt, and if this test is passed, then we start up the enumeration of all x -accidentally writable reals, and see if y is the first such code encountered. Since we know that $\varphi_p(x)$ does halt, we are guaranteed to find such a code, and the only question is if y is the first one or not.

Now, define the relation E on A by $(x, p, y) E (x', p', y')$ if and only if $x = x'$. This is clearly a decidable equivalence relation. Also, it is countable, since for any x , the only things that can be equivalent to (x, p, y) are other (x, q, z) , where z is determined by x and q . This relation is eventually enumerable, because for any x , we can eventually write out the halting problem x^∇ , which tells us which p have halting computation $\varphi_p(x)$, and for each of these we can go find the first x -accidentally writable real y coding the computation, and then assemble all these triples into a real to write out the equivalence class of (x, p, y) . Finally, E is not infinite time

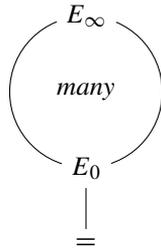


Figure 2 The countable Borel equivalence relations.

enumerable, since if it were, we would be able to write out the halting problem x^∇ just from the code of any halting computation (e.g., from a trivial computation), which is impossible. \square

We now turn toward an analysis of the structure of the infinite time enumerable equivalence relations. We begin by describing the most basic structure theory of the countable Borel equivalence relations. First, we have already seen that it is a consequence of Silver’s theorem that the equality relation $=$ is the minimum countable Borel equivalence relation. The relations E which are Borel reducible to $=$ are called *smooth*. By the Glimm-Effros dichotomy (Theorem 3.3), E_0 is the next-least countable Borel equivalence relation, in the strong sense that E_0 is Borel reducible to *any* nonsmooth Borel equivalence relation. Lastly, and somewhat surprisingly, there exists a *universal* countable Borel equivalence relation, denoted E_∞ . It is realized as the orbit equivalence relation induced by the left-translation action of the free group F_2 on its power set.

There were initially very few countable Borel equivalence relations known to lie in the interval (E_0, E_∞) . It is a fundamental result of Adams and Kechris [1, Theorem 1] that there exists a sequence $\{AK_\alpha\}$ of continuum many pairwise incomparable countable Borel equivalence relations. In summary, we have that the countable Borel equivalence relations are organized as in Figure 2.

We would like to develop an analogous picture for the infinite time enumerable relations. We first consider the question of whether the Silver dichotomy holds for the infinite time enumerable relations, that is, whether $=$ is the least complex such relation.

Theorem 4.8

- *There is a perfect set of infinite time eventual degrees.*
- *If E is infinite time eventually enumerable then $= \leq_B E$.*

The proof hinges on an unpublished result of Welch, which is stated below in Theorem 4.9. We shall present a new proof which is due to Hamkins. Welch’s original proof (based on methods from [20]) made use of the theory of forcing over models of very weak set theories, an approach that has also been suggested by Lubarsky and the referee of this paper. The idea is to develop and analyze the general theory of forcing over models of such weak theories, and in particular to analyze the complexity of the forcing relation over such structures, which allows one to deduce that $\lambda^c = \lambda$ and so on. This general theory succeeds more easily in the case of admissible structures,

that is, models of KP, and although L_Σ is not admissible, it is nevertheless a union of admissible structures, which makes things easier in this application. Nevertheless, in part in order to avoid the difficulties and uncertainty of the general development, we have chosen a more concrete approach which does not require that general theory.

Theorem 4.9 (Welch) *If c is an L_Σ -generic Cohen real, then $\lambda^c = \lambda$, $\zeta^c = \zeta$, and $\Sigma^c = \Sigma$.*

We remark that our proof of this result will easily relativize. That is, for any real z , if c is an $L_{\Sigma^z}[z]$ -generic Cohen real, then we shall have $\lambda^{z+c} = \lambda^z$, $\zeta^{z+c} = \zeta^z$, and $\Sigma^{z+c} = \Sigma^z$. Admitting this result, let us show how to complete the proof of Theorem 4.8.

Proof of Theorem 4.8 We begin by arguing that there is a perfect set of eventual degrees, from which it follows that $= \leq_B \equiv_{e\infty}$ (in fact, via a *continuous* reduction function). Since L_Σ is countable, there exists a perfect set \mathcal{G} of reals which are mutually generic over L_Σ . It suffices to show that for $g, g' \in \mathcal{G}$, if $g \neq g'$ then $g \not\equiv_{e\infty} g'$. Indeed, since g, g' are mutually generic we have $g' \notin L_\zeta[g]$. Furthermore, since $\zeta^g = \zeta$, it follows that $g' \notin L_{\zeta^g}[g]$; in other words, g' is not infinite time eventually writable from g .

Next, let $x \equiv_{e\infty}^z y$ if and only if x and y are infinite time eventually writable from one another using the parameter z . Then by our earlier remarks, Theorem 4.9 relativizes and we in fact have $= \leq_B \equiv_{e\infty}^z$ for any $z \in 2^\omega$.

Now, let E be an arbitrary infinite time eventually enumerable relation. There exists an enumeration function for E which is eventually computable in a parameter z , and so we have $E \subset \equiv_{e\infty}^z$. By the previous paragraph, there exists a Borel reduction f from $=$ to E_{∞}^z , and this f will also serve as a reduction from $=$ to E . \square

It is worth remarking that this argument also gives a reduction (the same function) from $=$ to J . We now return to the proof of Theorem 4.9. Recall the result of Welch we mentioned earlier, that for any z , every computation in z either halts before λ^z or repeats the stage ζ^z configuration by stage Σ^z , and, moreover, that the universal computation in z repeats for the first time with this pair of ordinals.

Proof of Theorem 4.9 Our strategy will be to show that for any infinite time Turing machine program e , $\varphi_e(c)$ repeats from stage ζ to stage Σ . Applying this to the case when e is the universal program, this implies that $\zeta^c = \zeta$ and $\Sigma^c = \Sigma$. After this, we shall argue separately that $\lambda^c = \lambda$.

The main idea is that instead of carrying out the computation $\varphi_e(c)$, which only exists in a world having c , we shall instead carry out a Boolean-valued computation using only the canonical name \dot{c} for the Cohen generic, which is coded by a real in the ground model. The inspiration here is that if c is fully V -generic, then every fact or aspect about the computation $\varphi_e(c)$, whether a given cell shows a 0 or 1 at a particular ordinal stage or whether the head is on a particular cell at a particular stage, is forced by some finite piece of the generic real c . This is the magic of forcing. We shall simply design a computation that keeps careful track of this information.

Let us now describe the Boolean computation or simulation of $\varphi_e(\dot{c})$. We embark on a computation that simulates $\varphi_e(\dot{c})$ by computing exactly what information about this computation is forced by which conditions. At each simulated stage of computation, the algorithm keeps track of the values of the cells, the head position

and the machine state, not with certainty, but with its corresponding Boolean value. That is, for each cell in the simulated computation, we reserve space in our actual computation to keep track of the conditions $p \in \mathbb{P}$ that force the current value of this cell to be 0 or to be 1. Similarly, we also keep track of the conditions that force that the head is currently at this cell, and for each state in the program e we keep track of the conditions that force that the simulated machine is currently in that state.

Initially, our simulation data should specify that all conditions force that the head is on the left-most cell and in the start state, and that all the cells on the work and output tapes are 0. For the input tape, which we intend to hold the generic real c , for each cell j we say that a condition p forces that the j th cell has value $p(j)$, if this is defined. Every condition forces that the cells on the scratch and output tapes are all initially 0.

At successor stages of simulation, we can easily update this data so as to carry out the simulated computational step. For example, if p forces that the head is at a certain position, reading a certain value and in a certain state, then we can adjust our data for the next step so that p forces the appropriate values and head position after one step of the program e . There is a subtle tidying-up issue, in that it could happen at a successor step that after this update, although previously a condition q did not force, say, a certain head position, nevertheless now $q \hat{\ } 0$ and $q \hat{\ } 1$ both force the same head positions (perhaps having arrived from different directions). In this case, we would want to say that q also forces this head position. More generally, if the collection of conditions forcing a particular feature (cell value, head position, or state) is dense below a given condition q , then in our update procedure we tidy up our data to show that q also forces this feature.

Let us now explain how to update the data at limit stages of computation. Of course, at any simulated limit stage, we want every condition to force that the head is now on the left-most cell in the limit state. It is somewhat more subtle, however, to update the cells on the tape correctly. The problem is that the data we now have available is the lim sup of the previous data for the cell values, which is not the same as the data for the lim sup of the cell values. Nevertheless, we will be able to recover the data we need. Note that p forces a particular cell value is 0 at limit time α if and only if there are densely many q below p such that for some $\beta < \alpha$, the condition q forces that this cell is 0 from β up to α . But we do have this information available in the lim sup of the previous data, since if q forced the cell was 0 from β up to α , then the limit of this data will continue to show that. Thus, we can correctly compute the correct forcing relation for the cell values on the tape at limit stages of the simulated computation. A simple inductive argument on the length of the computation now establishes that we have correctly calculated the forcing relation for the head position, machine states, and cell values at every stage of simulated computation.

If c is actually generic, then this Boolean-valued computation collapses to the actual computation of $\varphi_e(c)$ as follows. At every stage α , there are dense sets of conditions p forcing exactly where the head is, and what the state is, and what appears in each particular cell. By genericity, the generic filter will meet each of these dense sets, and so as far as conditions in c are concerned, the ghostly Boolean-valued computation follows along with the actual computation $\varphi_e(c)$. Indeed, we claim that if c is merely L_Σ -generic (meaning that c meets all predense sets for \mathbb{P} coded as elements of L_Σ), then the computation of $\varphi_e(c)$ at stage α is exactly what is forced

by some condition $p \in c$ in the Boolean-valued computation. This is certainly correct at the initial stage, and by induction it is preserved through successor stages and limits, because all the relevant dense sets are in L_Σ , and so c meets them as required in order to collapse the Boolean-valued computation.

The key observation now is that since the Boolean-valued computation is repeating from ζ to Σ , it follows that the true computation $\varphi_e(c)$ must also be repeating from ζ to Σ . And this is precisely what we had set out to prove. So we have established that $\zeta^c = \zeta$ and $\Sigma^c = \Sigma$.

We finally argue that $\lambda^c = \lambda$. Suppose that some e had the property that $\varphi_e(c)$ halts at some ordinal stage $\alpha > \lambda$. Then some condition p forces that $\varphi_e(c)$ halts at stage α . We may now run Boolean-valued computation and wait until p forces that halt is achieved. Since the simulated computation takes at least as long as the actual computation, this would allow us to halt beyond λ , a contradiction, since there are no clockable ordinals above λ . \square

A slew of questions follows. For instance, we have just seen in Theorem 4.8 that there is a perfect set of eventual degrees, and hence of infinite time computable degrees. It is natural to ask just how complex the infinite time Turing degree relations \equiv_∞ and $\equiv_{e\infty}$ actually are.

Question 4.10 *Does E_0 reduce (in any reasonable sense) to either \equiv_∞ or $\equiv_{e\infty}$?*

We have also just seen that a Silver dichotomy holds for infinite time enumerable relations. This leaves open the following related question.

Question 4.11 *Does a Glimm-Effros dichotomy hold for the infinite time enumerable equivalence relations? In other words, for any infinite time enumerable equivalence relation E do we have either $E \leq_c \Delta(2^\omega)$ or $E_0 \leq_c E$? (And similarly for infinite time eventually enumerable relations with respect to eventual reducibility.)*

We next address the question of whether there is a universal infinite time enumerable equivalence relation.

Theorem 4.12

- *If E is infinite time enumerable, then $E \leq_c E_\infty$.*
- *If E is infinite time eventually enumerable then $E \leq_e E_\infty$.*

Sketch of proof This is analogous to the proof that any countable Borel equivalence relation is Borel reducible to E_∞ . In that argument, the key point is that any countable Borel equivalence relation can be expressed as the orbit equivalence relation induced by the Borel action of a countable group. For our result, the key point is Theorem 4.2. \square

In particular, \equiv_∞ is eventually reducible to E_∞ . It is now natural to extend Question 4.10 to the following stronger statement.

Question 4.13 *Does E_∞ reduce (in any sense) to \equiv_∞ ? In other words, is \equiv_∞ universal infinite time eventually enumerable?*

Note that Slaman has shown that the arithmetic equivalence relation \equiv_{arith} on the reals is universal countable Borel. On the other hand, it is unknown whether \equiv_T is universal countable Borel. If it is, then the Martin Conjecture must fail. For a discussion of this question see [19].

Finally, we consider the question of whether there are incomparable infinite time enumerable relations. Indeed, it is not difficult to see from the proof of the Adams-Kechris theorem that there cannot even be a measurable reduction between any two AK_α . Hence, we obtain the following result for free.

Theorem 4.14 (Adams-Kechris) *There is a sequence $\{AK_\alpha\}$ of continuum many infinite time enumerable equivalence relations which are pairwise infinite time (eventually) computably incomparable.*

We conclude this section with a question regarding the following chain of refinements of \equiv_∞ .

Definition 4.15 For $\alpha < \omega_1$, let $x \equiv_\alpha y$ if and only if x and y are infinite time computable from one another (without parameters) by computations which halt in fewer than α steps. This is an equivalence relation whenever α is additively closed.

Proposition 4.16 *Each equivalence relation \equiv_α is countable and Borel.*

Proof The main point of interest is that \equiv_α are Borel. First note that \equiv_α is infinite time computable from an oracle for a real coding α , and it is easily seen that it is infinite time computable uniformly in at most $\alpha + \alpha$ steps. But by Theorem 2.1, any uniformly infinite time decidable set is Borel. \square

We have the equalities \equiv_0 is $\Delta(2^\omega)$ and \equiv_ω is E_0 . The union of the \equiv_α is again countable; it is precisely \equiv_∞ . Nothing is known about the rest of the \equiv_α for $\alpha < \omega_1$.

Question 4.17 *What is the structure of the \equiv_α under infinite time computable comparability? Are they linearly ordered with respect to infinite time computable reducibility?*

5 Some Tools for Showing Nonreducibility

In this section we will establish several nonreducibility results, that is, results which state that some equivalence relation E is not reducible to another equivalence relation F . As we have mentioned, many such nonreducibility results from the theory of Borel equivalence relations come from arguments which show that there cannot be an absolutely Δ^1_2 reduction from E to F . In this section, we give a survey of some of the nonreducibility results which apply also to Δ^1_2 reducibility. An account of further results of Δ^1_2 reducibility, including versions of the dichotomy theorems, can be found in [10, Chapter 9].

We begin with a sequence of absoluteness results which will pave the way for forcing arguments later on.

Proposition 5.1 *If A is an infinite time decidable set, then in any forcing extension there is an unambiguous interpretation of A and, moreover, it remains an infinite time decidable set.*

Proof If A is infinite time decidable by the program p and the real parameter z , define A of the forcing extension to be the set decided by p and z . To see that this is well-defined, suppose that programs p, q both compute A in the ground model. This is a Π^1_2 fact, and so by Shoenfield's absoluteness theorem, it remains true in the forcing extension. \square

We remark that the analogue of Proposition 5.1 holds for infinite time eventually decidable, infinite time semidecidable, and even for absolutely Δ_2^1 sets. Similarly, we have the following result.

Proposition 5.2 *If an equivalence relation E is infinite time enumerable, then E is infinite time enumerable in any forcing extension.*

Proof If E is infinite time enumerable, then there exists an infinite time computable f such that $f(x)$ codes $[x]_E$. Hence we have that for all $x, y \in 2^\omega$, the relation $x E y$ holds if and only if there exists $n \in \omega$ such that $y = f(x)_n$. This is a Π_2^1 assertion about the programs computing f and deciding E and hence it is absolute to forcing extensions. \square

Proposition 5.3 *Suppose that E, F are absolutely Δ_2^1 equivalence relations, and let f be an infinite time eventually computable reduction from E to F . Then in any forcing extension, f remains such a reduction.*

Proof By the remarks following Proposition 5.1, E, F , and f may be unambiguously interpreted in any forcing extension. Clearly, since E, F are Δ_2^1 , the statement

$$\forall x \forall y (x E y \leftrightarrow f(x) F f(y))$$

is Π_2^1 and hence absolute to forcing extensions. We must additionally check that f remains a total function in any extension. Indeed, suppose that the program e eventually computes the function f . Then f is total if and only if for every $x \in 2^\omega$ and every settled well-ordered sequence of snapshots according with e , the value $f(x)$ eventually appears on the output tape. This demonstrates that the assertion “ f is total” is Π_2^1 and hence it is absolute to forcing extensions. \square

We remark that by Theorem 3.7, Proposition 5.3 fails for infinite time semicomputable reductions f . On the other hand, the conclusion of Proposition 5.3 does hold for absolutely Δ_2^1 functions.

We now give several classical forcing methods for establishing nonreducibility results.

Definition 5.4 Let E be any equivalence relation. If \mathbb{P} is a notion of forcing then a \mathbb{P} -name τ is said to be a *virtual E -class* if the following hold:

- If G is \mathbb{P} -generic, then in $V[G]$, $\tau_G \not\equiv x$ for any $x \in V$;
- If $G \times H$ is \mathbb{P}^2 -generic, then in $V[G \times H]$, $\tau_G E \tau_H$.

We say that E is *pinned* if it doesn't admit a virtual class.

For instance, E_{set} admits a virtual class via the forcing $\mathbb{P} = \text{Coll}(\omega, 2^\omega)$ which adds an ω -sequence of reals with finite conditions. Any \mathbb{P} -generic sequence will list precisely the collection of ground model reals, and hence any two generics will be E_{set} equivalent. Similarly, it is easily seen that \cong_{WO} admits a virtual class via the forcing $\mathbb{Q} = \text{Coll}(\omega, \omega_1)$.

The key facts, essentially due to Hjorth, are that the countable Borel equivalence relations are pinned (see [15, Theorem 22]) and that there cannot be a Borel reduction from a nonpinned equivalence relation to a pinned equivalence relation (see [15, Lemma 20]). Using exactly the same methods, can show the following.

Proposition 5.5 *If E is an absolutely Δ_2^1 equivalence relation such that in any forcing extension, none of its classes can be changed by forcing, then E is pinned.*

Proof Suppose that E is not pinned, and let \mathbb{P} be a notion of forcing with a \mathbb{P} -name σ for a virtual class of E . Let g, h be mutually generic for \mathbb{P} , so that $\sigma_g E \sigma_h$ holds in $V[g, h]$. Since the classes of E , as interpreted in $V[g]$, cannot be changed by forcing, we must have $\sigma_h \in V[g]$. Since g, h are mutually generic, it follows that $V = V[g] \cap V[h]$, and hence $\sigma_h \in V$, a contradiction. \square

For instance, any countable infinite time eventually decidable equivalence relation satisfies the hypothesis of Proposition 5.5.

Proposition 5.6 *Let E and F be absolutely Δ_2^1 equivalence relations. If $E \leq_e F$ and F is pinned, then E is pinned.*

Proof Suppose that E is not pinned, and let \mathbb{P} be a notion of forcing with a \mathbb{P} -name σ for a virtual class of E . If f is an eventual reduction from E to F , it is easy to see that the natural \mathbb{P} -name for f applied to σ (let us call it $f(\sigma)$) has the property that if $G \times H$ is \mathbb{P}^2 -generic, then in $V[G \times H]$, we have $f(\sigma)_G F f(\sigma)_H$. Since F is pinned, \mathbb{P} forces that $f(\sigma) F y$ for some ground model real y . Now, by Shoenfield’s absoluteness theorem, there exists a ground model real x such that $f(x) F y$. It follows that \mathbb{P} forces that $f(\sigma) F f(x)$, and hence that σ is E -equivalent to the ground model real x , a contradiction. \square

Corollary 5.7 *E_{set} isn’t eventually reducible to E_∞ , or even to $\equiv_{e\infty}$.*

But recall that $\equiv_{e\infty}$ is infinite time eventually decidable; it is now unclear just where it should fit into the picture.

Question 5.8 *Is $\equiv_{e\infty}$ eventually reducible to the isomorphism relation \cong for countable structures?*

We next turn to cardinality arguments.

Proposition 5.9 *No infinite time computable equivalence relation which necessarily has 2^ω many classes is eventually reducible to \cong_{WO} .*

Proof Under $\neg\text{CH}$, this is clear since \cong_{WO} only has ω_1 many classes. So just force $\neg\text{CH}$ and appeal to Shoenfield’s absoluteness theorem. \square

Corollary 5.10 *Equality = on 2^ω is not eventually reducible to \cong_{WO} . Hence also $E_0, E_\infty, E_{\text{set}}, \cong_{HC}$, and so on, are not eventually reducible to \cong_{WO} .*

The following result shows that, moreover, \cong_{WO} does not reduce to E_∞ .

Proposition 5.11 *The equivalence relation \cong_{WO} does not computably reduce to any equivalence relation which is necessarily countable. In particular, \cong_{WO} does not eventually reduce to any infinite time enumerable equivalence relation.*

We should remark that similar arguments show that \cong_{WO} is not reducible to E_{set} (for instance, see [10, Theorem 9.4]).

Proof Let E be such a relation, and let f be an infinite time computable reduction from \cong_{WO} to E . Then $\text{im}(f)$ is Σ_2^1 , so we may let a be a parameter such that S is $\Sigma_2^1(a)$. By the Mansfield-Solovay theorem (see [13, Theorem 25.23]), if $\omega_1^{L[a]} < \omega_1 < 2^\omega$, then there is no $\Sigma_2^1(a)$ set of size \aleph_1 . Since $\text{im}(f)$ is clearly a $\Sigma_2^1(a)$ set of size \aleph_1 , we have reached a contradiction under these hypotheses.

Moreover, this situation can be forced over any model of ZFC. Since the proposition “the relation \cong_{WO} doesn’t computably reduce to E ” is Π_2^1 , Shoenfield’s absoluteness theorem implies that it holds. \square

Some of the relationships between the equivalence relations considered in this paper are summarized in Figure 3 (see the end of Section 3 for the definitions of these symbols). The dashed circles in the diagram indicate open questions—recall from Question 4.17 that the structure of the \equiv_a is not known. Similarly, it is not known how E_ζ , E_Σ , and $\equiv_{e\infty}$ compare with the other relations in the diagram.

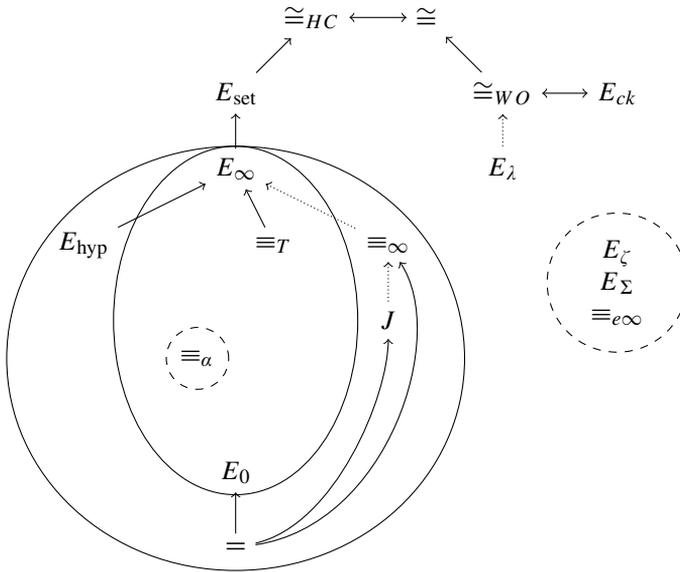


Figure 3 Solid arrows denote computable (or better) reductions; dotted arrows denote eventual reductions. The inner ellipse surrounds the nonsmooth countable Borel equivalence relations; the outer ellipse surrounds the infinite time eventually enumerable equivalence relations.

References

[1] Adams, S., and A. S. Kechris, “Linear algebraic groups and countable Borel equivalence relations,” *Journal of the American Mathematical Society*, vol. 13 (2000), pp. 909–43. [Zbl 0952.03057](#). [MR 1775739](#). [219](#)

[2] Feldman, J., and C. C. Moore, “Ergodic equivalence relations, cohomology, and von Neumann algebras. I,” *Transactions of the American Mathematical Society*, vol. 234 (1977), pp. 289–324. [Zbl 0369.22009](#). [MR 0578656](#). [215](#)

[3] Friedman, H., and L. Stanley, “A Borel reducibility theory for classes of countable structures,” *The Journal of Symbolic Logic*, vol. 54 (1989), pp. 894–914. [Zbl 0692.03022](#). [MR 1011177](#). [203](#)

- [4] Gao, S., *Invariant Descriptive Set Theory*, Pure and Applied Mathematics (Boca Raton). CRC Press, Boca Raton, 2009. [Zbl 1154.03025](#). [MR 2455198](#). [212](#)
- [5] Hamkins, J. D., R. Miller, D. Seabold, and S. Warner, “Infinite time computable model theory,” pp. 521–57 in *New Computational Paradigms: Changing Conceptions of What Is Computable*, edited by S. B. Cooper, B. Löwe, and A. Sorbi, Springer, 2007. [Zbl 1149.03024](#). [212](#)
- [6] Hamkins, J. D., “A simple maximality principle,” *The Journal of Symbolic Logic*, vol. 68 (2003), pp. 527–50. [Zbl 1056.03028](#). [MR 1976589](#). [213](#)
- [7] Hamkins, J. D., and A. Lewis, “Post’s problem for supertasks has both positive and negative solutions,” *Archive for Mathematical Logic*, vol. 41 (2002), pp. 507–23. [Zbl 1024.03043](#). [MR 1923194](#). [217](#)
- [8] Hamkins, J. D., and A. Lewis, “Infinite time Turing machines,” *The Journal of Symbolic Logic*, vol. 65 (2000), pp. 567–604. [Zbl 0963.03064](#). [MR 1771072](#). [205](#), [206](#), [207](#), [212](#), [216](#), [217](#)
- [9] Harrington, L. A., A. S. Kechris, and A. Louveau, “A Glimm-Effros dichotomy for Borel equivalence relations,” *Journal of the American Mathematical Society*, vol. 3 (1990), pp. 903–28. [Zbl 0778.28011](#). [MR 1057041](#). [211](#)
- [10] Hjorth, G., *Classification and Orbit Equivalence Relations*, vol. 75 of *Mathematical Surveys and Monographs*, American Mathematical Society, Providence, 2000. [Zbl 0942.03056](#). [MR 1725642](#). [223](#), [225](#)
- [11] Hjorth, G., and A. S. Kechris, “Borel equivalence relations and classifications of countable models,” *Annals of Pure and Applied Logic*, vol. 82 (1996), pp. 221–72. [Zbl 0933.03056](#). [MR 1423420](#). [203](#)
- [12] Jackson, S., A. S. Kechris, and A. Louveau, “Countable Borel equivalence relations,” *Journal of Mathematical Logic*, vol. 2 (2002), pp. 1–80. [Zbl 1008.03031](#). [MR 1900547](#). [214](#)
- [13] Jech, T., *Set Theory*, 3d millennium edition, Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2003. [Zbl 1007.03002](#). [MR 1940513](#). [225](#)
- [14] Kanamori, A., *The Higher Infinite. Large Cardinals in Set Theory from Their Beginnings*, 2d edition, Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2003. [Zbl 1022.03033](#). [MR 1994835](#). [209](#)
- [15] Kanoveĭ, V. G., and M. Reeken, “Some new results on the Borel irreducibility of equivalence relations,” *Rossiĭskaya Akademiya Nauk. Izvestiya. Seriya Matematicheskaya*, vol. 67 (2003), pp. 59–82. [Zbl 1068.03036](#). [MR 1957916](#). [224](#)
- [16] Kechris, A. S., *Classical Descriptive Set Theory*, vol. 156 of *Graduate Texts in Mathematics*, Springer-Verlag, New York, 1995. [Zbl 0819.04002](#). [MR 1321597](#). [212](#), [214](#)
- [17] Knight, J. F., S. Miller, and M. Vanden Boom, “Turing computable embeddings,” *The Journal of Symbolic Logic*, vol. 72 (2007), pp. 901–18. [Zbl 1123.03026](#). [MR 2354906](#). [204](#)
- [18] Stavi, J., and J. Väänänen, “Reflection principles for the continuum,” pp. 59–84 in *Logic and Algebra*, vol. 302 of *Contemporary Mathematics*, American Mathematical Society, Providence, 2002. [Zbl 1013.03059](#). [MR 1928384](#). [213](#)

- [19] Thomas, S., “Martin’s conjecture and strong ergodicity,” preprint, 2008. [222](#)
- [20] Welch, P., “Minimality arguments for infinite time Turing degrees,” pp. 425–36 in *Sets and Proofs (Leeds, 1997)*, edited by S. B. Cooper et al., vol. 258 of *London Mathematical Society Lecture Note Series*, Cambridge University Press, Cambridge, 1999. [Zbl 0937.03049](#). [MR 1720584](#). [219](#)
- [21] Welch, P., “Eventually infinite time Turing machine degrees: Infinite time decidable reals,” *The Journal of Symbolic Logic*, vol. 65 (2000), pp. 1193–203. [Zbl 0959.03025](#). [MR 1791371](#). [217](#)
- [22] Welch, P., “The length of infinite time Turing machine computations,” *The Bulletin of the London Mathematical Society*, vol. 32 (2000), pp. 129–36. [Zbl 1016.03040](#). [MR 1734198](#). [217](#)

The Graduate Center of The City University of New York
Mathematics Program
365 Fifth Avenue
New York NY 10016
USA
scoskey@nylogic.org
<http://math.rutgers.edu/~scoskey>

The Graduate Center of The City University of New York
Mathematics Program
365 Fifth Avenue
New York NY 10016
and
College of Staten Island of CUNY, Mathematics
2800 Victory Boulevard
Staten Island NY 10314
USA
jhamkins@gc.cuny.edu
<http://jdh.hamkins.org>