Computability 0 (0) 1
IOS Press

Primitive recursive equivalence relations and their primitive recursive complexity

Nikolay Bazhenov
Sobolev Institute of Mathematics, 4 Acad. Koptyug Ave., Novosibirsk, 630090, Russia
bazhenov@math.nsc.ru

Keng Meng Ng
School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
kmng@ntu.edu.sg

Luca San Mauro
Department of Mathematics "Guido Castelnuovo", Sapienza University of Rome, I-00185 Rome, Italy
luca.sanmauro@gmail.com

Andrea Sorbi
Department of Information Engineering and Mathematics, University of Siena, I-53100 Siena, Italy
andrea.sorbi@unisi.it

Abstract. The complexity of equivalence relations has received much attention in the recent literature. The main tool for such endeavour is the following reducibility: given equivalence relations R and S on natural numbers, R is computably reducible to S if there is a computable function $f\colon \omega \to \omega$ that induces an injective map from R-equivalence classes to S-equivalence classes. In order to compare the complexity of equivalence relations which are computable, researchers considered also feasible variants of computable reducibility, such as the polynomial-time reducibility. In this work, we explore **Peq**, the degree structure generated by primitive recursive reducibility on primitive recursive equivalence relations with infinitely many equivalence classes. In contrast with all other known degree structures on equivalence relations, we show that **Peq** has much more structure: e.g., we show that it is a dense distributive lattice. On the other hand, we also offer evidence of the intricacy of **Peq**, proving, e.g., that the structure is neither rigid nor homogeneous.

Keywords: Primitive recursive function, primitive recursive equivalence relation, lattice

1. Introduction

The classification of equivalence relations according to their complexity is a major research thread in logic. The following two examples stand out from the existing literature.

- In descriptive set theory, one often deals with equivalence relations defined on Polish spaces (e.g., 2^ω or ω^ω), which are classified in terms of *Borel embeddings*. The corresponding theory is now a consolidated field of modern descriptive set theory, which shows deep connections with topology, group theory, combinatorics, model theory, and ergodic theory (see, e.g., [1–4]).
- On the other hand, in computability theory, it is common to concentrate on equivalence relations on the natural numbers and compare their algorithmic content in terms of *computable reductions* (see, e.g., [5–9]).

Computable reducibility (for which we use the symbol $\leqslant_c$, and call *c-degrees* the elements of the corresponding degree structure) has been adopted to calculate the complexity of natural equivalence relations on ω, proving, e.g., that provable equivalence in Peano Arithmetic is Σ^0_1 complete [10], Turing equivalence on c.e. sets is Σ^0_4 complete [11], and the isomorphism relations on several familiar classes of computable structures (e.g., trees, torsion abelian groups, fields of characteristic 0 or p) are Σ^1_1 complete [12]. In parallel, there has been a growing interest in the abstract study of the poset of degrees generated by computable reducibility on the collection of equivalence relations of a certain complexity. Most notably, the poset **Ceers** of the c-degrees of computably enumerable equivalence relations (commonly known by the acronym *ceers*) has been thoroughly explored [13, 14]: e.g., it has been recently shown that its first-order theory is as complicated as true arithmetic [15]. Less is known about larger structures of c-degrees; but recent studies considered the Δ^0_2 case [16, 17] and the global structure **ER** of *all* c-degrees [18].

Yet, despite its classificatory power, computable reducibility has an obvious shortcoming: it is simply too coarse for measuring the relative complexity of computable equivalence relations. Indeed, it is easy to note that any two computable equivalence relations R and S with infinitely many classes are computably bi-reducible. A natural way to overcome this limitation is by adopting *feasible* reducibilities, as is done in [19, 20], where the authors prove that, relatively to these reducibilities, the isomorphism relations of finite fields, finite abelian groups, and finite linear orders have all the same complexity.

This paper focuses on a subcollection of computable equivalence relations, namely primitive recursive equivalence relations with domain ω. To classify primitive recursive equivalence relations, we adopt primitive recursive reducibility. More precisely, a primitive recursive equivalence relation R is *pr-reducible* to a primitive recursive equivalence relation S (notation: $R \leqslant_{pr} S$) if there exists a primitive recursive function f such that

$$(\forall x, y \in \omega)[x \; R \; y \Leftrightarrow f(x) \; S \; f(y)].$$

The main object of study of this paper is **Peq**, i.e., the degree structure of *pr*-degrees of primitive recursive equivalence relations R possessing infinitely many equivalence classes.

A final piece of motivation comes from the rapid emergence of *online structure theory* (see, e.g., [21–26]), a subfield of computable structure theory which deals with algorithmic situations in which unbounded search is not allowed, that is formalized by focusing, e.g., on *punctual* presentations of structures, rather than computable ones. An infinite structure $\mathcal{S}$ (in a finite signature L) is *punctual* if the domain of $\mathcal{S}$ equals ω, and the signature functions and relations of $\mathcal{S}$ are primitive recursive. Note that the study of primitive recursive structures dates back to the works of Mal'tsev [27, 28]: in Subsection 3.1 of [27], he introduces primitive recursive algebraic systems.

The present paper delivers three main theorems, by which we hope to convince the reader that **Peq** is a remarkable structure. First, we prove that **Peq** is surprisingly well-behaved. Many of the results obtained in Sections 3–7 are collected in the following main theorem:

Theorem 1.1. **Peq** *is a a topped dense distributive lattice, in which each degree is join-reducible and each degree below the top is meet-reducible.*

This is in sharp contrast with **Ceers** and all other established structures of c-degrees. In fact, these desirable properties led us to formulate the hypothesis that **Peq** could be a simple structure which might have a decidable first order theory. We were unable to prove this (it remains open whether the theory of **Peq** is decidable). But, as sometimes happens, aiming to prove that the structure is simple, we unveiled its hidden complexity. Most notably, we individuated a new technical notion (to be defined in Section 8). This notion is called *♦-property*, and it captures many combinatorial features of **Peq**. Remarkably, the ♦-property allows to distinguish intervals of **Peq**. So, the following result is our second main theorem:

Theorem 1.2. *An interval of* **Peq** *embeds the diamond lattice preserving* 0 *and* 1 *if and only if it satisfies the ♦-property.*

In the final section, we uncover even further structural complexity, confirming that the tameness of **Peq** was only apparent. These results are subsumed in our third main theorem:

Theorem 1.3. **Peq** *is neither rigid nor homogeneous and it contains nonisomorphic lowercones.*

The reader should be warned that working in a primitive recursive setting will affect our proof strategies: as any primitive recursive check converges, our constructions will be typically injury-free and requirements will be satisfied one by one. That said, to build primitive recursive objects, we won't need to worry about coding, and in fact we will rely on a restricted form of Church–Turing thesis (see Remark 2.3). Moreover, let us warn the reader that the difficulty of our proofs increases over the course of the paper. Indeed, the techniques employed in the first half of the paper, which is devoted to Theorem 1.1, are fairly straightforward. But in order to tackle Theorems 1.2 and 1.3, we had to develop new techniques which make the discussion in the second half of the paper more combinatorially involved.

2. Background, terminology, and notations

We review some background material and terminology. For background on computability theory, especially primitive recursive functions, the reader is referred to [29].

2.1. Equivalence relations

Unless stated otherwise, all our equivalence relations will be primitive recursive relations on domain ω. A *transversal* of an equivalence relation R is any set T such that $x \not R y$ for any distinct $x, y \in T$. The *principal transversal* T_R of a given equivalence relation R is the following transversal,

$$T_R := \{y \neq 0 : (\forall x < y)(x \not R y)\}.$$

Clearly, if R is primitive recursive, then T_R is a primitive recursive set. We often write $f \colon R \leqslant_{pr} S$ to mean that f is a primitive recursive reduction from R to S. As is customary in the theory of ceers, we denote by Id the identity on the natural numbers.

For our purposes, it is convenient to assume that all our equivalence relations have infinitely many equivalence classes. This is a minor restriction. Indeed, similarly to the case of ceers [7] (but differently from the case of Δ^0_2 equivalence relations [17]), primitive recursive equivalence relations with finitely many equivalence classes can be readily characterized. To do so, let Id_n denote the identity mod n and simply observe the following easy facts:

(1) $\mathrm{Id}_1 <_{pr} \mathrm{Id}_2 <_{pr} \cdots <_{pr} \mathrm{Id}_n <_{pr} \cdots$;
(2) every primitive recursive R with finitely many equivalence classes is pr-equivalent to Id_n, for some $n \geqslant 1$;
(3) for all $n \geqslant 1$, Id_n is pr-reducible to any primitive recursive R with infinitely many equivalence classes.

Definition 2.1. We say that an equivalence relation E is *infinitary* if E has infinitely many classes.

Remark 2.2. Henceforth, we will work with infinitary primitive recursive equivalence relations. We let **Peq** be the poset of pr-degrees of these relations under pr-reducibility.

2.2. A listing of the primitive recursive functions

Throughout the paper we will refer to an effective listing $\{p_e\}_{e\in\omega}$ of the primitive recursive functions, which can be found in many textbooks: see, e.g., [30] for a detailed definition of such a listing. Let $T(e, x, z)$ and U denote respectively Kleene's (primitive recursive) predicate and a primitive recursive function such that for every e, $\varphi_e(x) = U(\mu z\, T(e, x, z))$ (where φ_e denotes the partial computable function with index e in the standard listing of the partial computable functions), and let p be a recursive function such that $p_e = \varphi_{p(e)}$: since p comes from the *s-m-n*-theorem we may assume that p is primitive recursive. Let V be the primitive recursive predicate

$$V(e, x, y, s) \ \Leftrightarrow\ (\exists z < s)(T(p(e), x, z) \,\&\, U(z) = y) :$$

we will refer to $V(e, x, y, s)$ by saying that "$p_e(x)$ has converged to y in $< s$ steps", and we will denote this by $p_e(x)[s]\!\downarrow = y$. Similarly, it is primitive recursive to check whether "$p_e(x)$ has converged in $< s$ steps" (denoted by $p_e(x)[s]\!\downarrow$), i.e., $(\exists y < s)V(e, x, y, s)$.

2.3. Binary strings

We will use the standard notations and terminology about *finite binary strings*, which are the elements of the set $2^{<\omega}$. Let σ, τ be finite binary strings: we will denote by l^σ the length of σ; the concatenations of σ and τ will be denoted by $\sigma^\frown\tau$; if $i \in \{0, 1\}$ is a number then $\langle i \rangle$ denotes the string of length 1, consisting of the single bit i; the symbol λ denotes the empty string.

We will freely identify a set $X \subseteq \omega$ with its characteristic function, thus viewing X as a member of 2^ω. If $k \in \omega$ and X is a set, then the symbol $X \upharpoonright k$ denotes the initial segment of X (thus a member of $2^{<\omega}$) of length k. Given $\sigma \in 2^{<\omega}$ and a set X, let

$$\sigma * X = \begin{cases} \sigma(i), & \text{if } i < l^\sigma, \\ X(i - l^\sigma), & \text{otherwise.} \end{cases}$$

In analogy with the common usage for sets (where $\overline{X} = \{x : X(x) = 0\}$), given a string $\sigma \in 2^{<\omega}$, we denote $\overline{\sigma} = \{i : i < l^\sigma \ \& \ \sigma(i) = 0\}$, and we let $\#\,(0^\sigma) = \#(\overline{\sigma})$, i.e., the number of 0's in the range of σ.

We will assume a "primitive recursive coding" of the binary strings, with primitive recursive length function, projections, etc.

Remark 2.3. Throughout this paper we will often build primitive recursive functions or primitive recursive sets. It is sometimes convenient to use the following analogue of the Church–Turing thesis for primitive recursive functions:

Let φ_e be the e^{th} partial (general) computable function. Then (in accordance with what stated in Section 2.2) every primitive recursive function is equal to some φ_e where the computation for φ_e runs in primitively recursively many steps. Thus, to define a primitive recursive function f (or a set), it is enough to specify a general algorithm for computing $f(n)$ as long as the number of steps taken to decide $f(n)$ is bounded by a primitive recursive function.

This helps to avoid overly formal and cumbersome definitions, since it is often easy to see that the time bound is primitive recursive.

3. Normal form of primitive recursive equivalence relations

Recall that we prefer to work only with infinitary equivalence relations. Nevertheless, we note that the results of this section also hold for primitive recursive equivalence relations having only finitely many classes.

In the literature about computable reducibility, it is common to use the following way of encoding sets of numbers by equivalence relations: given $X \subseteq \omega$, one defines

$$xR_Xy \ \Leftrightarrow \ x, y \in X \text{ or } x = y.$$

Equivalence relations of the form R_X are called *1-dimensional* in [7], while c-degrees containing 1-dimensional equivalence relations are called *set-induced* in [16]. The interesting feature of set-induced degrees is that they offer algebraic and logical information about the overall structure of c-degrees: for example, in [8] it is proved that the first-order theory of ceers is undecidable, by showing that the interval $[\deg_c(\mathrm{Id}), \deg_c(R_K)]$ of the c-degrees is isomorphic to the interval $[\mathbf{0}_1, \mathbf{0}_1']$ of the 1-degrees, where $\mathbf{0}_1$ is the 1-degree of an infinite and co-infinite computable set and $\mathbf{0}_1'$ is the 1-degree of the halting problem K. Yet, set-induced degrees are far from exhausting the collection of all c-degrees. In fact, if R is an equivalence relation with two non-computable R-classes, then there is obviously no X such that $R \equiv_c R_X$.

When dealing with primitive recursive equivalence relations, the situation changes: all pr-degrees are set-induced. Specifically, the next theorem says that investigating primitive recursive equivalence relations under $\leqslant_{pr}$ is the same as investigating primitive recursive sets under a primitive recursive reducibility that is required to be bijective on the complements of the sets.

Theorem 3.1. *Let R and S be primitive recursive equivalence relations. Then there are primitive recursive sets X and Y such that the following statements are equivalent:*

(1) *$R \leqslant_{pr} S$;*
(2) *there is a primitive recursive function $g\colon X \leqslant Y$ so that*

 (a) *$x \in X$ if and only if $g(x) \in Y$, and*
 (b) *$g[\overline{X}] = \overline{Y}$.*

Proof. The proof of the theorem is conveniently separated into intermediate results. The next one, which is particularly useful when dealing with **Peq**, states that the *pr*-complexity of any R is entirely encoded in its principal transversal T_R.

Proposition 3.2 (Normal Form)**.** *For any primitive recursive equivalence R, we have*

$$R \equiv_{pr} R_{\overline{T_R}}.$$

Proof. It is immediate to see that the function $f(x) = (\mu y \leqslant x)[y\ R\ x]$ is primitive recursive and reduces R to $R_{\overline{T_R}}$. On the other hand, the following primitive recursive function g reduces $R_{\overline{T_R}}$ to R:

$$g(x) = \begin{cases} 0, & \text{if } x \in \overline{T_R}, \\ x, & \text{otherwise.} \end{cases}$$

Hence, $R \equiv_{pr} R_{\overline{T_R}}$. □

Since the lemma guarantees that $R \leqslant_{pr} S$ holds if and only if $R_{\overline{T_R}} \leqslant_{pr} R_{\overline{T_S}}$, we now wish to define the desired primitive recursive sets X and Y as $\overline{T_R}$ and $\overline{T_S}$, respectively. To conclude, we shall prove that item (2) above is satisfied. That is, we need to prove the following: if $R_{\overline{T_R}} \leqslant_{pr} R_{\overline{T_S}}$ is true, then there is a *pr*-reduction g witnessing this fact such that g maps $\overline{T_R}$ into $\overline{T_S}$ and $g[T_R] = g[T_S]$. The next couple of lemmas show that such a g always exists.

Lemma 3.3. *If $R_X \leqslant_{pr} R_Y$, then there is a reduction g from R_X to R_Y such that $g[X] \subseteq Y$.*

Proof. Let f be a primitive recursive function reducing R_X to R_Y such that $f[X] = \{a\}$ with $a \notin Y$. Fix $y \in Y$ and define

$$g(x) = \begin{cases} y, & \text{if } x \in X, \\ a, & \text{if } x \notin X \text{ and } f(x) \in Y, \\ f(x), & \text{if } x \notin X \text{ and } f(x) \notin Y. \end{cases}$$

It is easy to see that g is primitive recursive, maps X to Y, and reduces R_X to R_Y. □

Lemma 3.4. *If $R_X \leqslant_{pr} R_Y$ then there exists g such that $g\colon R_X \leqslant_{pr} R_Y$ and g hits all the R_Y-classes.*

Proof. Let $R_X \leqslant_{pr} R_Y$ via f. By Lemma 3.3, we may assume that f maps X to Y. Define

$$g(x) = \begin{cases} f(x), & \text{if } x \in X, \\ \mu y\,[y \leqslant \max\{f(i) : i \leqslant x\} \mathbin{\&} y \in \overline{Y} \smallsetminus \{g(i) : i < x\}], & \text{if } x \notin X. \end{cases}$$

Then g gives the reduction and is onto $\overline{Y}$: to show primitive recursiveness of g we use the fact that if x is the n-th element in $\overline{X}$, then the n-th element of $\overline{Y}$ in order of magnitude is $\leqslant \max\{f(i) : i \leqslant x\}$ by injectivity of f on $\overline{X}$. □

This concludes the proof of Theorem 3.1. □

It is worth highlighting a couple of features of the last technical lemma.

Remark 3.5. The function $g\colon R_X \leqslant_{pr} R_Y$ constructed in Lemma 3.4 is increasing on $\overline{X}$, i.e., if $x < y$ and $x, y \in \overline{X}$ then $g(x) < g(y)$. Moreover, the lemma allows to assume that, without loss of generality, any *pr*-reduction between primitive recursive equivalence relations is surjective on the equivalence classes of its target. This contrasts with the case of ceers, where (see [13]) if R, S are such that $R \leqslant_c S$ via a computable function f whose range hits all the equivalence classes of S, then the reduction can be inverted, i.e., we have $S \leqslant_c R$ as well.

In the rest of the paper, we will take advantage of the correspondence between reductions of primitive recursive equivalences and reductions of primitive recursive sets just discussed. We often construct, instead of a full equivalence relation R, only a primitive recursive set Y, and then we take R to be R_Y.

4. Incomparability of degrees in Peq

In this and the following section, we tackle some of the most natural questions that one can formulate about a new degree structure.

4.1. The greatest degree in Peq

We begin by proving that there is a greatest degree in **Peq**.

Proposition 4.1. **Peq** *has a greatest element. In fact, for all primitive recursive R, $R \leqslant_{pr} \mathrm{Id}$.*

Proof. Given R, let f be the primitive recursive function from the proof of the Normal Form Theorem (i.e., the function reducing R to $R_{\overline{T_R}}$). Notice that f is also a reduction from R to Id, as it maps equivalence classes to singletons. □

The *pr*-degree of Id contains many natural examples of equivalence relations. For example, in the literature about polynomial-time reducibility (see, e.g., [19]) researchers considered the isomorphism relations of familiar classes of *finite* structures, such as graphs, groups, trees, linear orders, Boolean algebras, and so forth. It is not difficult to see that all these relations turn out to be *pr*-equivalent to Id. Consider for instance GI, the isomorphism relation between finite graphs. On one hand, the problem of deciding whether two finite graphs are isomorphic is primitive recursive (in fact, it belongs to **NP**). On the other hand, a *pr*-reduction from Id to GI is readily obtained by assigning, to each n the empty graph on the domain $\{i : i < n\}$. Hence, GI is *pr*-equivalent to Id.

Anyway, the fact that the *pr*-degree of Id contains many natural representatives does not come as a surprise. In fact, construction of a computable function which is not primitive recursive requires non-trivial work (recall Ackermann's famous construction [31]). Next, we show that a primitive recursive equivalence relation R lies strictly below Id only if, when presented in normal form, the set of its singletons cannot be enumerated in a primitive recursive way without repetitions.

To be more precise, let us introduce first the following analogue of immunity for primitive recursive sets.

Definition 4.2. A set $X \subseteq \omega$ is *primitively recursively enumerable* (abbreviated by *p.r.e.*) if X is the range of an injective primitive recursive function. X is *primitively recursively immune* (abbreviated by *p.r.-immune*) if X is infinite and it has no infinite p.r.e. subset.

Remark 4.3. Note that we define a set as p.r.e. only if it has a primitive recursive enumeration which is *injective*. Without injectivity one would obtain all c.e. sets: it follows easily from Kleene's Normal Form Theorem that any nonempty c.e. set can be enumerated by a primitive recursive function. In contrast, Theorem 4.4 shows that the p.r.e. sets (as just defined) form a proper subclass of the c.e. sets, and in fact even of the primitive recursive sets.

The notion of *p.r.*-immunity allows to show that there exists a *pr*-degree strictly below the identity.

Theorem 4.4. *There exists an infinitary primitive recursive R such that $R <_{pr} \mathrm{Id}$.*

Proof. We first prove that, if Id reduces to some R_X, then the set X cannot be *p.r.*-immune.

Lemma 4.5. *If X is any set of numbers, then $\mathrm{Id} \nleqslant_{pr} R_X$ if and only if $\overline{X}$ is p.r.-immune.*

Proof. $(\Rightarrow)$: If $\overline{X}$ contains a p.r.e. set A, then $\mathrm{Id} \leqslant_{pr} R_X$ via any primitive recursive function that injectively lists A.

$(\Leftarrow)$: If $\mathrm{Id} \leqslant_{pr} R_X$ via some primitive recursive function f, then, with the exception of at most one element, the range of f is contained in $\overline{X}$. Therefore, $\mathrm{range}(f) \cap \overline{X}$ is an infinite p.r.e. set showing that $\overline{X}$ is not p.r.-immune. □

So, combining the last lemma with Proposition 4.1, we obtain that, in order to construct a primitive recursive equivalence relation which is strictly *pr*-reducible to Id, it suffices to build a primitive recursive set whose complement is p.r.-immune. This is done by the next proposition.

Proposition 4.6. *There exists a primitive recursive set Y whose complement is p.r.-immune.*

Proof. We construct Y in stages by approximating its characteristic function, i.e., $Y = \bigcup_{s\in\omega} \sigma_s$, where $l^{\sigma_s} = s$. The construction of Y is similar to that of a simple set and is rather straightforward. We describe it in some detail to let the readers familiarize themselves with the sort of machinery that we employ in more intricate constructions.

We aim at satisfying the following requirements:

P_e : if p_e is injective, then $\mathrm{range}(p_e) \cap Y \neq \emptyset$,

M : Y is co-infinite,

where $\{p_e\}_{e\in\omega}$ is a computable list of all primitive recursive functions, as in Section 2.2.

The strategy for a P_e-requirement works as follows. During the so-called "P_e-cycle" we enlarge the set Y (by setting $\sigma_{s+1} = \sigma_s{}^\frown\langle 1\rangle$ at the current stage s) until we see that one of the following conditions holds: either the function p_e is not injective, or we have $p_e(z) \in Y$ for some z. Let $s_0 + 1$ be the first stage at which we witness such a situation. We set $\sigma_{s_0+1} = \sigma_{s_0}{}^\frown\langle 0\rangle$, close the P_e-cycle, and move to satisfying the P_{e+1}-strategy (by opening the P_{e+1}-cycle).

The construction

At the beginning of any nonzero stage of the construction we assume that there exists exactly one open P-cycle. Section 2.2 will guarantee that the various checks involving p_e-computations and their convergence are primitive recursive.

Stage 0

Define $\sigma_0 = \lambda$; *open the P_0-cycle*. Thus at the beginning of the next stage, there will be exactly one open P-cycle, namely the P_0-cycle.

Stage $s + 1$

Assume that the P_e-cycle is the currently open cycle. We distinguish three cases:

(1) There are $l, m \leqslant s$ such that $p_e(l)[s]\downarrow = p_e(m)[s]\downarrow$: if so, *close the P_e-cycle*, define $\sigma_{s+1} = \sigma_s{}^\frown\langle 0\rangle$, and *open the P_{e+1}-cycle*.
(2) There is $m \leqslant s$ such that $p_e(m)[s]\downarrow = z$ and $\sigma_s(z)\downarrow = 1$: do the same as in (1).
(3) Otherwise: keep the P_e-cycle open and define $\sigma_{s+1} = \sigma_s{}^\frown\langle 1\rangle$.

Again it is immediate to see that the next stage will inherit from this stage exactly one open P-cycle.

The verification

The verification relies on the following lemmas.

Lemma 4.7. *Every P_e-cycle is eventually opened and is later closed forever.*

Proof. Notice that the P_e-cycle is opened at stage s if and only if $e = 0$ and $s = 0$, or $e > 0$ and we close at s the P_{e-1}-cycle.

So assume by induction that the lemma is true of every $i < e$: thus there exists a unique s_0 at which we open the P_e-cycle ($s_0 = 0$ if $e = 0$, or otherwise s_0 is the stage at which we close the P_{e-1}-cycle). If the P_e-cycle does not satisfy the claim then the construction implies the following: the function p_e is injective and $\sigma_{s+1} = \sigma_s{}^\frown\langle 1\rangle$ for all $s \geqslant s_0 + 1$. Therefore, since p_e is injective and Y is cofinite, there would be infinitely many elements m with $p_e(m) \in Y$. Thus, at some stage $s_1 + 1 \geqslant s_0 + 1$, the P_e-cycle would be closed by the item (2) of the construction, and never opened again. We obtain a contradiction, showing that the P_e-cycle satisfies the claim. □

Lemma 4.8. *Y is primitive recursive and co-infinite.*

Proof. It is enough to observe that for all s, the value of σ_s is decided at stage s, and therefore the function $s \mapsto \sigma_s$ is primitive recursive. Moreover, it is immediate to check that $l^{\sigma_s} = s$, for every s. Hence, $Y = \bigcup_{s\in\omega} \sigma_s$ is primitive recursive, as $Y(s) = \sigma_{s+1}(s)$.

Since every P_e-cycle eventually closes, there are infinitely many stages s with $\sigma_{s+1} = \sigma_s{}^\frown\langle 0\rangle$. This implies that the set $\overline{Y}$ is infinite. □

Lemma 4.9. *All P-requirements are satisfied.*

Proof. Suppose that p_e is an injective function. Consider the stage s_0 at which the P_e-cycle closes. Clearly, the closure of the P_e-cycle is triggered by item (2). Thus, there is a number $m \leqslant s_0$ with $\sigma_{s_0}(p_e(m))\downarrow = 1$. Hence, we have $p_e(m) \in Y$ and $\mathrm{range}(p_e) \cap Y \neq \emptyset$. □

Proposition 4.6 is proved. □

This concludes the proof of Theorem 4.4. □

4.2. Counterexamples to reducibilities

In the rest of the paper, we will often need to build a primitive recursive S such that, for a given R, we have $R \nleqslant_{pr} S$. To do so, we construct a primitive recursive set Y such that for every e, the requirement

$$P_e : p_e \text{ does not reduce } R \text{ to } R_Y,$$

is satisfied, and thus $S = R_Y$ is our desired equivalence relation. Typically, we construct an increasing sequence $\{\sigma_s : s \in \omega\}$ of strings in $2^{<\omega}$ so that $Y = \bigcup_s \sigma_s$.

At a stage $s + 1$ of the construction we say that p_e *shows a counterexample to* $R \leqslant_{pr} R_Y$ if there exist $l, m \leqslant s$ so that $p_e(l)[s]\downarrow$, $p_e(m)[s]\downarrow$, and $p_e(l)[s], p_e(m)[s] < l^{\sigma_s}$, and

$$l\,R\,m \;\Leftrightarrow\; (\sigma_s(p_e(l)) \neq \sigma_s(p_e(m))) \text{ or } (p_e(l) \neq p_e(m)\ \&\ \sigma_s(p_e(l)) = \sigma_s(p_e(m)) = 0).$$

A counterexample will guarantee, for the final Y, that

$$l\,R\,m \;\Leftrightarrow\; p_e(l)\,\not{R}_Y\,p_e(m),$$

as desired. Note that there is a primitive recursive procedure which, for a given σ_s, checks whether a counterexample is being shown.

Similarly, if for every e we want to satisfy the requirement

Q_e : p_e is not a reduction from R_Y to R,

at a stage $s+1$ of the construction we say that *p_e shows a counterexample to $R_Y \leqslant_{pr} R$* if there exist $l, m < l^{\sigma_s}$ so that $l \neq m$, $p_e(l)[s]\downarrow$, $p_e(m)[s]\downarrow$, and

$$\sigma_s(l) = \sigma_s(m) = 1 \;\Leftrightarrow\; p_e(l) \not{R}\, p_e(m) :$$

a counterexample guarantees for the final Y that $R_Y \not\leqslant_{pr} R$.

Finally, sometimes we build two primitive recursive sets X, Y in the form $X = \bigcup_s \sigma^X_s$, and $Y = \bigcup_s \sigma^Y_s$. At a stage $s+1$ of the construction we say that *p_e shows a counterexample to $R_X \leqslant_{pr} R_Y$* if there exist $l, m < l^{\sigma^X_s}$ so that $l \neq m$, $p_e(l)[s]\downarrow$, $p_e(m)[s]\downarrow$, and $p_e(l)[s], p_e(m)[s] < l^{\sigma^Y_s}$, and

$$\sigma^X_s(l) = \sigma^X_s(m) = 1 \Leftrightarrow (\sigma^Y_s(p_e(l)) \neq \sigma^Y_s(p_e(m))) \text{ or } (p_e(l) \neq p_e(m) \,\&\, \sigma^Y_s(p_e(l)) = \sigma^Y_s(p_e(m)) = 0).$$

4.3. Incomparability

In **Peq**, there is no least degree. In fact, we will now prove that Id is the only primitive recursive equivalence relation which is comparable with all other equivalence relations in **Peq**.

The next result will be a consequence also of Theorem 6.1 and Theorem 6.7 (to be discussed later). However, it may be useful to give a direct proof in order to introduce the incomparability strategy, which will be exploited in other constructions.

Theorem 4.10. *For any infinitary primitive recursive $R <_{pr}$ Id, there is primitive recursive S such that $S \;|_{pr}\; R$.*

Proof. Given $R <_{pr}$ Id, we build by stages a primitive recursive set Y such that $S = R_Y$ satisfies the claim. The requirements to be satisfied are

P_e : p_e is not a reduction from R to R_Y,

Q_e : p_e is not a reduction from R_Y to R.

The strategy

To satisfy P_e, one continues putting more and more fresh elements into Y, thus not increasing the number of R_Y-classes. By doing so, we will witness eventually that p_e maps two R-classes to a single R_Y-class, since the number of R-classes will outgrow the number of R_Y-classes.

To satisfy a given Q_e-requirement, we follow a dual strategy: for any fresh element, we declare the corresponding singleton as an R_Y-class. This ensures that we will find a pair of witnesses that show that p_e is not a reduction of R_Y to R, since otherwise we would have a reduction of Id to R as well.

The construction

We construct set Y in stages: in fact, at a stage s we define its initial segment σ_s of length s, and eventually we take $Y = \bigcup_{s \in \omega} \sigma_s$.

Stage 0

Define $\sigma_0 = \lambda$; *open the P_0-cycle.*

Stage $s+1$

Assume that R is the currently open cycle.

(1) $R = P_e$, for some e: If so, we distinguish two cases.

   (a) If p_e shows a counterexample to $R \leqslant_{pr} R_Y$ (see Section 4.2) then *close the P_e-cycle*. Define $\sigma_{s+1} = \sigma_s{}^\frown\langle 0\rangle$. *Open the Q_e-cycle .*

   (b) Otherwise, keep the P_e-cycle open and define $\sigma_{s+1} = \sigma_s{}^\frown\langle 1\rangle$.

(2) If $R = Q_e$, for some e, then distinguish two more cases.

   (a) If p_e shows a counterexample to $R_Y \leqslant_{pr} R$ (see Section 4.2) then *close the Q_e-cycle*. Define $\sigma_{s+1} = \sigma_s{}^\frown\langle 0\rangle$. *Open the P_{e+1}-cycle.*

   (b) Otherwise, keep the Q_e-cycle open and define $\sigma_{s+1} = \sigma_s{}^\frown\langle 0\rangle$.

This concludes the construction.

The verification

The verification relies on the following lemmas.

Lemma 4.11. *Y is primitive recursive.*

Proof. The function $s \mapsto \sigma_s$ is primitive recursive; moreover $l^{\sigma_s} = s$. Hence, Y is primitive recursive, as $Y(s) = \sigma_{s+1}(s)$. $\square$

Lemma 4.12. *All P-requirements are satisfied.*

Proof. The proof follows the lines of the analogous claim in the proof of Theorem 4.4. First of all, it easily follows by induction that the P_e-cycle is opened at stage 0 if $e = 0$, and at the stage at which the Q_{e-1}-cycle is closed if $e > 0$; and the Q_e-cycle is opened at the stage at which the P_e-cycle is closed. Assume that for every $i < e$ the P_i-cycle and the Q_i-cycle have been closed, and the corresponding requirements are satisfied. Then at the stage s_0 (with $s_0 = 0$ if $e = 0$, or s_0 is the stage when we close the Q_{e-1}-cycle) we open the P_e-cycle. Failure to close the P_e-cycle would entail that Y is cofinite, thus R_Y would have only finitely many classes, but p_e never showing a counterexample would give that $p_e \colon R \leqslant_{pr} R_Y$, a contradiction as R is not finite. Thus, at some stage, p_e shows a counterexample to $R \leqslant_{pr} R_Y$, whence P_e is satisfied. $\square$

Lemma 4.13. *All Q-requirements are satisfied.*

Proof. Assume that for every $i < e$ the Q_i-cycle has been closed, and for every $i \leqslant e$ the P_i-cycle has been closed, and the corresponding requirements are satisfied. If s_0 is the stage at which we open the Q_e-cycle (that is when we close the P_e-cycle) and for every $s \geqslant s_0 + 1$ we never close the cycle then this would entail that Y is finite (whence $R_Y \equiv_{pr} \mathrm{Id}$), but as p_e never shows a counterexample, this would give that $p_e : R_Y \leqslant_{pr} R$, whence $\mathrm{Id} \leqslant_{pr} R$, a contradiction. Thus, at some stage, p_e shows a counterexample to $R_Y \leqslant_{pr} R$, whence Q_e is satisfied. $\square$

This concludes the proof of Theorem 4.10. $\square$

From the last theorem, it follows that there is an infinite antichain of degrees in **Peq**.

Corollary 4.14. *There are primitive recursive equivalence relations $\{S_i\}_{i\in\omega}$ such that $S_i \mid_{pr} S_j$ for all $i \neq j$.*

Proof. The proof relies on the following observation

Observation. Suppose that $\{R_i : i \in \omega\}$ is a family of infinitary primitive recursive equivalence relations none of which is pr-equivalent to Id, and for which there exists a primitive recursive predicate $U(i, x, y)$ such that $x \mathrel{R_i} y$ if and only if $U(i, x, y)$. Then a straightforward modification of the proof of Theorem 4.10 will show that there exists a primitive recursive S such that $R_i \mid_{pr} S$, for every i. The construction in this case aims to build a primitive recursive set Y satisfying the following requirements $R_{\langle e,i\rangle}$ indexed by the values of the primitive recursive Cantor pairing function:

$$P_{\langle e,i\rangle} : p_e \text{ does not reduce } R_i \text{ to } R_Y,$$

$$Q_{\langle e,i\rangle} : p_e \text{ does not reduce } R_Y \text{ to } R_i.$$

The construction goes by opening and closing R-cycles as in the proof of the previous theorem. Checking if a counterexample is being shown is primitive recursive, since this can be done by using the primitive recursive predicate U, together with Section 2.2.

To finish the proof of the corollary, define by induction the following infinite antichain $\{S_n\}_{n\in\omega}$ of primitive recursive equivalence relations. Pick an infinitary $S_0 <_{pr}$ Id; having found $S_0, \dots, S_n$, apply the above observation to the family $\{R_i\}_{i\in\omega}$, where $R_i = S_i$ if $i < n$, and $R_i = S_n$ if $i \geqslant n$. □

5. The structure Peq is a distributive lattice

In this section, we study the structure of the pr-degrees under joins and meets. By the Normal Form Theorem we will confine ourselves to equivalence relations of the form R_X, where X is a primitive recursive set, and X is infinite and coinfinite.

The first result of this section provides us with a useful characterization of the reducibility $\leqslant_{pr}$. Informally speaking, the characterization connects the pr-degree of a relation R_X with the growth rate of the function

$$\#(0^X)[s] := \#(0^{X\upharpoonright(s+1)}),$$

i.e., the function which counts the number of singleton R_X-classes.

We emphasize that for a primitive recursive X, the corresponding function $\#(0^X)[s]$ is also primitive recursive.

Proposition 5.1. *Let X and Y be co-infinite primitive recursive sets. Then we have $R_X \leqslant_{pr} R_Y$ if and only if there exists a primitive recursive function $h(x)$ such that for all s,*

$$\#(0^X)[s] \leqslant \#(0^Y)[h(s)].$$

Proof. Suppose that $f\colon R_X \leqslant_{pr} R_Y$. By Remark 3.5, one may assume that for any pair of elements $k < l$ from $\overline{X}$, we have $f(k) < f(l)$. In addition, by Lemma 3.3, we assume that $f[X] \subseteq Y$. The desired primitive recursive function $h(s)$ is defined as follows:

$$h(s) = \begin{cases} f(k^*), & \text{if } k^* \text{ is the greatest number such that } k^* \leqslant s \text{ and } k^* \in \overline{X},\\ 0, & \text{if } (\forall k \leqslant s)(k \in X). \end{cases}$$

Indeed, suppose that $\#(0^X)[s] = N > 0$. Consider the set

$$\overline{X} \cap \{0, 1, \dots, s\} = \{k_1 < k_2 < \dots < k_N\}.$$

Then we have $f(k_1) < f(k_2) < \dots < f(k_N) = h(s)$, and hence, $\#(0^Y)[h(s)] \geqslant N = \#(0^X)[s]$.

To show the converse, let h be a primitive recursive function such that $\#(0^X)[s] \leqslant \#(0^Y)[h(s)]$ for all s. Without loss of generality, one may assume that $0 \in Y$. The desired primitive recursive reduction $g\colon R_X \leqslant_{pr} R_Y$ is defined by recursion on $x \in \omega$ as follows:

$$g(0) = \begin{cases} 0, & \text{if } 0 \in X, \\ \text{the least element from } \overline{Y}, & \text{if } 0 \in \overline{X}; \end{cases}$$

$$g(x+1) = \begin{cases} 0, & \text{if } x+1 \in X, \\ \mu y[y \leqslant h(x+1) \,\&\, y \in \overline{Y} \setminus \{g(z) : z \leqslant x\}], & \text{if } x+1 \in \overline{X}. \end{cases}$$

Suppose that $x \in \overline{X} \setminus \{0\}$ and the set $\overline{X} \cap \{0,1,\dots,x\}$ equals $\{k_1 < k_2 < \cdots < k_N = x\}$. Then the set $\overline{Y} \cap \{1,2,\dots,h(x)\}$ contains at least N elements, and this set has at most $N-1$ elements from range$(g \restriction x)$. Therefore, the function g is well-defined. In addition, it is easy to observe that g provides a reduction $R_X \leqslant_{pr} R_Y$. □

Proposition 4.5 can now be restated as:

Corollary 5.2. Id $\leqslant_{pr} R_Y$ *if and only if there is a primitive recursive function $h(x)$ such that $s \leqslant \#(0^Y)[h(s)]$ for all $s \in \omega$.*

5.1. Joins and meets

Now we are ready to prove that the partial order **Peq** has joins and meets, which make the structure a lattice. By slightly abusing notations, we will talk about suprema and infima of primitive recursive equivalence relations (referring of course to the poset **Peq** of the *pr*-degrees).

Theorem 5.3. *The structure* **Peq** *is a lattice.*

Proof. Suppose that X and Y are co-infinite primitive recursive sets such that $R_X \mid_{pr} R_Y$. Without loss of generality, we assume that $0 \in X \cap Y$. In order to prove the theorem, it is sufficient to show that the relations R_X and R_Y have supremum and infimum.

We define a set $Z_0 \subseteq \omega$ as follows: $0 \in Z_0$, and

$$s+1 \notin Z_0 \;\Leftrightarrow\; \max(\#(0^X)[s+1], \#(0^Y)[s+1]) > \max(\#(0^X)[s], \#(0^Y)[s]).$$

Recall that the functions $\#(0^X)[s]$ and $\#(0^Y)[s]$ are primitive recursive. Hence, it is easy to show that the set Z_0 is primitive recursive and co-infinite.

Claim 5.4. *R_{Z_0} is the supremum of R_X and R_Y.*

Proof. First, we note the following: $\#(0^Z)[0] = \#(0^X)[0] = \#(0^Y)[0] = 0$, and every set U satisfies

$$\#(0^U)[s+1] = \begin{cases} \#(0^U)[s] + 1, & \text{if } s+1 \notin U, \\ \#(0^U)[s], & \text{if } s+1 \in U. \end{cases}$$

These observations (together with an easy induction argument) imply that

$$\#(0^{Z_0})[s] = \max(\#(0^X)[s], \#(0^Y)[s]). \tag{1}$$

Thus, one can apply Proposition 5.1 for the function $h(x) = x$, and deduce that R_{Z_0} is an upper bound for both R_X and R_Y.

Now suppose that R_V is an arbitrary upper bound of R_X and R_Y. By Proposition 5.1, we choose primitive recursive functions h_X and h_Y such that $\#(0^X)[s] \leqslant \#(0^V)[h_X(s)]$ and $\#(0^Y)[s] \leqslant \#(0^V)[h_Y(s)]$. Then by (1), we obtain

$$\#(0^{Z_0})[s] \leqslant \max(\#(0^V)[h_X(s)], \#(0^V)[h_Y(s)]) = \#(0^V)[\max(h_X(s), h_Y(s))].$$

Hence, we deduce that $R_{Z_0} \leqslant_{pr} R_V$, and R_{Z_0} is the join of R_X and R_Y. □

Let now Z_1 be the set determined by the following: $0 \in Z_1$, and

$$s+1 \notin Z_1 \iff \min(\#(0^X)[s+1], \#(0^Y)[s+1]) > \min(\#(0^X)[s], \#(0^Y)[s]).$$

Claim 5.5. *R_{Z_1} is the infimum of R_X and R_Y.*

Proof. As in the previous claim, one can easily show that

$$\#(0^{Z_1})[s] = \min(\#(0^X)[s], \#(0^Y)[s]). \tag{2}$$

By Proposition 5.1, R_{Z_1} is a lower bound for R_X and R_Y.

Let R_V be a lower bound of R_X and R_Y. We fix primitive recursive functions q_X and q_Y such that $\#(0^V)[s] \leqslant \#(0^X)[q_X(s)]$ and $\#(0^V)[s] \leqslant \#(0^Y)[q_Y(s)]$. Then by (2),

$$\begin{aligned}\#(0^V)[s] &\leqslant \min(\#(0^X)[q_X(s)], \#(0^Y)[q_Y(s)]) \leqslant \min(\#(0^X)[\max(q_X(s), q_Y(s))], \#(0^Y)[\max(q_X(s), q_Y(s))]) \\ &= \#(0^{Z_1})[\max(q_X(s), q_Y(s))].\end{aligned}$$

Therefore, R_{Z_1} is the meet of R_X and R_Y. □

Theorem 5.3 is proved. □

Definition 5.6. Given primitive recursive sets X and Y, let us denote $R_X \vee R_Y$ the relation R_{Z_0}, constructed in the proof of Theorem 5.3, giving the supremum of R_X and R_Y. In addition, let us denote $Z_0 = X \vee Y$. Likewise, let us denote $R_X \wedge R_Y$ the relation R_{Z_1}, constructed in the proof above, giving the infimum of R_X and R_Y. We also denote $Z_1 = X \wedge Y$.

Corollary 5.7. *There are no minimal pairs inside* **Peq**.

Proof. Immediate. □

Note that in the proof of Theorem 5.3, we gave an explicit algorithm for building suprema and infima. This allows us to easily obtain the following:

Theorem 5.8. *The lattice* **Peq** *is distributive.*

Proof. Let X, Y, and Z be co-infinite primitive recursive sets. As discussed above, by $R_X \vee R_Y$ we denote the supremum of R_X and R_Y, and $R_X \wedge R_Y$ is the infimum of R_X and R_Y. We sketch the proof for the following distributivity law:

$$R_X \vee (R_Y \wedge R_Z) \equiv_{pr} (R_X \vee R_Y) \wedge (R_X \vee R_Z).$$

Suppose that $Q \equiv_{pr} R_X \vee (R_Y \wedge R_Z)$ and $S \equiv_{pr} (R_X \vee R_Y) \wedge (R_X \vee R_Z)$. We may assume that $Q = R_U$ and $S = R_V$, where U and V are primitive recursive sets such that $0 \in U \cap V$, and for every $s \in \omega$,

$$\#(0^U)[s] = \max(\#(0^X)[s], \min(\#(0^Y)[s], \#(0^Z)[s])),$$
$$\#(0^V)[s] = \min(\max(\#(0^X)[s], \#(0^Y)[s]), \max(\#(0^X)[s], \#(0^Z)[s])).$$

Since the structure $(\omega, \leqslant)$ is a linear order, for any numbers $x, y, z \in \omega$, we have

$$\max(x, \min(y, z)) = \min(\max(x, y), \max(x, z)).$$

Hence, it is clear that $\#(0^U)[s] = \#(0^V)[s]$ for every s, and $R_U = R_V$. This concludes the proof of Theorem 5.8. $\square$

6. Density

We prove now that the distributive lattice **Peq** is dense. This contrasts with the case of **Ceers** and **ER**, where each degree has a minimal cover (see [13, 18] for details). However, density is a phenomenon that often shows up when focusing on the subrecursive world. Mehlhorn [32] proved that the degree structures induced by many subrecursive reducibilities on sets (including the primitive recursive one) are dense. Similarly, Ladner [33] proved that if $\mathbf{P} \neq \mathbf{NP}$, then the poset of **NP** sets under polynomial-time reducibility is dense.

Density emerges also in the study of the online content of structures. More precisely, for a punctual structure $\mathcal{A}$, $\mathbf{FPR}(\mathcal{A})$ denotes the degree structure generated by primitive recursive isomorphisms on the collection of all punctual copies of $\mathcal{A}$. Bazhenov, Kalimullin, Melnikov, and Ng [34] recently proved the following: if a punctual infinite $\mathcal{A}$ is finitely generated, then the poset $\mathbf{FPR}(\mathcal{A})$ is dense.

Theorem 6.1 (Density). *If $R_X <_{pr} R_Z$ are infinitary primitive recursive equivalence relations, then there exists a primitive recursive set Y such that $R_X <_{pr} R_Y <_{pr} R_Z$.*

Proof. We will satisfy the following requirements, for every $e \in \omega$:

P_e : p_e does not reduce R_Y to R_X,

Q_e : p_e does not reduce R_Z to R_Y,

M : $R_X \leqslant_{pr} R_Y$,

N : $R_Y \leqslant_{pr} R_Z$,

where $\{p_e\}_{e\in\omega}$ is a computable listing of all primitive recursive functions, see Remark 2.2.

Assume that $f\colon R_X \leqslant_{pr} R_Z$. Assume also, without loss of generality, that $0 \in X \cap Z$, and that Z is infinite.

The environment

At stage $s+1$ we inherit from stage s a finite binary string σ_s^Y of length $s+1$; moreover we will let $\sigma_s^X = X \restriction s+1$ and $\sigma_s^Z = Z \restriction s+1$. For $U \in \{X, Y, Z\}$, we will denote $\#\left(0^U\right)[s] = \#\left(0^{\sigma_s^U}\right)$ (see Section 2.3 for the notation $\#\left(0^\tau\right)$, where τ is a finite binary string).

The strategies

Let us sketch the strategy to achieve $R_Y \nleqslant_{pr} R_X$. When we attack for the first time the requirement at stage s_0, we are given the strings $\sigma_{s_0}^X, \sigma_{s_0}^Y, \sigma_{s_0}^Z$, for which we have guaranteed that $\#\left(0^Y\right)[s_0] = \#\left(0^Z\right)[s_0]$.

We open the so called P_e-cycle: until p_e does not show a counterexample to $R_Y \leqslant_{pr} R_X$, we keep copying larger and larger pieces of Z in Y, so that starting from the input s_0+1, the set Y looks like Z from the input s_0+1. If this process goes on forever, then we would eventually get $R_Z \leqslant_{pr} R_Y$: the initial segment $\sigma^Z_{s_0}$ of Z which is not copied by the copying procedure can be mapped by the reduction to $\sigma^Y_{s_0}$ as the two strings have the same number of 0's; if $i < s_0 + 1$ is such that $\sigma^Z_{s_0}(i) = 1$ (i.e., $Z(i) = 1$), then the reduction maps i to $0 \in Y$. Thus, eventually we get that p_e does show a counterexample to $R_Y \leqslant_{pr} R_X$, otherwise $R_Y \leqslant_{pr} R_X$, but then $R_Z \leqslant_{pr} R_Y \leqslant_{pr} R_X$.

When a counterexample shows up, we close the P_e-cycle and we move to the next requirement, opening the Q_e-cycle. (In fact, before opening the Q_e-cycle, we have to go through a transition phase to reach a stage t at which $\#\left(0^Y\right)[t] = \#\left(0^X\right)[t]$.) This also shows that P_e is eventually satisfied.

The strategy to achieve $R_Z \nleqslant_{pr} R_Y$ is similar, opening and closing the so called Q_e-cycle: until p_e does not show a counterexample to $R_Z \leqslant_{pr} R_Y$, we keep copying larger and larger pieces of X in Y, so that starting from the input s_0+1 (where s_0 is when the cycle was opened) the set Y looks like X from s_0+1. In order to implement this procedure in a correct way, we require the following: when we start the Q_e-cycle at s_0, we have $\#\left(0^Y\right)[s_0] = \#\left(0^X\right)[s_0]$.

Again, the Q_e-cycle cannot go on forever, otherwise we would get $R_Z \leqslant_{pr} R_Y$ (since p_e never shows a counterexample), but on the other hand the copying procedure would give $R_Y \leqslant_{pr} R_X$, yielding a contradiction. After a counterexample shows up, there will be a transition phase, at the end of which we will reach a stage t at which $\#\left(0^Y\right)[t] = \#\left(0^Z\right)[t]$.

It remains to explain how we achieve that $R_X \leqslant_{pr} R_Y \leqslant_{pr} R_Z$. For this, we guarantee that at each step s we have

$$\#\left(0^X\right)[s] \leqslant \#\left(0^Y\right)[s] \leqslant \#\left(0^Z\right)[s],$$

so that we can search in a bounded way for the images in Y of the 0's in σ^X_s, and for the images in Z of the 0's in σ^Y_s. This, together with the facts that s is in the domains of both σ^X_s and σ^Y_s, and the mappings $s \mapsto \sigma^U_s$ are primitive recursive, will give the desired reductions.

Remark 6.2. As $R_X \leqslant_{pr} R_Z$, we may assume that if $\sigma \subset X, \tau \subset Z$ have the same length, then $\#\left(0^\sigma\right) \leqslant \#\left(0^\tau\right)$: for this, one can replace Z with the join $X \vee Z$ if needed.

The construction

The construction is in stages.

Stage 0

Let $Y(0) = 1$, and $\sigma^Y_0 = \lambda$. *Open the P_0-cycle*. Notice that $\#\left(0^X\right)[0] = \#\left(0^Y\right)[0] = \#\left(0^Z\right)[0] = 0$.

Stage $s + 1$

We distinguish two relevant cases:

Case 1) Suppose that we are within a previously opened P_e-cycle which has not been declared closed yet. We assume by induction that when we opened (say at s_0) the cycle, we had $\#\left(0^X\right)[s_0] \leqslant \#\left(0^Y\right)[s_0] = \#\left(0^Z\right)[s_0]$.

Copying phase

(Copy R_Z in R_Y.) If we have not yet moved to the $P_e \to Q_e$-transition phase, then let

$$\sigma^Y_{s+1} = \sigma^Y_s{}^\frown\langle Z(s+1)\rangle.$$

Notice that by the assumption in Remark 6.2, after this we still have

$$\#\left(0^X\right)[s+1] \leqslant \#\left(0^Y\right)[s+1] = \#\left(0^Z\right)[s+1].$$

After this, if p_e has shown a counterexample to $R_Y \leqslant_{pr} R_X$ (as defined in Section 4.2) then enter the *$P_e \to Q_e$-transition phase*:

Transition phase

Carry out the following.

(1) If $\#\left(0^X\right)[s+1] = \#\left(0^Y\right)[s+1]$, then exit from the transition phase. We *close the P_e-cycle* and *open the Q_e-cycle*.
(2) If $\#\left(0^X\right)[s+1] < \#\left(0^Y\right)[s+1]$, then let

$$\sigma^Y_{s+1} = \sigma^Y_s{}^\frown\langle 1\rangle :$$

(when $X(s+1)=0$, this has the effect of making $\#\left(0^X\right)[s+1] = \#\left(0^X\right)[s]+1$, whereas $\#\left(0^Y\right)[s+1] = \#\left(0^Y\right)[s]$) and go to (1), remaining in this transition phase.

Notice that at each stage t within a P_e-cycle we have by the assumption in Remark 6.2

$$\#\left(0^X\right)[t] \leqslant \#\left(0^Y\right)[t] \leqslant \#\left(0^Z\right)[t],$$

and when we close the P_e-cycle, we have

$$\#\left(0^X\right)[t] = \#\left(0^Y\right)[t] \leqslant \#\left(0^Z\right)[t].$$

CASE 2) Suppose that we are within a previously opened Q_e-cycle which has not been declared closed yet. We assume by induction that when we opened (say at s_0) the cycle, we had $\#\left(0^X\right)[s_0] = \#\left(0^Y\right)[s_0] \leqslant \#\left(0^Z\right)[s_0]$.

Copying phase

(Copy R_X in R_Y.) Let

$$\sigma^Y_{s+1} = \sigma^Y_s{}^\frown\langle X(s+1)\rangle.$$

Notice that by the assumption in Remark 6.2, after this we still have $\#\left(0^X\right)[s+1] = \#\left(0^Y\right)[s+1] \leqslant \#\left(0^Z\right)[s+1]$ if we had $\#\left(0^X\right)[s] = \#\left(0^Y\right)[s] \leqslant \#\left(0^Z\right)[s]$. After this, if p_e has shown a counterexample to $R_Z \leqslant_{pr} R_Y$ (as defined in Section 4.2), then enter the *$Q_e \to P_{e+1}$-transition phase*:

Transition phase

Carry out the following.

(1) If $\#\left(0^Y\right)[s+1] = \#\left(0^Z\right)[s+1]$, then exit from the transition phase. We *close the Q_e-cycle* and *open the P_{e+1}-cycle*.
(2) If $\#\left(0^Y\right)[s+1] < \#\left(0^Z\right)[s+1]$, then let

$$\sigma^Y_{s+1} = \sigma^Y_s{}^\frown\langle 0\rangle :$$

(when $Z(s+1)=1$, this has the effect of making $\#\left(0^Y\right)[s+1] = \#\left(0^Y\right)[s]+1$ whereas $\#\left(0^Z\right)[s+1] = \#\left(0^Z\right)[s]$) and go to (1), remaining in this transition phase.

Notice that at each stage t within a Q_e-cycle we have by the assumption in Remark 6.2

$$\#\left(0^X\right)[t] \leqslant \#\left(0^Y\right)[t] \leqslant \#\left(0^Z\right)[t],$$

and when we close the Q_e-cycle, we have

$$\#\left(0^X\right)[t] \leqslant \#\left(0^Y\right)[t] = \#\left(0^Z\right)[t].$$

The verification

The verification relies on the following lemmas.

Lemma 6.3. *For each e, the requirements P_e and Q_e are satisfied.*

Proof. As in the proof of Theorem 4.10, it easily follows by induction that the P_e-cycle is opened at stage 0 if $e = 0$, and at the stage at which the Q_{e-1}-cycle is closed if $e > 0$. The Q_e-cycle is opened at the stage at which the P_e-cycle is closed.

Assume that for every $i < e$ the P_i-cycle and the Q_i-cycle have been closed, and the corresponding requirements are satisfied. Then at the stage s_0 (with $s_0 = 0$ if $e = 0$, or s_0 is the stage when we close the Q_{e-1}-cycle if $e > 0$) we open the P_e-cycle. If p_e never shows a counterexample to $R_Y \leqslant_{pr} R_X$, then we claim that $R_Z \leqslant_{pr} R_X$, a contradiction.

To show this claim, notice that in this case (i.e., should p_e never show a counterexample to $R_Y \leqslant_{pr} R_X$, implying that $R_Y \leqslant_{pr} R_X$) we would have $Y = \sigma^Y_{s_0} * Z$. Then $R_Z \leqslant_{pr} R_Y$ by a primitive recursive function q which matches up the zeros in $\sigma^Z_{s_0}$ with those of $\sigma^Y_{s_0}$ (using that both strings have the same number of zeros, since $\#\left(0^Y\right)[s_0] = \#\left(0^Z\right)[s_0]$), $q(i) = 0$ if $Z(i) = 1$ and $i \leqslant s_0$, and $q(i) = i$ for $i \geqslant s_0 + 1$. It would follow that $R_Z \leqslant_{pr} R_X$, a contradiction.

Thus, at some stage p_e shows a counterexample to $R_Y \leqslant_{pr} R_X$, whence P_e is satisfied. Moreover, since $\overline{X}$ is infinite, the transition phase of the cycle will end, since eventually X will produce enough 0's to match up with those which are present in σ^Y at the beginning of the $P_e \to Q_e$-transition phase of the P_e-cycle. Therefore, the cycle will be closed.

Similarly, assume that for every $i < e$ the Q_i-cycle has been closed, and for every $i \leqslant e$ the P_i-cycle has been closed, and the corresponding requirements are satisfied. If s_0 is the stage at which we open the Q_e-cycle (that is when we close the P_e-cycle) and for every $s \geqslant s_0$ we never close the cycle, then $R_Z \leqslant_{pr} R_Y$, and thus an argument similar to the one given above would entail that R_Z would be pr-reducible to R_X, as the construction would ensure in this case that $Y = \sigma^Y_{s_0} * X$ and $\#\left(0^X\right)[s_0] = \#\left(0^Y\right)[s_0]$, giving that $R_Y \leqslant_{pr} R_X$. Finally, the $Q_e \to P_{e+1}$-transition phase ends, since Z is infinite.

Hence, all P- and Q- requirements are satisfied. □

Claim 6.4. *Y is primitive recursive.*

Proof. The function $s \mapsto \sigma^Y_s$ is primitive recursive, and $Y(s) = \sigma^Y_{s+1}(s)$. □

Lemma 6.5. *$R_X \leqslant_{pr} R_Y \leqslant_{pr} R_Z$.*

Proof. We need to define two primitive recursive functions g, h which provide reductions $g\colon R_X \leqslant_{pr} R_Y$ and $h\colon R_Y \leqslant_{pr} R_Z$. Using that the functions q^Y, q^Z where $q^Y(s) = \sigma^Y_s$ and $q^Z(s) = \sigma^Z_s$ are primitive recursive, and at each stage t we have that $\#\left(0^X\right)[t] \leqslant \#\left(0^Y\right)[t]$, define

$$g(s) = \begin{cases} 0, & \text{if } X(s) = 1, \\ \min\{i < l^Y_s : \sigma^Y_s(i) = 0 \,\&\, (\forall j < s)[i \neq g(j)]\}, & \text{if } X(s) = 0. \end{cases}$$

Similarly, using that at each stage t we have $\#\left(0^Y\right)[t] \leqslant \#\left(0^Z\right)[t]$, we can define

$$h(s) = \begin{cases} 0, & \text{if } Y(s) = 1, \\ \min\{i < l^Z_s : \sigma^Z_s(i) = 0 \,\&\, (\forall j < s)[i \neq h(j)]\}, & \text{if } Y(s) = 0. \end{cases}$$

It is not hard to see that g and h provide the desired pr-reductions. □

The last lemma ensures that the global requirements M and N are both satisfied. In combination with Lemma 6.3, this means that R_Y lies strictly in between R_X and R_Z, as desired. Theorem 6.1 is proved. □

Upwards and downwards density are immediate consequences of Theorem 4.10, the existence of infima (Theorem 5.3), and Theorem 6.1:

Corollary 6.6. *If $R <_{pr} \mathrm{Id}$ and R is infinitary, then there are infinitary S_0 and S_1 such that*

$$S_0 <_{pr} R <_{pr} S_1 <_{pr} \mathrm{Id}.$$

Proof. Upwards density (i.e., the existence of S_1) is a particular case of Theorem 6.1. For downward density (i.e., the existence of S_0), recall that if R is in **Peq**, then by Theorem 4.10, there exists S such that $R \mid_{pr} S$: thus, $S_0 := R \wedge S$ is a primitive recursive equivalence relation such that $S_0 <_{pr} R$. □

We now combine the density strategy of the previous theorem with the incomparability strategy exploited in Theorem 4.10.

Theorem 6.7 (Density plus incomparability)**.** *If $R_X <_{pr} R_T <_{pr} R_Z$ are infinitary primitive recursive equivalence relations, then there exists a primitive recursive set Y such that $R_X <_{pr} R_Y <_{pr} R_Z$ and $R_T \mid_{pr} R_Y$.*

Proof. Suppose that $R_X <_{pr} R_T <_{pr} R_Z$ are primitive recursive equivalence relations. To build Y, a trivial modification of Theorem 6.1 suffices.

In the previous proof, we close the P_e-cycle in Case 1 of Step $s+1$ when we see that p_e has shown a counterexample to $R_Y \leqslant_{pr} R_X$. For the purpose of the present proof, we now ask to *close the P_e-cycle* in Case 1 of Stage $s+1$ when we have seen that p_e has shown a counterexample to $R_Y \leqslant_{pr} R_T$, and we have matched up through the transition phase $\#\left(0^X\right) = \#\left(0^Y\right)$: should p_e never show a counterexample to $R_Y \leqslant_{pr} R_T$, then (as in the proof of Theorem 6.1) our copying phase of Case 1 would end up with making $R_Z \leqslant_{pr} R_Y$, giving $R_Z \leqslant_{pr} R_T$, a contradiction.

Similarly, here we ask to *close the Q_e-cycle* in Case 2 of Step $s+1$ when we see that p_e shows a counterexample to $R_T \leqslant_{pr} R_Y$, and we have matched up through the transition phase $\#\left(0^Y\right) = \#\left(0^Z\right)$. Should p_e never show a counterexample to $R_T \leqslant_{pr} R_Y$, then (as in the proof of Theorem 6.1) our copying phase of Case 2 would end up with making $R_Y \leqslant_{pr} R_X$, giving $R_T \leqslant_{pr} R_X$, a contradiction. □

Remark 6.8. Notice that the two previous theorems provide another proof of Theorem 4.10: Indeed, given an infinitary primitive recursive $R <_{pr} \mathrm{Id}$, it is enough to pick by Corollary 6.6 infinitary relations S_0, S_1 such that $S_0 <_{pr} R <_{pr} S_1$, so that by density plus incomparability there exists $S \mid_{pr} R$, with the stronger specification that S lies between S_0 and S_1.

Notice also that by a straightforward extension of the argument in Theorem 6.7 (in the same vein as in the argument for Corollary 4.14), one can show that if $R <_{pr} S$ inside **Peq**, then one can build an infinite antichain whose members all lie between R and S.

7. Join- and meet-reducibility

In a poset $\langle P, \leqslant \rangle$ an element $a \in P$ is *join-reducible* if in P there are $b, c < a$ such that a is the join of b, c, and a is *meet-reducible* if there are $b, c > a$ such that a is the meet of b, c.

Before showing that in **Peq** every element is join-reducible, and every $R <_{pr} \mathrm{Id}$ is meet-reducible, let us introduce some notations and simple observations which will be useful in the rest of this section.

In analogy with the principal function $p_{\overline{X}}$ of the complement $\overline{X}$ (where $X \subseteq \omega$), given a string $\sigma \in 2^{<\omega}$, let also $p_{\overline{\sigma}}$ denote the order preserving finite bijection $p_{\overline{\sigma}} : \{n : n < \#\left(0^\sigma\right)\} \longrightarrow \overline{\sigma}$ (the notation $\overline{\sigma}$ has been introduced in Section 2.3). Again in analogy with what we have done for sets (see Definition 5.6), we give the following definition.

Definition 7.1. Given $\sigma, \tau \in 2^{<\omega}$ such that $l^{\sigma} = l^{\tau} = h$ and $\#\left(0^{\sigma}\right) = \#\left(0^{\tau}\right) = m$, let $\sigma \vee \tau$ be the string with $l^{\sigma\vee\tau} = h$ and such that $(\sigma \vee \tau)(i) = 0$ if and only if $i = \min(p_{\overline{\sigma}}(n), p_{\overline{\tau}}(n))$, for some $n < m$. Dually, define $\sigma \wedge \tau$ to be the string with $l^{\sigma\wedge\tau} = h$ and such that $(\sigma \wedge \tau)(i) = 0$ if and only if $i = \max(p_{\overline{\sigma}}(n), p_{\overline{\tau}}(n))$, for some $n < m$.

Notice that for σ, τ as in the definition, we have $\#\left(0^{\sigma\vee\tau}\right) = \#\left(0^{\sigma\wedge\tau}\right) = m$.

Lemma 7.2. *Let (σ_0, σ_1) be a pair of strings such that $l^{\sigma_0} = l^{\sigma_1} = h$ and $\#\left(0^{\sigma_0}\right) = \#\left(0^{\sigma_1}\right)$; let (τ_0, τ_1) be another pair of strings such that $l^{\tau_0} = l^{\tau_1} = h'$ and $\#\left(0^{\tau_0}\right) = \#\left(0^{\tau_1}\right)$; finally, let Y_0, Y_1 be a pair of sets such that $\sigma_0 \subset Y_0$ and $\sigma_1 \subset Y_1$. Then, for an operation $\Box \in \{\vee, \wedge\}$,*

(1) *for every $i < h'$ we have*

$$(\sigma_0{}^\frown\tau_0 \Box \sigma_1{}^\frown\tau_1)(h + i) = (\tau_0 \Box \tau_1)(i);$$

(2) *$\sigma_0 \Box \sigma_1 \subset Y_0 \Box Y_1$, and for every $i < h$, we have*

$$(Y_0 \Box Y_1)(i) = (\sigma_0 \Box \sigma_1)(i).$$

Proof. The proof is immediate. Let $\#\left(0^{\sigma_0}\right) = \#\left(0^{\sigma_1}\right) = m$, and $\#\left(0^{\tau_0}\right) = \#\left(0^{\tau_1}\right) = m'$. Item (1) follows from the fact that $\sigma_0{}^\frown\tau_0 \Box \sigma_1{}^\frown\tau_1$ has $m + m'$ zeros: the first m ones of them (in order of magnitude) come from comparing the pairs $(p_{\overline{\sigma_0}}(n), p_{\overline{\sigma_1}}(n))$ with $n < m$; and the last m' ones of them come from comparing the pairs $(p_{\overline{\sigma_0{}^\frown\tau_0}}(m + n), p_{\overline{\sigma_1{}^\frown\tau_1}}(m + n)$ with $n < m'$, which amounts to comparing the pairs $(p_{\overline{\tau_0}}(n), p_{\overline{\tau_1}}(n))$ with $n < m'$.

Finally, (2) follows easily from (1). $\Box$

Theorem 7.3. *Each infinitary primitive recursive R_Z is join-reducible.*

Proof. Let R_Z be a primitive recursive equivalence relation in normal form. As $\mathrm{Id} \equiv_{pr} R_E$, with E denoting the set of even numbers, we can always assume that Z is infinite and co-infinite.

We will build $R_{Y_0}, R_{Y_1} <_{pr} R_Z$ such that $R_{Y_0} \vee R_{Y_1} = R_Z$. To do so, we will satisfy the following requirements:

P_e: p_e does not reduce R_Z to R_{Y_1},

Q_e: p_e does not reduce R_Z to R_{Y_0},

N: $R_Z = R_{Y_0} \vee R_{Y_1}$.

Notice that the N-requirement is actually requesting that $R_Z = R_{Y_0} \vee R_{Y_1}$, not just $R_Z \equiv_{pr} R_{Y_0} \vee R_{Y_1}$.

We will build Y_0, Y_1 in stages by approximating their characteristic functions, i.e., $Y_i = \bigcup_{s\in\omega} \sigma_s^{Y_i}$ for $i \in \{0, 1\}$. In the construction at each stage s we will use also the string σ_s^Z which, we recall, is the initial segment of Z with length s.

The construction

We adopt the same terminology and notations as those employed in Theorem 6.1. As in the proof of that theorem, at each stage, the construction can be either in a *copying phase* or in a *transition phase*.

Stage 0

$\sigma_0^{Y_0} = \sigma_0^{Y_1} = \lambda$. *Open the P_0-cycle*, which will be implemented starting from next stage.

Stage $s + 1$

We distinguish two cases.

Case 1. Suppose that we are within a previously opened cycle P_e, which has not been declared closed yet. We assume by induction that we have $\#\left(0^{Y_1}\right)[s + 1] \leqslant \#\left(0^{Y_0}\right)[s + 1] \leqslant \#\left(0^{Z}\right)[s + 1]$.

Copying phase

If we have not yet moved to the $P_e \to Q_e$-transition phase, then we copy R_Z into R_{Y_0}: let $\sigma^{Y_0}_{s+1} = \sigma^{Y_0}_s{}^\frown\langle Z(s)\rangle$ and $\sigma^{Y_1}_{s+1} = \sigma^{Y_1}_s{}^\frown\langle 1\rangle$. After this, if p_e shows a counterexample to $R_Z \leqslant_{pr} R_{Y_1}$, then go to the *$P_e \to Q_e$-transition phase* which will be implemented starting from the next stage.

Transition phase

Suppose that we are within the $P_e \to Q_e$-transition phase. Let $Y_1(s+1) = 0$ and $Y_0(s+1) = Z(s)$. After this, if $\#\left(0^{Y_0}\right)[s+1] = \#\left(0^{Y_1}\right)[s+1]$, then *close the P_e-cycle* and *open the Q_e-cycle* which will be implemented starting from next stage; otherwise, stay in this transition phase.

CASE 2. Suppose that we are within a previously opened Q_e-cycle, which has not been declared closed yet. We assume by induction that we have $\#\left(0^{Y_0}\right)[s+1] \leqslant \#\left(0^{Y_1}\right)[s+1] \leqslant \#\left(0^{Z}\right)[s+1]$.

Copying phase

If we have not yet moved to the $Q_e \to P_{e+1}$-transition phase, then we copy R_Z into R_{Y_1}: let $\sigma^{Y_0}_{s+1} = \sigma^{Y_0}_s{}^\frown\langle 1\rangle$ and $\sigma^{Y_1}_{s+1} = \sigma^{Y_1}_s{}^\frown\langle Z(s)\rangle$. After this, if p_e shows a counterexample to $R_Z \leqslant_{pr} R_{Y_0}$, then go to the *$Q_e \to P_{e+1}$-transition phase* which will be implemented starting from the next stage; otherwise, stay in this transition phase.

Transition phase

Suppose that we are within the $Q_e \to P_{e+1}$-transition phase. Let $Y_0(s+1) = 0$ and $Y_1(s+1) = Z(s)$. After this, if $\#\left(0^{Y_0}\right)[s+1] = \#\left(0^{Y_1}\right)[s+1]$, then *close the Q_e-cycle* and *open the P_{e+1}-cycle* which will be implemented starting from next stage; otherwise, stay in this transition phase.

Notice that for every s, the constructed initial segment $\sigma^{Y_i}_s$ of Y_i has length s.

(We note that the distinction between "copying phase" and "transition phase" can be misleading, as in the transition phase of Case 1 we still keep copying Z into Y_0 as we were doing during the copying phase, and similarly in the transition phase of Case 2 we still keep copying Z into Y_1 as we were doing during the copying phase.)

The verification

Y_0 and Y_1 are primitive recursive as $Y_i(s) = \sigma^{Y_i}_{s+1}(s)$, and the mapping $s \mapsto \sigma^{Y_i}_{s+1}$ is primitive recursive.

The rest of the verification is based on the following lemmas.

Lemma 7.4. *The P- and Q- requirements are satisfied. Moreover, if s is a stage at which we close a cycle, then* $\#\left(0^{Y_0}\right)[s] = \#\left(0^{Y_1}\right)[s]$.

Proof. As in the proof of Theorem 4.10 and Theorem 6.1, if s is any stage, then at s we are either in an open P_e- or Q_e-cycle for exactly one e.

Eventually any P- or Q- cycle will be closed. This is easily seen by induction. Suppose that at stage s_0 we open the P_e-cycle (the P_0-cycle is opened at stage 0). Then eventually p_e shows a counterexample to $R_Z \leqslant_{pr} R_{Y_1}$ (thus P_e is satisfied), otherwise our copying procedure would put all fresh elements into Y_1, giving that R_{Y_1} has only finitely many equivalence classes, contradicting that R_Z has infinitely many equivalence classes. After p_e has shown a counterexample, we start the $P_e \to Q_e$-transition phase. During the transition we keep all fresh elements out of Y_1. This makes the number of 0's of Y_1 growing as fast as possible, while we copy Z in Y_0. Eventually, we will witness a stage s at which $\#\left(0^{Y_0}\right)[s] = \#\left(0^{Y_1}\right)[s]$; otherwise, Z would be finite contradicting the fact that Z is infinite. This shows that the P_e-cycle is eventually closed, and we open the P_e-cycle.

By a similar argument we can prove that each Q_e-cycle is eventually opened, then closed, and the corresponding requirement is satisfied. □

Lemma 7.5. *The N-requirement is satisfied.*

Proof. We want to show that $Z = Y_0 \vee Y_1$. Let $s_0 < s_1 < \dots$ be the sequence of stages at which we open a cycle, with $s_0 = 0$. Our argument is by induction on the index n of s_n. Assume by induction that, for every $i < s_n$ we have $(Y_0 \vee Y_1)(i) = Z(i)$: this is true if $n = 0$. Assume that at s_n we open a P_e-cycle, the other case being similar: this cycle is closed at the stage s_{n+1}.

By construction, $\sigma^{Y_0}_{s_{n+1}} = \sigma^{Y_0}_{s_n}{}^\frown\tau_0$ and $\sigma^{Y_1}_{s_{n+1}} = \sigma^{Y_1}_{s_n}{}^\frown\tau_1$, where

$$\tau_0 = \langle Z(s_n), \dots, Z(s_{n+1}-1)\rangle \text{ and } \tau_1 = 1^{h\frown}0^k,$$

for some h, k with $h + k = s_{n+1} - s_n$ (here, for $i \in \{0,1\}$, i^m denotes the string of length m giving value i on all its inputs). As in τ_1 the bit 0 appears only in the final segment 0^k, for every $i < s_{n+1} - s_n$ we have that

$$(\tau_0 \vee \tau_1)(i) = Z(s_n + i).$$

It then follows from Lemma 7.2 that for every $j < s_{n+1} - s_n$ we have

$$(Y_0 \vee Y_1)(s_n + j) = (\tau_0 \vee \tau_1)(j) = Z(s_n + j),$$

giving that $(Y_0 \vee Y_1)(i) = Z(i)$ for all $i < s_{n+1}$. □

This concludes the verification. □

By a symmetric argument, one can also show the following.

Theorem 7.6. *Any $R_Z <_{pr} \mathrm{Id}$ is meet-reducible.*

Proof. The requirements are

$$P_e\colon\ p_e \text{ does not reduce } R_{Y_1} \text{ to } R_Z,$$

$$Q_e\colon\ p_e \text{ does not reduce } R_{Y_0} \text{ to } R_Z,$$

$$M\colon\ R_Z = R_{Y_0} \wedge R_{Y_1}.$$

The proof and the construction are similar to the previous theorem, with the modifications that whenever in the previous theorem in a copying or transition phase we added the bit i to Y_0 or Y_1, we now add the bit $1 - i$. Notice for instance that for every e, p_e eventually shows a counterexample to $R_{Y_1} \leqslant_{pr} R_Z$ as otherwise now Y_1 would be eventually finite, thus $R_{Y_1} \equiv_{pr} \mathrm{Id}$, and thus $\mathrm{Id} \leqslant_{pr} R_Z$, a contradiction. So P_e is satisfied. A similar argument shows that each Q_e is satisfied.

In order to show that $R_Z = R_X \wedge R_Y$, notice that this time (assuming that at s_n we open a P_e-cycle, the other case being similar) $\sigma^{Y_0}_{s_{n+1}} = \sigma^{Y_0}_{s_n}{}^\frown\tau_0$ and $\sigma^{Y_1}_{s_{n+1}} = \sigma^{Y_1}_{s_n}{}^\frown\tau_1$ where $\tau_1 = 0^{h\frown}1^k$, for some h, k with $h + k = s_{n+1} - s_n$, and $\tau_0 = \langle Z(s_n), \dots, Z(s_{n+1}-1)\rangle$. It then follows from Lemma 7.2 (as in τ_1 the 0's show up before the 1's) that for every $j < s_{n+1} - s_n$ we have

$$(Y_0 \wedge Y_1)(s_n + j) = (\tau_0 \wedge \tau_1)(j) = Z(s_n + j).$$

Thus by induction on the index n of s_n, we can show that $(Y_0 \wedge Y_1)(i) = Z(i)$ for all $i < s_n$. □

8. Embedding of the diamond lattice

So far, we highlighted that **Peq** is a remarkably well-behaved degree structure, being a dense distributive lattice. Moreover, we proved that degrees below the top are not distinguishable with respect to join- or meet-reducibility. In these two remaining sections, we will turn the perspective upside down, focusing on some fairly unexpected ill-behaviour of **Peq**. In particular, in this section we show that some intervals $R <_{pr} S$ (inside **Peq**) embed the diamond lattice, and some others don't. In fact, we will offer a complete characterization of the intervals which embed the diamond lattice, by relying on a combinatorial property of primitive recursive sets X and Y, named $\blacklozenge$-property, which intuitively says that there are infinitely many initial segments of the natural numbers up to which X and Y have an equivalent number of zeros.

Given a set U, throughout the section we agree, as in the proof of Theorem 6.1, that σ_s^U denotes the initial segment of U having length $s+1$ and $\#\left(0^U\right)[s]$ denotes the cardinality of $\overline{\sigma_s^U}$. To avoid trivial cases, we also assume that here we consider only primitive recursive sets X that are infinite.

Definition 8.1. We say that a pair (X,Y) of primitive recursive sets satisfies *the $\blacklozenge$-property* if there is a pair (X^*,Y^*) of primitive recursive sets such that $R_{X^*} \equiv_{pr} R_X$ and $R_{Y^*} \equiv_{pr} R_Y$ and

$$(\forall s)(\exists t \geqslant s)\left[\#\left(0^{X^*}\right)[t] = \#\left(0^{Y^*}\right)[t]\right].$$

We say that a stage s is an *equilibrium point* for a pair (X,Y) of primitive recursive sets if

$$\#\left(0^X\right)[s] = \#\left(0^Y\right)[s].$$

Theorem 8.2. *An interval $[R,S]$ of* **Peq** *embeds the diamond lattice preserving* 0 *and* 1 *if and only if (R,S) has the $\blacklozenge$-property.*

Proof. By Lemma 3.2, there exist primitive recursive sets such that $R \equiv_{pr} R_X$ and $S \equiv_{pr} R_Z$.

($\Rightarrow$): Suppose that for some primitive recursive sets Y_0 and Y_1, we have that R_X, R_{Y_0}, R_{Y_1}, and R_Z form a diamond. Without loss of generality, one may assume that $0 \in Y_0$. We shall prove that the pair (Y_0,Y_1) has infinitely many equilibrium points.

Suppose that s^* is the last equilibrium point for (Y_0,Y_1). Then, without loss of generality, we may assume that for any $s > s^*$,

$$\#\left(0^{Y_0}\right)[s] > \#\left(0^{Y_1}\right)[s].$$

Let $n^* := \#\left(0^{Y_0}\right)[s^*+1]$. For a number $m \geqslant n^*$, as $p_{\overline{Y_0}}(m) > s^*+1$, we have

$$m+1 = \#\left(0^{Y_0}\right)[p_{\overline{Y_0}}(m)] > \#\left(0^{Y_1}\right)[p_{\overline{Y_0}}(m)],$$

and thus $p_{\overline{Y_0}}(m) < p_{\overline{Y_1}}(m)$. By the definition of $Y_0 \vee Y_1$, we deduce that for every $m \geqslant n^*$, we have $p_{\overline{Y_0 \vee Y_1}}(m) = p_{\overline{Y_0}}(m)$. Therefore, the function

$$f(x) := \begin{cases} 0, & \text{if } x \in Y_0 \vee Y_1, \\ p_{\overline{Y_0}}(l), & \text{if } x = p_{\overline{Y_0 \vee Y_1}}(l) \text{ for some } l < n^*, \\ x, & \text{otherwise,} \end{cases}$$

provides a pr-reduction from $R_{Y_0} \vee R_{Y_1}$ into R_{Y_0}, which gives a contradiction.

Let $s_0 < s_1 < s_2 < \dots$ be the sequence of all equilibrium points for (Y_0, Y_1). We choose an infinite subsequence of equilibrium points

$$s_0^* < s_1^* < s_2^* < \dots$$

such that for every $i \in \omega$, $\#\left(0^{Y_0}\right)[s_i^*] < \#\left(0^{Y_0}\right)[s_{i+1}^*]$.

Let $t_i := \#\left(0^{Y_0}\right)[s_i^*]$. By Lemma 7.2 it is clear that

$$\#\left(0^{Y_0 \vee Y_1}\right)[s_i^*] = \#\left(0^{Y_0 \wedge Y_1}\right)[s_i^*] = t_i,$$

and hence, the pair (X, Z) satisfies the $\blacklozenge$-property.

($\Leftarrow$): On the other hand, we need to show that the $\blacklozenge$-property is sufficient for embedding the diamond. So, suppose that (X, Z) satisfies the $\blacklozenge$-property. We will prove that there are primitive recursive R_{Y_0}, R_{Y_1} such that the infimum (resp. supremum) of R_{Y_0} and R_{Y_1} is pr-equivalent to R_X (resp. R_Z).

Without loss of generality, assume that $0 \in X \cap Z$. Observe also that we may assume that the following hold:

(1) the pair (X, Z) has infinitely many equilibrium points;
(2) for all s, $\#\left(0^Z\right)[s] \geqslant \#\left(0^X\right)[s]$.

To see that this can be assumed, first choose X, Z with infinitely many equilibrium points; they must exist since (X, Z) has the $\blacklozenge$-property. Second, replace Z with $X \vee Z$ if needed; note that if t is an equilibrium point for (X, Z), then by Lemma 7.2 it should be an equilibrium point for $(X, X \vee Z)$ as well.

We will build sets Y_0, Y_1 in stages, satisfying the following requirements:

$$\begin{aligned} Q_e &: \ p_e \text{ does not reduce } R_Z \text{ to } R_{Y_0}, \\ P_e &: \ p_e \text{ does not reduce } R_Z \text{ to } R_{Y_1}, \\ M &: \ R_X = R_{Y_0} \wedge R_{Y_1}, \\ N &: \ R_Z = R_{Y_0} \vee R_{Y_1}. \end{aligned}$$

It is easy to see that the above requirements are sufficient: in particular, we do not need to prove that $R_{Y_i} \not\leqslant_{pr} R_X$. Notice also that we require R_X and R_Z to be in fact equal, and not just pr-equivalent, to $R_{Y_0} \wedge R_{Y_1}$ and $R_{Y_0} \vee R_{Y_1}$, respectively: therefore, we get for free that R_{Y_0} and R_{Y_1} lie in the interval determined by R_X and R_Z. As always, we will build Y_0, Y_1 in stages by approximating their characteristic functions, i.e., $Y_i = \bigcup_{s \in \omega} \sigma_s^{Y_i}$ for $i \in \{0, 1\}$. Each string $\sigma_s^{Y_i}, \sigma_s^X, \sigma_s^Z$ we define or deal with at a stage s has length $s + 1$.

The strategies

In order to achieve that p_e does not reduce R_Z to R_{Y_0} we open the Q_e-cycle and employ a copying procedure, copying R_X into R_{Y_0}, until we see that p_e shows a counterexample to $R_Z \leqslant_{pr} R_{Y_0}$. Meanwhile we employ a corresponding copying procedure, copying R_Z into R_{Y_1}.

When seeing that p_e shows a counterexample at stage, say, $s+1$, by our assumption on always being $\#\left(0^X\right)[t] \leqslant \#\left(0^Z\right)[t]$, it may happen that $\#\left(0^X\right)[s+1] < \#\left(0^Z\right)[s+1]$. If so, before closing the Q_e-cycle we open the so-called $Q_e \to P_e$-transition phase, which consists (still copying R_X into R_{Y_0} and R_Z into R_{Y_1}) in prolonging bit by bit $\sigma_{s+1}^{Y_0}$ (which the construction has guaranteed to have the same number of 0's as σ_{s+1}^X) with the bits of X, and in prolonging bit by bit $\sigma_{s+1}^{Y_1}$ (which the construction has guaranteed to have the same number of 0's as σ_{s+1}^Z) with the bits of Z, until we reach the next equilibrium point of (X, Z): at this point we close the Q_e-cycle and we open the P_e-cycle.

The described procedure has the goal of making it possible to apply Lemma 7.2 and conclude that the bits added to $\sigma^{Y_0}_{s+1}$ and $\sigma^{Y_1}_{s+1}$ since when we opened the Q_e-cycle satisfy

$$(\sigma^{Y_0}_{s+1} \vee \sigma^{Y_1}_{s+1})(i) = (X \vee Z)(i) \text{ and } (\sigma^{Y_0}_{s+1} \wedge \sigma^{Y_1}_{s+1})(i) = (X \wedge Z)(i),$$

so as to eventually get $Y_0 \vee Y_1 = X \vee Z$ and $Y_0 \wedge Y_1 = X \wedge Z$.

In order to achieve that p_e does not reduce R_Z to R_{Y_1}, we use (in an obvious way) a similar strategy, this time copying R_X into R_{Y_1} and R_Z into R_{Y_0}; we go into the $P_e \to Q_{e+1}$-transition phase when p_e shows a counterexample. Finally, after reaching the next equilibrium point, we close the P_e-cycle and open the Q_{e+1}-cycle.

The construction

Unless otherwise specified, we adopt the same terminology and notation employed in Theorem 6.1.

Stage 0

$\sigma^{Y_0}_0 = \sigma^{Y_1}_0 = \langle 1 \rangle$. *Open the Q_0-cycle.*

Stage $s+1$

There are two cases.

CASE 1. Suppose that we are within a previously opened Q_e-cycle, which has not been declared closed. We assume by induction that we have

$$\#\left(0^X\right)[s+1] = \#\left(0^{Y_0}\right)[s+1] \leqslant \#\left(0^{Y_1}\right)[s+1] = \#\left(0^Z\right)[s+1],$$

and the first stage s' of this particular Q_e-cycle has the following property

$$\#\left(0^X\right)[s'] = \#\left(0^{Y_0}\right)[s'] = \#\left(0^{Y_1}\right)[s'] = \#\left(0^Z\right)[s'].$$

Copying phase

If we have not yet moved to the $Q_e \to P_e$-transition phase, then we copy R_X into R_{Y_0} and copy R_Z into R_{Y_1}: let $\sigma^{Y_0}_{s+1} = \sigma^{Y_0}_s{}^\frown\langle X(s+1)\rangle$ and $\sigma^{Y_1}_{s+1} = \sigma^{Y_1}_s{}^\frown\langle Z(s+1)\rangle$. After this, if p_e shows a counterexample to $R_Z \leqslant_{pr} R_{Y_0}$ then go to the *$Q_e \to P_e$-transition phase*, which will be implemented starting from the next stage.

Transition phase

Suppose that we are within the transition phase of a previously opened Q_e-cycle, which has not been declared closed yet. Let $\sigma^{Y_0}_{s+1} = \sigma^{Y_0}_s{}^\frown\langle X(s+1)\rangle$ and $\sigma^{Y_1}_{s+1} = \sigma^{Y_1}_s{}^\frown\langle Z(s+1)\rangle$. After this, if $\#\left(0^{Y_0}\right)[s+1] = \#\left(0^{Y_1}\right)[s+1]$, then *close the Q_e-cycle*, and *open the P_e-cycle*, which will be processed starting from next stage.

CASE 2. Suppose that we are within a previously opened P_e-cycle, which has not been declared closed yet. We assume by induction that

$$\#\left(0^X\right)[s+1] = \#\left(0^{Y_1}\right)[s+1] \leqslant \#\left(0^{Y_0}\right)[s+1] = \#\left(0^Z\right)[s+1],$$

and when we had opened this cycle, we had $\#\left(0^{Y_1}\right)[s'] = \#\left(0^{Y_0}\right)[s']$.

Copying phase

If we have not yet moved to the $P_e \to Q_{e+1}$-transition phase, then let $\sigma^{Y_0}_{s+1} = \sigma^{Y_0}_s{}^\frown\langle Z(s+1)\rangle$ and $\sigma^{Y_1}_{s+1} = \sigma^{Y_1}_s{}^\frown\langle X(s+1)\rangle$. After this, if p_e has shown a counterexample for $p_e\colon R_Z \leqslant_{pr} R_{Y_1}$, then *move to the $P_e \to Q_{e+1}$-transition phase*.

Transition phase

Suppose that we are within the transition phase of a previously opened P_e-cycle, which has not been declared closed yet. Let $\sigma^{Y_0}_{s+1} = \sigma^{Y_0}_s{}^\frown\langle Z(s+1)\rangle$ and $\sigma^{Y_1}_{s+1} = \sigma^{Y_1}_s{}^\frown\langle X(s+1)\rangle$. After this, if $\#\left(0^{Y_0}\right)[s+1] = \#\left(0^{Y_1}\right)[s+1]$, then *close the P_e-cycle*, and *open the Q_{e+1}-cycle*, which will be processed starting from next stage.

The verification

The sets Y_0, Y_1 are primitive recursive as $Y_i(s) = \sigma^{Y_i}_s(s)$ (recall that $l^{Y_i}_s = s+1$) and the mapping $s \mapsto \sigma^{Y_i}_s$ is primitive recursive. The rest of the verification is based on the following lemmas.

Lemma 8.3. *The P- and Q- requirements are satisfied.*

Proof. Similarly to the proofs of Theorem 5.3 and Theorem 6.1, it is easily seen by induction that every P- or Q-cycle is opened and later closed, and exactly one cycle is open at each stage. Consider for instance a Q_e-cycle, and assume that it was opened at stage s_0, and we started processing the cycle from s_0+1. Assume also that

$$\#\left(0^X\right)[s_0] = \#\left(0^{Y_0}\right)[s_0].$$

Should p_e never show a counterexample to $R_Z \leqslant_{pr} R_{Y_0}$, then it would be $R_Z \leqslant_{pr} R_X$. Indeed, in this case we would eventually get $Y_0 = \sigma^{Y_0}_{s_0} * X$ (see Section 2.3 for the notation): thus, $p_e\colon R_Z \leqslant_{pr} R_{Y_0}$ would imply $R_Z \leqslant_{pr} R_X$. Therefore, eventually we do get a counterexample, and requirement Q_e is satisfied.

After p_e shows a counterexample, we start the transition phase: when we open it (say, at s_1), we have

$$\#\left(0^{Y_0}\right)[s_1] \leqslant \#\left(0^{Y_1}\right)[s_1].$$

By our assumption that (X,Z) has the $\blacklozenge$-property, it follows (by prolonging σ^{Y_0} as X, and σ^{Y_1} as Z) that eventually $\#\left(0^{Y_0}\right)$ catches up with $\#\left(0^{Y_1}\right)$, thus we reach a stage $s+1$ when $\#\left(0^{Y_0}\right)[s+1] = \#\left(0^{Y_1}\right)[s+1]$. At this stage, we close the Q_e-cycle and we open the P_e-cycle.

A similar claim holds for P_e-cycles. Note that in the second part of the cycle, the transition phase waits until the inequality $\#\left(0^{Y_1}\right)[t] \leqslant \#\left(0^{Y_0}\right)[t]$ reaches a stage $s+1$ such that $\#\left(0^{Y_1}\right)[s+1] = \#\left(0^{Y_0}\right)[s+1]$.

We also conclude that all P- and Q-requirements are satisfied. □

Lemma 8.4. *The M-requirement and the N-requirement are satisfied.*

Proof. Let $0 = s_0 < s_1 < \ldots$ be an infinite sequence of stages s at which we have

$$\#\left(0^X\right)[s] = \#\left(0^{Y_0}\right)[s] = \#\left(0^{Y_1}\right)[s] = \#\left(0^Z\right)[s].$$

For instance, this happens when we open cycles.

Let $i \in \omega$ and let n be such that $i < s_{n+1} - s_n$. Suppose that at s_n we open a Q-cycle — then $\sigma^{Y_0}_{s_{n+1}} = \sigma^{Y_0}_{s_n}{}^\frown\tau_0$ and $\sigma^{Y_1}_{s_{n+1}} = \sigma^{Y_1}_{s_n}{}^\frown\tau_1$, where $\tau_0(i) = X(s_n+1+i)$ and $\tau_1(i) = Z(s_n+1+i)$. Since the pairs $(\sigma^{Y_0}_{s_n}, \sigma^{Y_1}_{s_n})$, (τ_0,τ_1), and (Y_0,Y_1) satisfy the assumptions of Lemma 7.2, it follows by induction on the index n of s_n that

$$(Y_0 \wedge Y_1)(i) = (X \wedge Z)(i)$$
$$(Y_0 \vee Y_1)(i) = (X \vee Z)(i).$$

Hence, we deduce that $R_{Y_0} \wedge R_{Y_1} = R_X$ and $R_{Y_0} \vee R_{Y_1} = R_Z$. Lemma 8.4 is proved. □

This concludes the verification. Theorem 8.2 is proved. □

9. On the intricacy of Peq

In this final section, we deepen the analysis of **Peq**, unveiling further structural complexity. Most notably, we will focus on the automorphisms of **Peq**, proving that this degree structure is neither rigid nor homogeneous. We will also show that **Peq** contains nonisomorphic lowercones. These results will require both to further explore the consequences of the ♦-property defined above and to introduce another property, named *slowness*, concerning the rate at which a primitive recursive set shows its zeros. We conclude the section by collecting a number of interesting open questions, which may motivate future work. We are particularly interested in whether the theory of **Peq** is decidable or not.

Remark 9.1. (Redefining the symbol $\#\left(0^X\right)[s]$.) In this section, for technical reasons, we will take $\#\left(0^X\right)[s]$ to be the number of $i \leqslant k$ such that $X(i) = 0$, where k is the largest such that $X(j)\downarrow$ in at most s many steps for all $j \leqslant k$. So $\#\left(0^X\right)[s]$ is the number of zeroes that we can see in the characteristic function of X after evaluating it for s many steps.

The next lemma is an analogue of Proposition 5.1 — it expresses $R_X \leqslant_{pr} R_Y$ in terms of the growth rates of $\#\left(0^X\right)$ and $\#\left(0^Y\right)$:

Lemma 9.2. *Given any X, Y, $R_X \leqslant_{pr} R_Y$ if and only if there exists a primitive recursive function p such that for every s, $\#\left(0^X\right)[s] \leqslant \#\left(0^Y\right)[p(s)]$.*

Proof. Suppose that $R_X \leqslant_{pr} R_Y$ via f. For each s we let $p(s)$ be the least stage $t > s$ such that $Y(n)[t]\downarrow$ for all $n \leqslant f(s)$. Then $\#\left(0^X\right)[s] \leqslant \#\left(0^Y\right)[p(s)]$.

Now conversely fix p. For each m we find the first stage t for which we have $X \upharpoonright (m+1)[t]\downarrow$. Let $h(m) = p(t)$. Then $h(p_{\overline{X}}(n)) \geqslant p_{\overline{Y}}(n)$ for all n and by Proposition 5.1, $R_X \leqslant_{pr} R_Y$. □

It is easy to see that inside **Peq**, there are $R <_{pr} S$ such that the pair (R, S) satisfies the ♦-property: By Theorem 4.10 take a pair of incomparable $Y_0 \mid_{pr} Y_1$, then $R = Y_0 \wedge Y_1 <_{pr} S = Y_0 \vee Y_1$ has the ♦-property. In fact, by Theorems 7.3 and 7.6, given any $R <_{pr}$ Id there is some $S >_{pr} R$, and given any S there is some $R <_{pr} S$ such that (R, S) has the ♦-property. So every degree in **Peq** is the top and (if it is not Id) the bottom of an interval with the ♦-property.

However, since the ♦-property is a property of a pr-degree and not of a set, it is not totally obvious why there should be an interval that does *not* satisfy the ♦-property. We prove a lemma which expresses the ♦-property as a property about sets. Note that the ♦-property does not apriori require the two sets to be comparable, and the characterization below holds in general.

Lemma 9.3. *A pair (X, Y) satisfies the ♦-property if and only if there exist primitive recursive functions p and q such that $\#\left(0^X\right)[s] = \#\left(0^Y\right)[p(s)] = \#\left(0^Y\right)[t] = \#\left(0^X\right)[q(t)]$ for infinitely many s, t.*

Proof. Suppose that (X, Y) satisfies the ♦-property. Fix (X^*, Y^*) witnessing that the pair (X, Y) has the ♦-property, so that

$$(\forall s)(\exists t \geqslant s)\left[\#\left(0^{X^*}\right)[t] = \#\left(0^{Y^*}\right)[t]\right],$$

and functions $f_{X,X^*}, f_{X^*,X}, f_{Y,Y^*}$ and $f_{Y^*,Y}$ satisfying

$$\#\left(0^X\right)[s] \leqslant \#\left(0^{X^*}\right)[f_{X,X^*}(s)],\ \ \#\left(0^{X^*}\right)[s] \leqslant \#\left(0^X\right)[f_{X^*,X}(s)],$$

$$\#\left(0^Y\right)[s] \leqslant \#\left(0^{Y^*}\right)[f_{Y,Y^*}(s)],\ \ \#\left(0^{Y^*}\right)[s] \leqslant \#\left(0^Y\right)[f_{Y^*,Y}(s)]$$

for every s, respectively (applying Lemma 9.2). Then obviously we should take p and q to be the composition of the given functions in the correct order. More specifically, we let $p(s) =$ the least stage $t \leqslant f_{Y^*,Y}(f_{X,X^*}(s+1))$ such that $\#\left(0^X\right)[s] = \#\left(0^Y\right)[t]$, and if t cannot be found then let $p(s) = s$. Similarly, let $q(t) =$ the least stage $u \leqslant f_{X^*,X}(f_{Y,Y^*}(t+1))$ such that $\#\left(0^Y\right)[t] = \#\left(0^X\right)[u]$, and if u cannot be found then let $q(t) = t$.

We first check that p works. Let w and j be such that

$$\#\left(0^{X^*}\right)[w] = \#\left(0^{Y^*}\right)[w] = j.$$

Let s be the greatest stage such that $\#\left(0^X\right)[s] = j$. Then as $\#\left(0^{X^*}\right)[w] = j$ and $\#\left(0^X\right)[s+1] = j+1$, we certainly have $w < f_{X,X^*}(s+1)$. Then

$$f_{Y^*,Y}(f_{X,X^*}(s+1)) \geqslant f_{Y^*,Y}(w)$$

and also

$$\#\left(0^Y\right)[f_{Y^*,Y}(w)] \geqslant \#\left(0^{Y^*}\right)[w] = j.$$

Therefore, the bound $f_{Y^*,Y}(f_{X,X^*}(s+1))$ is large enough, and we have

$$\#\left(0^X\right)[s] = \#\left(0^Y\right)[p(s)] = j.$$

A similar argument holds for q.

Now conversely, assume that p and q exist. It is easy to see that we can make p and q nondecreasing. We wish to show that (X,Y) satisfies the ♦-property. An obvious candidate for X^* is a set satisfying $\#\left(0^{X^*}\right)[p(s)] = \#\left(0^X\right)[s]$ for every s, and then we can take $Y^* = Y$, so that $\#\left(0^{X^*}\right)[p(s)] = \#\left(0^{Y^*}\right)[p(s)]$ holds for infinitely many s. Unfortunately, in order to do this, we will need to compute p^{-1} which in general is not primitive recursive. So we will have to use both p and q to define X^* and Y^*.

We call (s,t) a *good pair* if

$$\#\left(0^X\right)[s] = \#\left(0^Y\right)[p(s)] = \#\left(0^Y\right)[t] = \#\left(0^X\right)[q(t)];$$

by the hypothesis there are infinitely many good pairs.

First, suppose that there are infinitely many good pairs (s,t) such that $p(s) \geqslant s$. Define $c(w)$ to be the largest value of $u \leqslant w$ such that $\#\left(0^X\right)[u] \leqslant \#\left(0^Y\right)[w]$ or $p(u) < w$. Notice that the function c is primitive recursive and non-decreasing. Therefore, so is the function $d(w) = \min\left\{d(w-1)+1, \#\left(0^X\right)[c(w)]\right\}$. Furthermore, d has the property that for any w, $d(w+1) \leqslant d(w)+1$, and that for any w there is some t satisfying $w \leqslant t \leqslant 2w$ such that $d(t) = \#\left(0^X\right)[c(w)]$. (Recall our convention that for any set Z and any stage t, $\#\left(0^Z\right)[t+1] \leqslant \#\left(0^Z\right)[t]+1$). Therefore we can define the primitive recursive set X^* satisfying $\#\left(0^{X^*}\right)[w] = d(w)$ for all w. Take $Y^* = Y$.

Let $\hat{d}(w)$ be the largest value $\leqslant w$ such that $\#\left(0^X\right)[\hat{d}(w)] = d(w)$, which is also primitive recursive. Therefore, by Lemma 9.2 we obviously have $R_{X^*} \leqslant_{pr} R_X$. Now we observe that for each s, $s \leqslant c(w)$ where $w = \max\{s, p(s)+1\}$. Therefore

$$\#\left(0^{X^*}\right)[2w] = d(2w) \geqslant \#\left(0^X\right)[c(w)] \geqslant \#\left(0^X\right)[s],$$

showing that $R_{X^*} \geqslant_{pr} R_X$. Now it follows by a straightforward induction on w that the following claim is true:

$$d(w) \geqslant \max\left\{\#\left(0^X\right)[u] \mid \#\left(0^X\right)[u] \leqslant \#\left(0^Y\right)[w] \text{ for some } u \leqslant w\right\}$$

(using the fact that

$$\#\left(0^X\right)[c(w)] \geqslant \max\left\{\#\left(0^X\right)[u] \mid \#\left(0^X\right)[u] \leqslant \#\left(0^Y\right)[w] \text{ for some } u \leqslant w\right\}).$$

Now take (s,t) to be a good pair with $p(s) \geqslant s$. By the claim above, we have $\#\left(0^{X^*}\right)[p(s)] \geqslant \#\left(0^X\right)[s]$. If they were not equal, then $d(p(s))$ would have to be larger than $\#\left(0^X\right)[s]$ which means that

$$\#\left(0^X\right)[s] < d(p(s)) \leqslant \#\left(0^X\right)[c(p(s))].$$

But then as $c(p(s)) \geqslant s$ and p is nondecreasing, we have $p(c(p(s))) \geqslant p(s)$, which means, by the definition of $c(p(s))$, that

$$\#\left(0^X\right)[c(p(s))] \leqslant \#\left(0^Y\right)[p(s)] = \#\left(0^X\right)[s],$$

a contradiction. Thus we conclude that

$$\#\left(0^{X^*}\right)[p(s)] = \#\left(0^X\right)[s] = \#\left(0^Y\right)[p(s)] = \#\left(0^{Y^*}\right)[p(s)].$$

If there are infinitely many good pairs (s,t) such that $q(t) \geqslant t$ then we repeat the above, now taking $X^* = X$ and Y^* defined analogously, using q in place of p. So we assume that there are infinitely many good pairs (s,t) such that $p(s) < s$ and $q(t) < t$. We claim that we can take $X = X^*$ and $Y = Y^*$. Fix a good pair (s,t) such that $p(s) < s$, $q(t) < t$, and

$$\#\left(0^X\right)[s] = \#\left(0^Y\right)[p(s)] = \#\left(0^Y\right)[t] = \#\left(0^X\right)[q(t)] = j.$$

Suppose that

$$\min\{w \mid \#\left(0^X\right)[w] = j\} \leqslant \min\{w \mid \#\left(0^Y\right)[w] = j\}.$$

Now this means that

$$p(s) \geqslant \min\{w \mid \#\left(0^X\right)[w] = j\}$$

and since $p(s) < s$ we have that

$$\#\left(0^X\right)[p(s)] = j = \#\left(0^Y\right)[p(s)].$$

On the other hand, if

$$\min\{w \mid \#\left(0^X\right)[w] = j\} \geqslant \min\{w \mid \#\left(0^Y\right)[w] = j\},$$

then

$$\#\left(0^Y\right)[q(t)] = j = \#\left(0^X\right)[q(t)]. \qquad \square$$

Corollary 9.4. *Let $R_X \leqslant_{pr} R_Y$. Then (X,Y) satisfies the $\blacklozenge$-property if and only if there exists a primitive recursive function q such that $\#\left(0^Y\right)[t] = \#\left(0^X\right)[q(t)]$ for infinitely many t. Furthermore, we can take q to be nondecreasing, and we may also replace "$\#\left(0^Y\right)[t] = \#\left(0^X\right)[q(t)]$" with "$\#\left(0^Y\right)[t] \leqslant \#\left(0^X\right)[q(t)]$".*

Proof. If $R_X \leqslant_{pr} R_Y$, then we fix by Lemma 9.2, a function g such that $\#\left(0^X\right)[s] \leqslant \#\left(0^Y\right)[g(s)]$ for every s. But $p(s) \leqslant g(s)$ for every s, where $p(s)$ is the least such that $\#\left(0^X\right)[s] = \#\left(0^Y\right)[p(s)]$. □

Corollary 9.5. *If $R_X \leqslant_{pr} R_Y$ and (X, Y) has the ♦-property, then every subinterval of (X, Y) also has the ♦-property.*

Proof. Suppose $R_X \leqslant_{pr} R_{X'} \leqslant_{pr} R_{Y'} \leqslant_{pr} R_Y$. Restricting the diamond R_C, R_D to the subinterval (X', Y') does not automatically do it, since for instance, $R_C \vee R_{X'}$ could be above $R_{Y'}$.

We may assume that $\#\left(0^X\right)[t] = \#\left(0^Y\right)[t]$ for infinitely many t. We fix functions f and g such that for every t, $f(t)$ and $g(t)$ are the least such that $\#\left(0^{Y'}\right)[t] = \#\left(0^Y\right)[g(t)]$ and $\#\left(0^X\right)[t] = \#\left(0^{X'}\right)[f(t)]$. Now given any t let $q(t) = f(u)$ where u is the largest such that $u < g(t+1)$ and $\#\left(0^X\right)[u] = \#\left(0^Y\right)[u]$. If u cannot be found, let $q(t) = t$. By Corollary 9.4 it remains to check that $\#\left(0^{Y'}\right)[w] = \#\left(0^{X'}\right)[q(w)]$ for infinitely many w. Suppose that $\#\left(0^X\right)[t] = \#\left(0^Y\right)[t] = j$ for some t, j. Let w be the largest such that $\#\left(0^{Y'}\right)[w] = j$. Then

$$\#\left(0^Y\right)[g(w+1)] = \#\left(0^{Y'}\right)[w+1] = j+1$$

and therefore $t < g(w+1)$. By the minimality of $g(w+1)$, we have

$$\#\left(0^X\right)[u] = \#\left(0^Y\right)[u] = j$$

for the chosen u, and therefore

$$\#\left(0^{X'}\right)[q(w)] = \#\left(0^{X'}\right)[f(u)] = \#\left(0^X\right)[u] = j = \#\left(0^{Y'}\right)[w]. \qquad \square$$

Lemma 9.3 characterizes the ♦-property in terms of the relative growth rates of $\#\left(0^X\right)$ and $\#\left(0^Y\right)$. For our next purpose it shall be convenient to express the ♦-property in terms of the relative growth rates of $p_{\overline{X}}$ and $p_{\overline{Y}}$. The term "$n+1$" in the next lemma is important; by Remark 9.14 we cannot replace $p_{\overline{Y}}(n+1)$ with $p_{\overline{Y}}(n)$.

Lemma 9.6. *Let $R_X \leqslant_{pr} R_Y$. Then (X, Y) satisfies the ♦-property if and only if there exists a primitive recursive function r such that $p_{\overline{X}}(n) \leqslant r(p_{\overline{Y}}(n+1))$ for infinitely many n.*

Proof. Suppose that (X, Y) has the ♦-property. Fix q as in Corollary 9.4. Let $r(m) = q(u)$, where u is the least stage such that $Y(i)[u]\downarrow$ for all $i \leqslant m$. Now let t and n be such that $\#\left(0^Y\right)[t] = \#\left(0^X\right)[q(t)] = n$. Notice that $p_{\overline{X}}(n) < q(t)$. Let $m = p_{\overline{Y}}(n+1)$ and u be such that $r(m) = q(u)$. Then since $\#\left(0^Y\right)[t] = n$, we have $t < u$, which means that $r(m) = q(u) \geqslant q(t) > p_{\overline{X}}(n)$.

Now suppose that r exists; obviously we may assume that r is nondecreasing. Define $q(t)$ to be the largest stage $u \leqslant v$ such that $\#\left(0^X\right)[u] = \#\left(0^Y\right)[t]$, where v is the least stage such that $X(j)[v]\downarrow$ for all $j \leqslant r(t+1)$; if this does not exist, let $q(t) = t$. Now let n be such that $p_{\overline{X}}(n) \leqslant r(p_{\overline{Y}}(n+1))$, and let t be the largest such that $\#\left(0^Y\right)[t] = n$. We check that $\#\left(0^Y\right)[t] = \#\left(0^X\right)[q(t)]$. We have $\#\left(0^Y\right)[t+1] = n+1$ and thus $p_{\overline{Y}}(n+1) < t+1$ which means that

$$r(t+1) \geqslant r(p_{\overline{Y}}(n+1)) \geqslant p_{\overline{X}}(n).$$

This means that $q(t)$ will be equal to some largest u such that

$$\#\left(0^X\right)[u] = \#\left(0^Y\right)[t] = n.$$

Hence $\#\left(0^X\right)[q(t)] = \#\left(0^Y\right)[t]$. □

Returning to our question as to whether every interval has the ♦-property, we can now make use of Lemma 9.6 to construct an interval that does not have the ♦-property. In fact, we can show that every degree in **Peq** is the top of an interval that does not satisfy the ♦-property:

Proposition 9.7. *Given any infinitary R_Y, there is some $R_X <_{pr} R_Y$ such that (X, Y) does not have the ♦-property.*

Proof. We first note that given any total computable function F there is a co-infinite primitive recursive set X such that $p_{\overline{X}}(n) \geqslant F(n)$ for every n. Now given any Y we let F to be a computable function that is fast growing enough so that for every primitive recursive function r, $F(n) > r(p_{\overline{Y}}(n+1))$ for almost all n. (Notice that $p_{\overline{Y}}$ is not necessarily primitive recursive). Now we take X such that $p_{\overline{X}}(n) \geqslant F(n)$ for all n. Then $R_X \leqslant_{pr} R_Y$ by Proposition 5.1 and (X, Y) does not have the ♦-property by Lemma 9.6. This of course implies that $Y \nleqslant_{pr} X$. □

On the other hand, it is not the case that every pr-degree is the bottom of an interval that does not have the ♦-property. For instance, if (X, T) has the ♦-property then by Corollary 9.5 there is no Y such that $R_X \leqslant_{pr} R_Y$ and (X, Y) does not have the ♦-property.

By Theorem 8.2, the ♦-property is definable in the language of partial orders. This property will be crucial in our subsequent analysis of **Peq**. The next most natural step when studying a new degree structure is to verify whether the degree structure is rigid or homogeneous. We introduce an important definition that shall soon prove to be very useful:

Definition 9.8. Given any primitive recursive set X, we define $X^{[-1]}$ to be the set defined by:

$$X^{[-1]}(n) = \begin{cases} 1, & \text{if } n \text{ is the least such that } X(n) = 0, \\ X(n), & \text{otherwise.} \end{cases}$$

In particular, $\#\left(0^{X^{[-1]}}\right)[t] = \max\{0, \#\left(0^{X}\right)[t] - 1\}$ for every stage t.

An immediate consequence of the definition is:

Lemma 9.9. $R_{X^{[-1]}} <_{pr} R_X$ *if and only if* $R_X <_{pr} \mathrm{Id}$.

Proof. Since $\#\left(0^{X^{[-1]}}\right)[t] \leqslant \#\left(0^{X}\right)[t]$ for every t, we have $R_{X^{[-1]}} \leqslant_{pr} R_X$ (by Lemma 9.2), so we have to show that $R_{X^{[-1]}} \geqslant_{pr} R_X$ if and only if $R_X \geqslant_{pr} \mathrm{Id}$. Suppose that g reduces Id to R_X. Let n be the least element not in X. If $n \notin \mathrm{range}(g)$ then g is already a reduction from Id to $R_{X^{[-1]}}$, and if $g(m) = n$ then the function $h(k) = g(k+m+1)$ will reduce Id to $R_{X^{[-1]}}$.

Conversely suppose that f reduces R_X to $R_{X^{[-1]}}$. Define the function F by the following. Let $F(0) = \max f([0, n])$ where n is the second element not in X. Let $F(k+1) = \max f([0, F(k)])$. Then for each k, there are at least $k+1$ many distinct elements not in X which are smaller than $F(k)$. The function F can easily be used to define a reduction of Id to R_X. □

Our first question about rigidity is easily answered by Lemma 9.9:

Theorem 9.10. $(\mathbf{Peq}, \leqslant)$ *is not rigid.*

Proof. The map $\deg(R_X) \mapsto \deg\left(R_{X^{[-1]}}\right)$ is a non-trivial automorphism. In fact, it fixes $\deg(\mathrm{Id})$ and moves every other degree to a strictly smaller degree. □

Corollary 9.11. *The only definable degree is the greatest degree,* $\deg(\mathrm{Id})$. *No finite set of degrees is definable except for* $\{\deg(\mathrm{Id})\}$.

We turn now our attention to the question of how homogeneous the structure $(\mathbf{Peq}, \leqslant)$ is. (Un)fortunately, the structure $(\mathbf{Peq}, \leqslant)$ is neither rigid nor homogeneous, which indicates that the structure is not as trivial as might seem at first glance. This justifies further investigations into the degree structure of **Peq**.

Definition 9.12. We call a co-infinite primitive recursive set X *slow* if for every primitive recursive function r, $p_{\overline{X}}(n+1) > r(p_{\overline{X}}(n))$ holds for almost every n.

A slow set generates its zeros slower than any primitive recursive recursive function can predict (infinitely often). Slow sets obviously exist. An immediate consequence of Lemma 9.6 is:

Corollary 9.13. *If X is slow then (X, Id) does not satisfy*[1] *the ⧫-property.*

Thus, if X is slow and (Y, Id) satisfies the ⧫-property (Y exists, by Theorem 6.1), then no automorphism of $(\mathbf{Peq}, \leqslant)$ can map $\deg(Y)$ to $\deg(X)$. This fact also means that uppercones of **Peq** are not always isomorphic to each other.

We also note that the converse to Corollary 9.13 is false: Given any X we can easily find some Y such that $R_{X^{[-1]}} \leqslant_{pr} R_Y \leqslant_{pr} R_X$ and Y is not slow; to do this we can arrange $p_{\overline{Y}}(n+1) = p_{\overline{Y}}(n) + 1$ for infinitely many n. Then for each such Y, (Y, Id) cannot satisfy the ⧫-property, by Corollary 9.5.

When we turn to lowercones however, the situation is less obvious. Since every degree in **Peq** is the top of an interval with the ⧫-property as well as the top of (another) interval without the ⧫-property, it is not clear how we can immediately distinguish two lowercones from each other using the ⧫-property, similarly to how we separated uppercones. In fact, the lowercone $\{\deg(R_Y) \mid R_Y \leqslant_{pr} R_X\}$ is isomorphic to the lower cone $\{\deg(R_Y) \mid R_Y \leqslant_{pr} R_{X^{[-1]}}\}$.

Hence it is entirely conceivable that every lowercone $\{\deg(R_Y) \mid R_Y \leqslant_{pr} R_X\}$ is isomorphic to $(\mathbf{Peq}, \leqslant)$. From the point of view of each degree of R_Y where $R_Y \leqslant_{pr} R_X$, the set X has no delay, since the zeros of X are always generated no slower than the zeros of Y. Hence we might expect to always be able to extend any partial embedding of **Peq** into $\{\deg(R_Y) \mid R_Y \leqslant_{pr} R_X\}$. We will show that this is not the case. The key to our analysis lies (again!) in the operator $X \mapsto X^{[-1]}$.

Remark 9.14. By Lemma 9.6, $(X^{[-1]}, X)$ will have the ⧫-property for any X. Consequently, we cannot replace "$p_{\overline{Y}}(n+1)$" in Lemma 9.6 with "$p_{\overline{Y}}(n)$"; for instance, if Y is slow and $X = Y^{[-1]}$.

Even though an interval of the form $(X^{[-1]}, X)$ will always satisfy the ⧫-property, the same is not true of an interval of the form $(X^{[-2]}, X)$, where $X^{[-2]} = \left(X^{[-1]}\right)^{[-1]}$.

Lemma 9.15. *$(X^{[-2]}, X)$ satisfies the ⧫-property if and only if X is not slow.*

Proof. We apply Lemma 9.6 and noting that $p_{\overline{X^{[-2]}}}(n) = p_{\overline{X}}(n+2)$. □

Lemma 9.16. *Given any $R_Y \leqslant_{pr} R_X$, either (Y, X) satisfies the ⧫-property or $R_Y \leqslant_{pr} R_{X^{[-1]}}$.*

Proof. If $R_Y \nleqslant_{pr} R_{X^{[-1]}}$ then $\#\left(0^Y\right)[s] > \#\left(0^{X^{[-1]}}\right)[s]$ for infinitely many s, which means that $\#\left(0^Y\right)[s] \geqslant \#\left(0^X\right)[s]$ for infinitely many s. Apply Corollary 9.4. □

Lemma 9.16 tells us that $R_{X^{[-1]}}$ bounds all R_Y below R_X such that (Y, X) does not have the ⧫-property. This will allow us to define the map $\deg(R_X) \mapsto \deg(R_{X^{[-1]}})$. Towards this, we prove another lemma:

Lemma 9.17. *Let R_X and R_Y be primitive recursive equivalence relations with $R_{X^{[-1]}} \nleqslant_{pr} R_Y$. Then there is some Z such that $R_Z \leqslant_{pr} R_{X^{[-1]}}$, (Z, X) does not have the ⧫-property and $R_Z \nleqslant_{pr} R_Y$.*

[1]Strictly speaking, we should write $(X, 2\omega)$ instead of (X, Id) here.

Proof. Fixing a computable listing of all primitive recursive functions (as in Section 2.2), it is not hard to construct a collection $\{p_e\}_{e\in\omega}$ of primitive recursive functions so that every p_e is strictly increasing and $p_{e+1}(n) > p_e(n)$. This listing $(e,n) \mapsto p_e(n)$ is total computable but of course not primitive recursive. We will also assume that $p_e(x) = \psi(x)$ for some total computable function which halts in fewer than $\hat{p}_e(x)$ many steps, where $\hat{p}_e$ is some primitive recursive function. All indices can be found effectively.

We now define the set Z in stages. Since Z must be primitive recursive, at every stage s we must decide $Z \upharpoonright s+1$. In the below construction, at each stage $s+1$, we will declare $\#\left(0^Z\right)[s+1] = \#\left(0^Z\right)[s]$ or $\#\left(0^Z\right)[s]+1$; in the former case we mean that we set $Z(s+1) = 1$ and in the latter case we set $Z(s+1) = 0$.

At stage s we will have a parameter $V(s)$ which stands for the index such that at stage s we are attempting to make $\#\left(0^Z\right)[s] \nleq \#\left(0^Y\right)[p_{V(s)}(s)]$. At stage $s = 0$ we declare $0 \in Z$ (i.e. $\#\left(0^Z\right)[0] = 0$) and set $V(0) = 0$. Suppose we have the value of $\#\left(0^Z\right)[s]$ and $V(s) = e$. Compute $p_e(k)$ for one more step (where k was the input that was last processed). If there is a new convergence $p_e(k)$ seen at this step such that $\#\left(0^X\right)[k] \neq \#\left(0^X\right)[k+1]$, we take

$$\#\left(0^Z\right)[s+1] = \min\{\#\left(0^Z\right)[s]+1, \#\left(0^X\right)[s]-1\}.$$

Check if $\#\left(0^Z\right)[t] > \#\left(0^Y\right)[p_e(t)]$ for any $t \leqslant s$ for which we have already found the value of $p_e(t)$. If so we increase V by one. In all other cases take $\#\left(0^Z\right)[s+1] = \#\left(0^Z\right)[s]$, and go to the next stage with the same value of V.

The above gives a primitive recursive description of the set Z. Since $\#\left(0^Z\right)[s] < \#\left(0^X\right)[s]$ for every s, we have $R_Z \leqslant_{pr} R_{X^{[-1]}}$. Notice that the construction processes the inputs k sequentially; namely, the construction begins with $k = 0$ and $V = 0$ and waits for $p_0(0)$ to converge (this takes $\hat{p}_0(0)$ many stages). It then moves on to $k = 1$ and waits for $p_0(1)$ to converge, and so on. If ever, the construction decides to increase the value of V while waiting on, say, $p_0(5)$, then we will move on to wait for $p_1(5)$ to converge, then $p_1(6)$, and so on. Let $k^*(s)$ be the value of k being processed by the construction at stage s. Since $\{p_e\}_{e\in\omega}$ are all total, $\lim_s k^*(s) = \infty$. Define the sequence $\{k_i\}_{i\in\omega}$ such that $\#\left(0^{X^{[-1]}}\right)[k_i] = i$ and $\#\left(0^{X^{[-1]}}\right)[k_i+1] = i+1$. Take $k_{-1} = -1$.

Claim 9.18. *For every stage s we have $\#\left(0^Z\right)[s] = i$, where $k_{i-1} < k^*(s) \leqslant k_i$.*

Proof. If $s = 0$, then $i = k^*(0) = 0$ and so $\#\left(0^Z\right)[0] = 0$. Assume $\#\left(0^Z\right)[s] = i$ where $k_{i-1} < k^*(s) \leqslant k_i$. Since the value of $\#\left(0^Z\right)[s+1]$ is decided at the end of stage s, we have to examine what the construction did at stage s. At stage s we would increase the value of $\#\left(0^Z\right)$ only if $p_{V(s)}(k^*(s))$ is found to converge at that stage and $k^*(s) = k_i$. In that case $k^*(s+1) = k_i+1$ and so $k_i < k^*(s+1) \leqslant k_{i+1}$, and so we have to check that $\#\left(0^Z\right)[s+1] = i+1$. But note that as $k^*(s) < s$ we have

$$\#\left(0^{X^{[-1]}}\right)[s] \geqslant \#\left(0^{X^{[-1]}}\right)[k^*(s+1)] = i+1 = \#\left(0^Z\right)[s]+1,$$

and so

$$\#\left(0^Z\right)[s+1] = \min\{\#\left(0^Z\right)[s]+1, \#\left(0^X\right)[s]-1\} = i+1. \qquad \square$$

Next, we verify that $\lim_s V(s) = \infty$; suppose not. Let t_0 be the least such that $V(t_0) = e$ and $V(s) = e$ for almost all s. Let $t_0 < t_1 < t_2 < \cdots$ be the stages such that $p_e(l_i)$ first converged at stage t_i, where $i > 0$, $k^*(t_i) = l_i < k^*(t_i+1)$ and $l_i = k_{i+t_0}$. Note that $l_1 > t_0$. By our convention above, we have that $t_i = \hat{p}_e(l_i)$. By Claim 9.18 we see that $\#\left(0^Z\right)[t] = \#\left(0^Z\right)[t_{i+1}]$ for every t and i such that $t_i < t \leqslant t_{i+1}$.

For every k and $i > 0$ such that $l_i < k \leqslant l_{i+1}$, we have $t_i+1 = \hat{p}_e(l_i)+1 \leqslant \hat{p}_e(l_i+1) \leqslant \hat{p}_e(k) \leqslant \hat{p}_e(l_{i+1}) = t_{i+1}$, and since $\#\left(0^Z\right)[t_i+1] = \#\left(0^Z\right)[t_{i+1}]$, we also have $\#\left(0^Z\right)[\hat{p}_e(k)] = \#\left(0^Z\right)[t_{i+1}]$. Therefore, we have

$$\#\left(0^{X^{[-1]}}\right)[k] \leqslant \#\left(0^{X^{[-1]}}\right)[l_{i+1}] \quad \text{(by Claim 9.18)}$$

$$\begin{aligned} &= \#\left(0^Z\right)[t_{i+1}] \\ &= \#\left(0^Z\right)[\hat{p}_e(k)] \quad \text{(as the construction never increased } V \text{ after } t_0\text{)} \\ &\leqslant \#\left(0^Y\right)[p_e(\hat{p}_e(k))]. \end{aligned}$$

This shows that $R_{X^{[-1]}} \leqslant_{pr} R_Y$, contrary to our assumption.

Now that we know $\lim_s V(s) = \infty$, for every e there must be a stage s of the construction where we saw $\#\left(0^Z\right)[t] > \#\left(0^Y\right)[p_e(t)]$ for some $t \leqslant s$, which means that $R_Z \nleqslant_{pr} R_Y$. Now consider some primitive recursive function p_e. Let $V(s) = e$ for some s. For each $k > k^*(s)$ we let $t > s$ be the stage where $k^*(t) = k < k^*(t+1)$, and i be such that $k_{i-1} < k \leqslant k_i$. By Claim 9.18, $\#\left(0^Z\right)[t] = i$. We now have

$$\begin{aligned} \#\left(0^Z\right)[p_e(k)] &\leqslant \#\left(0^Z\right)[\hat{p}_e(k)] \\ &\leqslant \#\left(0^Z\right)[\hat{p}_{V(t)}(k)] && \text{(at stage } t \text{ we saw } p_{V(t)}(k) \text{ converge)} \\ &= \#\left(0^Z\right)[t] \\ &= i \\ &= \#\left(0^{X^{[-1]}}\right)[k_i] \\ &= \#\left(0^{X^{[-1]}}\right)[k] \\ &< \#\left(0^X\right)[k]. \end{aligned}$$

Thus, (Z, X) does not have the ♦-property. □

Now we are ready to apply the analysis started above.

Corollary 9.19. *The map* $\deg(R_X) \mapsto \deg(R_{X^{[-1]}})$ *is definable in* $(\mathbf{Peq}, \leqslant)$.

Proof. Apply Lemmas 9.16 and 9.17 to see the following. Given primitive recursive R_X and R_Y, we have $R_Y \equiv_{pr} R_{X^{[-1]}}$ if and only if the following hold:

(1) $R_Y \leqslant_{pr} R_X$,
(2) For every $R_Z \leqslant_{pr} R_X$ such that (Z, X) does not have the ♦-property, $R_Z \leqslant_{pr} R_Y$, and
(3) If R_V has properties (1) and (2), then $R_V \geqslant_{pr} R_Y$.

That is, we may define $R_{X^{[-1]}}$ as the least degree below R_X that is an upper bound for the set of all degrees $R_Z \leqslant_{pr} R_X$ such that (Z, X) does not have the ♦-property. Since the ♦-property is definable, so is the set of all pairs $(\deg(R_X), \deg(R_{X^{[-1]}}))$. □

Corollary 9.20. *If ψ is an automorphism of* $(\mathbf{Peq}, \leqslant)$*, then for any R_X, we have that* $\psi(\deg(R_{X^{[-1]}})) = \deg(R_{Y^{[-1]}})$*, where* $R_Y \equiv_{pr} \psi(R_X)$.

Corollary 9.21. *If ψ is an automorphism of* $(\mathbf{Peq}, \leqslant)$*, then R_X is slow if and only if $\psi(R_X)$ is slow.*

Proof. By Lemma 9.15, R_X is slow if and only if $(X^{[-2]}, X)$ does not have the ♦-property, if and only if $\left(\psi(X^{[-2]}), \psi(X)\right)^2$ does not have the ♦-property. But then $\psi(X^{[-2]}) = \psi(X)^{[-2]}$ which is equivalent to the fact that $\psi(X)$ is slow. □

[2]The notation is not formally correct, but has the obvious meaning here.

We are now able to give a negative answer to our question above regarding whether every lowercone is isomorphic to **Peq**. In fact, *no* proper lowercone can be isomorphic to **Peq**. Even though each lowercone is principal, our intuition that we should be able to replicate the structure below Id in any lowercone by "relativising" is entirely incorrect.

Corollary 9.22. *No proper lowercone can be isomorphic to* (**Peq**, $\leqslant$).

Proof. If $R_X <_{pr}$ Id then by Lemma 9.9, $R_{X^{[-1]}} <_{pr} R_X$. Thus, by Lemma 9.16, $\deg(R_{X^{[-1]}})$ is a degree strictly below $\deg(R_X)$ that is an upperbound on the set of all degrees $R_Z \leqslant_{pr} R_X$ such that (Z, X) does not have the ⧫-property (and thus does not embed the diamond). By Lemma 9.17, (applied with $R_X = R_{X^{[-1]}} = \mathrm{Id}$), no degree strictly below $\deg(\mathrm{Id})$ can serve as the image of $\deg(R_{X^{[-1]}})$. □

Our analysis on lowercones exploits the unique property satisfied by $X^{[-1]}$, and thus can only be applied to separate an incomplete (or proper) lowercone from **Peq**. Using our analysis, we can still say a little more; we show that not every pair of proper lowercones are isomorphic:

Corollary 9.23. *If X is slow and Y is not, then their lowercones cannot be isomorphic (as posets).*

Proof. Because $X^{[-1]}$ (and similarly $Y^{[-1]}$) is the least upperbound of the set of all $R_Z \leqslant_{pr} R_X$ such that (Z, X) does not embed the diamond, any isomorphism between the two lowercones must send $X^{[-1]}$ to $Y^{[-1]}$ and therefore must map $X^{[-2]}$ to $Y^{[-2]}$. However, as X is slow and Y is not, by Lemma 9.15, $(Y^{[-2]}, Y)$ satisfies the ⧫-property whereas $(X^{[-2]}, X)$ does not. □

Since it is not hard to construct a pair of incomparable degrees, one of which is slow and the other is not, we immediately have a pair of incomparable lowercones that are not isomorphic. This leaves the intriguing question as to whether any pair of incomparable lowercones are isomorphic. We leave this question open:

Question 9.24. Are there incomparable degrees $R_X \mid_{pr} R_Y$ with isomorphic lowercones?

Question 9.25. Are there continuum many automorphisms of (**Peq**, $\leqslant$)?

Given that (Corollary 9.11) no finite set of degrees except for $\{\deg(\mathrm{Id})\}$ is definable (without parameters), it may be difficult to apply the analysis of [13, 15] to encode finite graphs into **Peq**, which would involve having to define finite sets of degrees, albeit with a parameter. So we also ask:

Question 9.26. Is the first order theory of (**Peq**, $\leqslant$) decidable?

Acknowledgements

Bazhenov was supported by the Mathematical Center in Akademgorodok under agreement No. 075-15-2019-1613 with the Ministry of Science and Higher Education of the Russian Federation. Ng was supported by MOE Tier 1 grant RG23/19. San Mauro was supported by the Austrian Science Fund FWF, Project M 2461. Sorbi is a member of INDAM-GNSAGA, and was partially supported by PRIN 2017 Grant "Mathematical Logic: models, sets, computability".

References

[1] H. Friedman and L. Stanley, A Borel reducibility theory for classes of countable structures, *J. Symb. Logic* **54**(3) (1989), 894–914. doi:10.2307/2274750.
[2] L.A. Harrington, A.S. Kechris and A. Louveau, A Glimm-Effros dichotomy for Borel equivalence relations, *J. Amer. Math. Soc.* **3**(4) (1990), 903–928.

[3] S. Gao, *Invariant descriptive set theory*, CRC Press, Boca Raton, 2008.

[4] G. Hjorth, Borel equivalence relations, in: *Handbook of set theory*, Springer, Dordrecht, 2010, pp. 297–332. doi:10.1007/978-1-4020-5764-9_5.

[5] Y.L. Ershov, *Theory of numberings*, Nauka, Moscow, 1977, in Russian.

[6] Y.L. Ershov, Theory of numberings, in: *Handbook of Computability Theory*, E.G. Griffor, ed., Stud. Logic Found. Math., Vol. 140, North-Holland, Amsterdam, 1999, pp. 473–503. doi:10.1016/S0049-237X(99)80030-5.

[7] S. Gao and P. Gerdes, Computably enumerable equivalence relations, *Stud. Log.* **67**(1) (2001), 27–59. doi:10.1023/A:1010521410739.

[8] U. Andrews, S. Lempp, J.S. Miller, K.M. Ng, L. San Mauro and A. Sorbi, Universal computably enumerable equivalence relations, *J. Symb. Logic* **79**(1) (2014), 60–88. doi:10.1017/jsl.2013.8.

[9] S. Coskey, J.D. Hamkins and R. Miller, The hierarchy of equivalence relations on the natural numbers under computable reducibility, *Computability* **1**(1) (2012), 15–38. doi:10.3233/COM-2012-004.

[10] C. Bernardi and A. Sorbi, Classifying positive equivalence relations, *J. Symb. Logic* **48**(3) (1983), 529–538. doi:10.2307/2273443.

[11] E. Ianovski, R. Miller, K.M. Ng and A. Nies, Complexity of equivalence relations and preorders from computability theory, *J. Symb. Logic* **79**(3) (2014), 859–881. doi:10.1017/jsl.2013.33.

[12] E.B. Fokina, S.-D. Friedman, V. Harizanov, J.F. Knight, C. McCoy and A. Montalbán, Isomorphism relations on computable structures, *J. Symb. Logic* **77**(1) (2012), 122–132. doi:10.2178/jsl/1327068695.

[13] U. Andrews and A. Sorbi, Joins and meets in the structure of ceers, *Computability* **8**(3–4) (2019), 193–241. doi:10.3233/COM-180098.

[14] U. Andrews, S. Badaev and A. Sorbi, A survey on universal computably enumerable equivalence relations, in: *Computability and Complexity*, A. Day, M. Fellows, N. Greenberg, B. Khoussainov, A. Melnikov and F. Rosamond, eds, Lect. Notes Comput. Sci., Vol. 10010, Springer, Cham, 2017, pp. 418–451. doi:10.1007/978-3-319-50062-1_25.

[15] U. Andrews, N. Schweber and A. Sorbi, The theory of ceers computes true arithmetic, *Ann. Pure Appl. Logic* **171**(8) (2020), 102811. doi:10.1016/j.apal.2020.102811.

[16] K.M. Ng and H. Yu, On the degree structure of equivalence relations under computable reducibility, *Notre Dame J. Form. Log.* **60**(4) (2019), 733–761. doi:10.1215/00294527-2019-0028.

[17] N. Bazhenov, M. Mustafa, L. San Mauro, A. Sorbi and M. Yamaleev, Classifying equivalence relations in the Ershov hierarchy, *Arch. Math. Logic* **59**(7–8) (2020), 835–864. doi:10.1007/s00153-020-00710-1.

[18] U. Andrews, D. Belin and L. San Mauro, On the structure of computable reducibility on equivalence relations of natural numbers, 2021, arXiv preprint arXiv:2105.12534.

[19] S. Buss, Y. Chen, J. Flum, S. Friedman and M. Müller, Strong isomorphism reductions in complexity theory, *J. Symb. Logic* **76**(4) (2011), 1381–1402. doi:10.2178/jsl/1318338855.

[20] S. Gao and C. Ziegler, On polynomial-time relation reducibility, *Notre Dame J. Form. Log.* **58**(2) (2017), 271–285. doi:10.1215/00294527-3867118.

[21] N. Bazhenov, R. Downey, I. Kalimullin and A. Melnikov, Foundations of online structure theory, *Bull. Symbolic Logic* **25**(2) (2019), 141–181. doi:10.1017/bsl.2019.20.

[22] R. Downey, A. Melnikov and K.M. Ng, Foundations of online structure theory II: The operator approach, *Log. Methods Comput. Sci.* **17**(3) (2021), 6:1–6:35. doi:10.46298/lmcs-17(3:6)2021.

[23] I. Kalimullin, A. Melnikov and K.M. Ng, Algebraic structures computable without delay, *Theoret. Comput. Sci.* **674** (2017), 73–98. doi:10.1016/j.tcs.2017.01.029.

[24] A.G. Melnikov, Eliminating unbounded search in computable algebra, in: *Unveiling Dynamics and Complexity*, J. Kari, F. Manea and I. Petre, eds, Lecture Notes in Computer Science, Vol. 10307, Springer, Cham, 2017, pp. 77–87. doi:10.1007/978-3-319-58741-7_8.

[25] A. Melnikov and K.M. Ng, A structure of punctual dimension two, *Proc. Amer. Math. Soc.* **148**(7) (2020), 3113–3128. doi:10.1090/proc/15020.

[26] V.L. Selivanov and S. Selivanova, Primitive recursive ordered fields and some applications, in: *Computer Algebra in Scientific Computing - 23rd International Workshop, CASC 2021*, F. Boulier, M. England, T.M. Sadykov and E.V. Vorozhtsov, eds, Lect. Notes Comput. Sci., Vol. 12865, Springer, 2021, pp. 353–369. doi:10.1007/978-3-030-85165-1_20.

[27] A.I. Mal'tsev, Constructive algebras. I, *Russ. Math. Surv.* **16**(3) (1961), 77–129. doi:10.1070/RM1961v016n03ABEH001120.

[28] A.I. Mal'cev, *The metamathematics of algebraic systems. Collected papers: 1936–1967*, Vol. 66, North-Holland, Amsterdam, 1971.

[29] P. Odifreddi, *Classical recursion theory: The theory of functions and sets of natural numbers*, Elsevier, 1992.

[30] P.G. Hinman, *Recursion-theoretic hierarchies*, Springer, Berlin, 1978.

[31] W. Ackermann, Zum Hilbertschen Aufbau der reellen Zahlen, *Math. Ann.* **99** (1928), 118–133. doi:10.1007/BF01459088.

[32] K. Mehlhorn, Polynomial and abstract subrecursive classes, *J. Comput. System Sci.* **12**(2) (1976), 147–178. doi:10.1016/S0022-0000(76)80035-9.

[33] R.E. Ladner, On the structure of polynomial time reducibility, *J. ACM* **22**(1) (1975), 155–171. doi:10.1145/321864.321877.

[34] N. Bazhenov, I. Kalimullin, A. Melnikov and K.M. Ng, Online presentations of finitely generated structures, *Theoret. Comput. Sci.* **844** (2020), 195–216. doi:10.1016/j.tcs.2020.08.021.