A CEER that is uniformly effectively inseparable but not uniformly finitely precomplete

Josiah Jacobsen-Grocott

Division of Mathematical Sciences
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
josiah.jg@ntu.edu.sg
https://josiahjg.github.io

Abstract. Computably enumerable equivalences relations (CEERs) are equivalence relations on $\mathbb{N}$ where the relation is given by a c.e. subset of $\mathbb{N}^2$. These have been studied by numerous people, and have some interesting properties and a degree structure. Of interest to us in this paper are uniformly effectively inseparable (u.e.i.) CEERs and uniformly finitely pre-complete (u.f.p.) CEERs. These are both types of universal ceers with particularly well behaved equivalence classes. It is known that every u.f.p. CEER is u.e.i, but it was not known if this implication reversed. The purpose of this paper is to show that it does not reverse by giving a construction of a ceer that is u.e.i. but not u.f.p.

Keywords: Computability, CEERs, Universal CEERs, Inseparable sets

1 Introduction

Computably enumerable equivalence relataions (CEERs) have been studied for there relationship to finitely presented groups [10], numberings [7] and as a natural class of structures in computability theory [6].

Definition 1. *a CEER is a c.e. equivalence relation. More formally,* $R \subseteq \mathbb{N} \times \mathbb{N}$ *is a CEER is if it is a c.e. set and an equivalence relation on* $\mathbb{N}$*.*

Some simple examples of CEERs include the identity function, id, where each equivalence class has only one element and the CEER consisting of only one equivalence class, $\mathbb{N}^2$, usually refereed to as id_1.

There is a natural reduction on CEERs that is induced by computable homomorphisms between CEERs.

Definition 2. *For CEERs, R and S we say that $R \leq S$ if there is a computable function f such that $xRy \iff f(x)Sf(y)$.*

This reduction is reminiscent of m-reducibility and indeed there is a relationship between the two. If A and B are c.e. sets then we can builds CEERs R_A and

R_B defined by $xR_Ay \iff x = y \vee x, y \in A$. If A and B are also non-computable then we have that $A \leq_1 B$ if and only if $R_A \leq R_B$.

Another relationship with m-reducibility is that if $R \leq S$ (reduction on CEERs) then $R \leq_m S$ (m-reduction on sets of pairs). However the converse does not hold. While the halting set, K, and R_K may be universal for the c.e. 1-degrees, R_K is not a universal CEER.

Definition 3. *A CEER, U, is universal if $R \leq U$ for all CEERs R.*

By taking the transitive, symmetric, reflexive closure of a c.e. subset of $\mathbb{N} \times \mathbb{N}$ we get a CEER. Hence it is possible to turn an effective listing of the c.e. sets, $(W_e)_e$, into an effective listing of all CEERs, $(R_e)_e$. Using this listing, we can give an example of a universal CEER: $U = \bigoplus_e R_e$ ($\langle e, x\rangle U \langle i, y\rangle \iff e = i \wedge xR_ey$).

A more interesting example of a universal CEER is logical equivalence of sentences with respect to the axioms of Peano arithmetic, $\equiv_{\mathrm{PA}}$. Via an appropriate encoding or sentences as numbers we can see that this is indeed a c.e. equivalence relation as two sentences are equivalent if and only if there is a formal proof of their equivalence from the axioms of PA.

As well as being a more natural example of a universal CEER, $\equiv_{\mathrm{PA}}$ has some interesting properties not shared with U, for instance, equivalence classes of $\equiv_{\mathrm{PA}}$ are inseparable sets.

Definition 4. *Two disjoint sets $A, B \subset \mathbb{N}$ are* inseparable *if there is no computable set C such that $A \subseteq C$ and $B \subseteq \overline{C}$.*

In fact, $\equiv_{\mathrm{PA}}$ satisfies a much stronger condition [5][2]:

Definition 5. *A CEER, R, is* uniformly effectively inseparable *(u.e.i.) if the is a total computable function $p : \mathbb{N}^4 \to \mathbb{N}$ such that for all $x, y, e, i \in \mathbb{N}$ if $\neg(xRy)$, $[x]_R \subseteq W_e$, $[y]_R \subseteq W_i$ and $W_e \cap W_i = \emptyset$ then $p(x, y, e, i) \notin W_e \cup W_i$.*

It has been shown [2] that every u.e.i. CEER is universal. The converse, however, does not hold, as there are universal CEERs, U for example, that contain some finite, and hence separable, equivalence classes.

Some other examples of u.e.i. CEERs can be found by restricting the domain of $\equiv_{\mathrm{PA}}$ to Σ_n sentences. Given an effective listing of all Σ_n sentences, $(\psi_e)_e$, we can define $\sim_n$ as $e \sim_n i$ if and only if $\mathrm{PA} \vdash \psi_e \leftrightarrow \psi_i$. These $\sim_n$ CEERs satisfy a stronger condition, that of being precomplete [9]:

Definition 6. *A CEER, R, is* precomplete *if there is a total computable function $f : \mathbb{N}^2 \to \mathbb{N}$ such that if $\varphi_e(i)\downarrow$, then $f(e, i)R\varphi_e(i)$. Such a function is called a* totalizer *for R.*

We can think of f as giving us a way to extend any partial computable function φ_e to a total computable one, $f(e, \cdot)$, that agrees with $\varphi_e(i)$ up to the R-equivalence. It may be that $f(e, i) \neq \varphi_e(i)$ but we will at least have $f(e, i)R\varphi_e(i)$.

In a sense, we have exhausted the examples of precomplete CEERs, as every precomplete CEER is computably isomorphic [8].

$\equiv_{\mathrm{PA}}$ is not precomplete, however it does satisfy a slightly weaker property, that of being u.f.p.

Definition 7. *[2] A CEER, R, is* uniformly finitely precomplete *(u.f.p.) if there is a total computable function f that takes as input a finite set and two numbers, such that if $\varphi_e(i) \downarrow= c$ and $c \in [D]_R$, then $f(D, e, i) R \varphi_e(i)$. Such a function is also referred to as a* totalizer *for R.*

We can think of u.f.p. as being precomplete for functions with a finite range. The existing work has given us the following series of implications [1]:

$$\text{precomplete} \implies \text{u.f.p.} \implies \text{u.e.i} \implies \text{universal} \tag{1}$$

The first and last implication are strict, as witnessed by $\sim_n$,$\equiv_{\mathrm{PA}}$ and U respectively. The purpose of this paper is show that the middle implication is strict, that is:

Theorem 1. *There is a CEER that is u.e.i. but not u.f.p.*

This problem has a bit of a history. In a survey paper on universal CEERs [1] that explores the properties of u.f.p. and u.e.i, Theorem 1 is stated as fact with a citation to a then forthcoming paper [3]. However, between the publication of the two papers errors were found in the proof and Theorem 1 remained an open question.

The author was introduced to this problem in a course on CEERs run by Uri Andrews, an author on both [1] and [3]. In this course, Andrews expressed his desire to see a proof Theorem 1, or its negation, and helped the author come up with the proof in this paper, pointing out errors in several earlier versions. The author would like to thank Uri Andrews for his generous help and encouragement.

2 The Proof

In this section we give the proof of Theorem 1, using a priority construction. We have broken this proof down in to subsections explaining the requirements, the strategies, the verification and the tool we use to keep track of when elements are not equivalent.

2.1 The requirements

We construct our CEER R and witness of uniform effective inseparability p using a finite injury argument with two types of requirements. The first will ensure effective inseparability:

$$\mathcal{S}_{x,y,e,i} : [x]_R \subseteq W_e \wedge [y]_R \subseteq W_i \wedge W_e \cap W_i = \emptyset \implies p(x, y, e, i) \notin W_e \cup W_i$$

These requirements are not actually part of the finite injury. They will be able to act without injuring or being injured and will ensure that our CEER is uniformly effectively inseparable.

The second type of requirement is:

$$\mathcal{P}_n : \varphi_n \text{ is not a totalizer for } R$$

These requirements can injure each other and have priority $\mathcal{P}_0 < \mathcal{P}_1 < \ldots$. At each stage s of the construction, we will first let the highest priority $\mathcal{P}_i$ requirement, with $i < s$, that wants to act follow its strategy. Then we let all $\mathcal{S}_{x,y,e,i}$ requirements act for $\langle x, y, e, i\rangle < s$.

2.2 Union—find

Before we explain the strategies for these requirements, we will go over how we represent the equivalence classes in this construction. The are ideas here take inspiration from the Union—Find data structure used in computer science [4].

This construction will be done in stages, building R and p as uniform increasing unions of finite relations. We will build R as the transitive symmetric closure of a directed graph E. We build E in stages alongside R.

As we build E, we will ensure that if xEy and xEz, then $y = z$, so each vertex maps to at most one other vertex. We will also ensure that E is a forest, that is it has no cycles. This means that, to see if two numbers, x, y are in the same equivalence class, we can follow the paths $xEx_1E \ldots Ex_n$ and $yEy_1E \ldots Ey_m$ until we get to dead ends x_n, y_m and xRy if and only if $x_n = y_m$. We call the dead end x_n the *representative* of $[x]_R$ and denote it as $\text{rep}(x)$.

To start with, $E_0 = \emptyset$, so each number will be the representative of its own equivalence class. If we then add an edge xEy then the two equivalence classes merge into a new one with representative $y = \text{rep}(x)$.

Note that the representative of each equivalence class is the only member that does not have an edge coming out of it. This means that if x and y are both representatives of their respective equivalence classes at some stage s and at some later stage t they have both merged into one equivalence class then at least one of x or y now has an edge coming out of it. This give us a way of proving two equivalence classes never collapse.

To help keep track of these equivalence classes we will assign numbers to strategies. When a number n is assigned to a strategy $\mathcal{R}$ we will say that $\mathcal{R}$ has *control* of n. While $\mathcal{R}$ has control of n no other requirement is allowed to add an edge from n. This means that if $\mathcal{R}$ has control of n and m, then it can ensure that $\neg(nRm)$. However if $\mathcal{R}$ only has control of n and not m then it whatever requirement that has control of m can add the edge mEn, so $\mathcal{R}$ cannot keep the two numbers in separate equivalence classes. However, if some other requirement adds the edge mEn, then that makes $n = \text{rep}(m)$, so from that point on $\mathcal{R}$ effectively has control over m.

2.3 The strategies

We first explain the strategy for $\mathcal{S}_{x,y,e,i}$. It starts by picking a witness $w = p(x, y, e, i)$ that has not been used in the construction so far. First, if at any stage s in the future we see xR_sy then this requirement is considered satisfied. We no longer need the witness w so we can let a $\mathcal{P}$ requirement claim it as a witness for its strategy and cede control of w to that requirement. Second, if at any stage t we have not given away our witness w and see $w \in W_{e,t} \cup W_{i,t}$, then if $w \in W_e$ we set $wE_{t+1}y$ and if $w \in W_i$ then set $wE_{t+1}x$.

Since these requirements have complete control over what their witness can point to unit they finish acting, and because equivalence classes merging does not cause them problems, we can see that they will never be injured.

Now we describe the strategy for $\mathcal{P}_n$. When this requirement is initialized we pick two witnesses m, k and a code e such that $\varphi_e(e)$ is the program that runs this construction and outputs a value of this strategy's choosing at some possible future stage. We now wait until a future stage s where we see $\varphi_n(\{m,k\}, e, e) \downarrow = c$ for some c. If we get to such a stage, then we injure all lower priority $\mathcal{P}$ requirements decided the value of $\varphi_e(e)$ as follows.

We now consider a new directed graph. The vertices are the representatives for R_s-equivalences classes, and for each active $\mathcal{S}$ requirements there is an edge from its witness w to $\mathrm{rep}_s(x)$ and an edge to from w to $\mathrm{rep}_s(y)$. We consider the set of vertices reachable from $\mathrm{rep}_s(c)$. If this set omits m then we set $\varphi_e(e) = m$ and consider the requirement satisfied unless injured later.

If this set contains m then we consider a path $\mathrm{rep}_s(c) = w_0, w_1, \ldots, m$. Since the only vertices with edges out of them are witnesses for $\mathcal{S}$ requirements, we know that $w_0, \ldots, w_{n-1}$ are all witnesses for $\mathcal{S}$ requirements. Let $x_0, \ldots, x_{n-1}$ being the other vertices such that there is an edge from w_j to x_j (note that the x_j's may not be distinct from each other, or the w_j, but this is not a problem). We now perform a series of collapses adding E-edges $mE_{s+1}x_{n-1}, \ldots, w_1E_{s+1}x_0$. The first collapse allows us to gain control of the witness w_{n-1} from its § strategy, which allows us to perform the next collapse, and so on. We have now lost control of $\mathrm{rep}_s(m)$, but have gained control of $w_0 = \mathrm{rep}_s(c)$ as its requirement no longer needs it. We set $\varphi_e(e) = k$ as $w_0 \neq k$ and we have control of both these numbers.

2.4 Verification

Now we verify in detail that the construction works and that the resulting CEER R is indeed u.e.i. but not u.f.p.

Claim. At each stage s, E_s is a forest of finite trees where each vertex has outdegree ≤ 1, the function $x \mapsto \mathrm{rep}_s(x)$ is a well defined function and each active witness w for a strategy has $w = \mathrm{rep}_s(w)$

Proof. We prove this using induction. The base case holds as $E_0 = \emptyset$ and $\mathrm{rep}_0(x) = x$. Suppose that rep_s is well defined and E_s is a forest of finite trees where each vertex has outdegree ≤ 1. At stage s we only let at most one $\mathcal{P}$ requirement and $s-1$ many $\mathcal{S}$ requirements act. When each requirements acts it only adds finitely many edges, so E_{s+1} is finite. Before adding an edge, we require a strategy to make sure it is not forming a cycle, so we ensure that E_{s+1} is a tree.

The only time we add an edge, it is an edge from an active witness for a strategy that becomes inactive at stage $s+1$, so we maintain that the outdegree of each vertex is at most 1 in E_{s+1} and that each active witness at stage $s+1$ has outdegree 0. If we look at a connected component of E_{s+1} it is a finite tree and each vertex has outdegree at most 1, so there is a unique vertex of outdegree 0 that will become the Representative of all vertices in that component. So rep_{s+1} is well defined.

Claim. If xRy then there is a stage s such that $\mathrm{rep}_s(x) = \mathrm{rep}_s(y)$.

Proof. If xRy then there is a stage s such that xR_sy. This means that x and y are in the same connected component of E_s. So it must be that $\mathrm{rep}_s(x) = \mathrm{rep}_s(y)$.

Claim. All $\mathcal{S}_{x,y,e,i}$ requirements are satisfied.

Proof. Consider a requirement $\mathcal{S} = \mathcal{S}_{x,y,e,i}$. There a three cases to consider:

1. If xRy then this requirement is satisfied.
2. If $\neg(xRy)$ and the strategy for $\mathcal{S}$ never added any edges, then, since $\neg(xRy)$, this can only be because $w = p(x, y, e, i) \notin W_e \cup W_i$. So W_e, W_i do not separate $[x]_R$ and $[y]_R$.
3. If the strategy for $\mathcal{S}$ added an edge, then it did so because it saw $w = p(x, y, e, i) \in W_e \cup W_i$. In this case it ensured that $w \in [x]_R \cap W_i$ or $w \in [y]_R \cap W_e$, so W_e, W_i do not separate $[x]_R$ and $[y]_R$.

So, as long as there are two distinct equivalence classes, R will be u.e.i.

Claim. Each $\mathcal{P}_n$ requirement is satisfied, and R is not u.f.p.

Proof. We will use a finite injury argument with induction. For the requirement $\mathcal{P}_n$ let s_0 be a stage after witch all higher priority requirements have stopped acting. So $\mathcal{P}_n$ will chose fresh witnesses m and k, and program e, and observe the computation of $\varphi_n(\{m, k\}, e, e)$. If $\varphi_n(\{m, k\}, e, e)\uparrow$ then $\mathcal{P}$ is satisfied as φ_n cannot be a totalizer. Suppose that at stage $s \geq s_0$ we see $\varphi_n(\{m, k\}, e, e)\downarrow= c$ then there are two cases we must consider.

1. suppose that we performed at series of collapses to gain control over $w = \mathrm{rep}_s(c)$. Then $\varphi_e(e) = k$. Since m was reachable from w, it must be that w was a witness for some $\mathcal{S}$ requirement or $w = m$. In either case, $w \neq k$, so $\mathrm{rep}_s(c) \neq \mathrm{rep}_s(k)$. This requirement is now finished acting. Since no higher priority requirement acts after stage s_0, w and k remain witnesses for this requirement, and we have, for all t, $\mathrm{rep}_s(c) \neq \mathrm{rep}_s(k)$. So by claim 2.4, $\neg(cRk)$, and thus φ_n is not a totalizer.
2. For the second case, suppose that m was not reachable from $\mathrm{rep}_s(c)$. Then $\varphi_e(e) = m$. We will use induction to show that for all $t \geq s$, we have that neither m, nor any witness for a lower priority requirement, is reachable from $\mathrm{rep}_t(c)$ in the graph of representatives. This will ensure that $\forall t(\mathrm{rep}_t(c) \neq m)$, and so $\neg(cRm)$.

The base case holds, since we are assuming m was not reachable at stage s and all lower priority requirements were just injured. Suppose that at stage t, we have that none of the vertices reachable from $\mathrm{rep}_t(c)$ are m or a witness for a lower priority requirement. Now we consider what effects the actions of strategies will have on this graph.

(a) If any requirement is initialized at stage t, then it picks for its witness a large number, not seen in the construction so far, and thus not one of finite set of representatives that are reachable from $\mathrm{rep}_t(c)$. This may add edges to the graph, but it will not change the set of vertices reachable from $\mathrm{rep}_t(c)$.

(b) If some $\mathcal{S}_{x,y,e,i}$ requirement acts, then, without loss of generality we see w collapse with the class x. In our graph of representatives this corresponds to a contraction along the edge from w to $\mathrm{rep}_t(x)$ and a deletion of the edge from w to $\mathrm{rep}_t(y)$. Contraction and deletion will not make more vertices reachable from $\mathrm{rep}_{t+1}(c)$, so this cannot break the induction hypothesis.

(c) If a $\mathcal{P}$ requirements acts at stage t adding an edge to E_{t+1}, then it will may gain control of some witness w. Since higher priority requirements will not act after stage s, $\mathcal{P}$ must be a lower priority requirement. Thus, one of its witnesses was reachable from w at stage t by inductive hypothesis, and so w was not reachable from $\mathrm{rep}_t(c)$ at stage t. So the collapse $\mathcal{P}$ performed to gain control of w and any collapses $\mathcal{P}$ performs using w will not effect the graph of vertices reachable from $\mathrm{rep}_{t+1}(c)$ at stage $t+1$.

Putting all cases together, we can see that the set of representatives reachable from $\mathrm{rep}_{t+1}(c)$ at stage $t+1$ is a subset of those reachable form $\mathrm{rep}_t(c)$ at stage t. So the induction holds.

So in either case we have that φ_n is not a totalizer.

Since every $\mathcal{P}_n$ is satisfied, there is no totalizer and R is not u.f.p. or id_1. Hence R is u.i.e but not u.f.p.

References

1. Andrews, U., Badaev, S., Sorbi, A.: A survey on universal computably enumerable equivalence relations. In: Computability and complexity, Lecture Notes in Comput. Sci., vol. 10010, pp. 418–451. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-50062-1_25, https://doi-org.remotexs.ntu.edu.sg/10.1007/978-3-319-50062-1_25
2. Andrews, U., Lempp, S., Miller, J.S., Ng, K.M., San Mauro, L., Sorbi, A.: Universal computably enumerable equivalence relations. J. Symb. Log. **79**(1), 60–88 (2014). https://doi.org/10.1017/jsl.2013.8, https://doi-org.remotexs.ntu.edu.sg/10.1017/jsl.2013.8
3. Andrews, U., Sorbi, A.: Jumps of computably enumerable equivalence relations. Ann. Pure Appl. Logic **169**(3), 243–259 (2018). https://doi.org/10.1016/j.apal.2017.12.001, https://doi-org.remotexs.ntu.edu.sg/10.1016/j.apal.2017.12.001

4. Arden, B.W., Galler, B.A., Graham, R.M.: An algorithm for equivalence declarations. Communications of the ACM **4**(7), 310–314 (1961)
5. Bernardi, C.: On the relation provable equivalence and on partitions in effectively inseparable sets. Studia Logica **40**(1), 29–37 (1981). https://doi.org/10.1007/BF01837553, https://doi-org.remotexs.ntu.edu.sg/10.1007/BF01837553
6. Eršov, J.L.: Positive equivalences. Algebra i Logika **10**, 620–650 (1971)
7. Eršov, J.L.: Teoriya numeratsiĭ. Matematicheskaya Logika i Osnovaniya Matematiki. [Monographs in Mathematical Logic and Foundations of Mathematics], "Nauka", Moscow (1977)
8. Lachlan, A.H.: A note on positive equivalence relations. Z. Math. Logik Grundlag. Math. **33**(1), 43–46 (1987). https://doi.org/10.1002/malq.19870330106, https://doi-org.remotexs.ntu.edu.sg/10.1002/malq.19870330106
9. Mal'tsev, A.I.: Totally enumerated sets. Algebra i Logika Sem. **2**(2), 4–29 (1963)
10. Miller, III, C.F.: On group-theoretic decision problems and their classification, Annals of Mathematics Studies, vol. No. 68. Princeton University Press, Princeton, NJ; University of Tokyo Press, Tokyo (1971)