

LATTICE EMBEDDINGS AND PUNCTUAL LINEAR ORDERS

MARINA DORZHEVA AND ELLEN HAMMATT 

Abstract. We investigate the primitive recursive content of linear orders. We prove that the punctual degrees of rigid linear orders, the order of the integers $\mathbb{Z}$, and the order of the rationals $\mathbb{Q}$ embed the diamond (preserving supremum and infimum). In the cases of rigid orders and the order $\mathbb{Z}$, we further extend the result to embed the atomless Boolean algebra; we leave the case of $\mathbb{Q}$ as an open problem. We also show that our results for the rigid orders, in fact, work for orders having a computable infinite invariant rigid sub-order.

§1. Introduction. In computable algebra, the main objects are computably presented algebraic structures, such as groups, fields, Boolean algebras, and linear orders. A *computable presentation* of a structure $\mathcal{A}$ is a coding of $\mathcal{A}$ by natural numbers so that all relations and functions in the language of $\mathcal{A}$ become computable on their respective indices.

Mal'cev [35] proposed that *computable* presentations of algebraic structures should be studied up to *computable* isomorphism. The idea is that the maps used to compare the objects of study should preserve all of the properties of these objects that we wish to study. This approach is the suitable computable analogue of, e.g., comparing *topological* groups under *topological* group-isomorphism and identifying differentiable manifolds up to *diffeomorphism*. Mal'cev defined a structure to be *computably categorical* (or *autostable*) if it has a unique computable presentation up to computable isomorphism. For example, any finitely generated group is computably categorical, and so is the countable dense order of the rationals (folklore).

The study of the number of computable presentations up to computable isomorphism (i.e., computable dimension) has accumulated many results and techniques; we cite the books [2, 39, 40]. For example, there exist structures having exactly two computable presentations up to computable isomorphism [20]. Computably categorical structures can be rather complicated in general; for instance, their index set is Π_1^1 -complete [18]. However, non-trivial examples of computably categorical structures or structures having finitely many computable presentations have to be *specifically constructed* using intricate techniques. In contrast with these results, in many standard classes, computably categorical structures are easy to describe. For example, a linear order is computably categorical if and only if it

Received November 14, 2024.

2020 *Mathematics Subject Classification*. Primary 03C57, 03D45, 03D99.

Key words and phrases. computable structure, punctual structure, primitive recursion, linear orders, lattice embeddings.

© The Author(s), 2025. Published by Cambridge University Press on behalf of The Association for Symbolic Logic. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted re-use, distribution, and reproduction in any medium, provided the original work is properly cited.

0022-4812/00/0000-0000

DOI:10.1017/jsl.2025.10106



has only finitely many successive pairs [22, 41]. Similar descriptions of computable categoricity are known for Boolean algebras [23, 33], broad classes of abelian groups [21, 36, 44], trees of finite height [34], and ordered abelian groups [19]. In all these cases, if a structure in the class is not computably categorical, then it has infinitely many computable presentations up to computable isomorphism.

Thus, it seems that most algebraic structures encountered in mathematical practice are not particularly interesting within the classical framework of computable categoricity. Consequently, one may question the underlying reasons for this phenomenon. One possible explanation is as follows: the approach discussed above does not assume any time or resource restrictions on our algorithms.

The unbounded search is used so profoundly throughout the literature that most of the algorithmic procedures developed within this framework appear to be completely impractical. It is remarkable that, with some work, it can often be illustrated that many computably presented algebraic structures possess polynomial-time presentations. This is true, for instance, for linear orders, Boolean algebras, torsion-free abelian groups, and for many other broad classes of structures; we cite [1, 7–11, 25].

As was observed in [29], almost all such proofs are focused on making algorithms primitive recursive, and then with a bit more work, it can be shown that the bounds are polynomial-time. On the other hand, to show a structure has no polynomial-time presentation, it is often easiest to show that it does not have a primitive recursive presentation by diagonalising against all primitive recursive presentations.

Motivated by these observations and beginning with [29], there has been a line of investigation into the theory of primitive recursive processes in algebra and analysis. We also cite the survey [5], which is, however, rapidly ageing. There have been many recent results that appeared after the publication of [5], to name a few: [3, 4, 6, 13–17, 27, 28, 31, 38, 43]. The main definition in these investigations is as follows.

DEFINITION 1.1 [29]. A structure with finite language is a *punctual* (formally, fully primitive recursive) presentation if it has $\mathbb{N}$ as its domain and all functions and relations in the language of $\mathcal{A}$ are primitive recursive on $\mathbb{N}$.

Following the ideas of Mal'cev, we would like to compare punctual structures up to primitive recursive isomorphism. Now the difference with general computable structures becomes immediately apparent. Notice that the inverse of a primitive recursive function $f : \omega \rightarrow \omega$ may not be primitive recursive. This induces a natural *reduction* on punctual presentations of a given structure.

DEFINITION 1.2. Let $PR(\mathcal{M})$ be the collection of punctual presentations of $\mathcal{M}$. For $\mathcal{A}, \mathcal{B} \in PR(\mathcal{M})$, we say that $\mathcal{A}$ is *punctually reducible* to $\mathcal{B}$, written $\mathcal{A} \leq_{pr} \mathcal{B}$, if there is a primitive recursive isomorphism $p : \mathcal{A} \rightarrow \mathcal{B}$. We write $\mathcal{A} \cong_{pr} \mathcal{B}$ and say that $\mathcal{A}$ and $\mathcal{B}$ are *punctually equivalent* if $\mathcal{A} \leq_{pr} \mathcal{B}$ and $\mathcal{B} \leq_{pr} \mathcal{A}$. Then the *punctual degrees* of $\mathcal{M}$, $\mathbf{PR}(\mathcal{M})$, are defined to be $PR(\mathcal{M}) / \cong_{pr}$.

We remark that $\mathcal{A} \cong_{pr} \mathcal{B}$ does not necessarily imply that there is an isomorphism f between the structures so that both f and f^{-1} are primitive recursive.

Some initial steps in the investigation of the punctual degrees of structures were taken in [26]. Perhaps unsurprisingly, with some effort, pathological examples of structures having unusual or unexpected punctual degrees can be manufactured; e.g., [6, 38]. What is certainly unexpected is that the study of the punctual degrees of

even the most familiar structures has proven to be a challenging task. In [37], it was discovered that the punctual degrees of the dense linear order, the random graph, and the universal countable abelian p -group are pairwise non-isomorphic; note all three structures are homogeneous and computably categorical. In [30], it was shown that any countable distributive lattice can be embedded into the punctual degrees of the semi-ring of the natural numbers, as well as into the punctual degrees of any rigid finitely generated (infinite) structure. In [13], it was established that the punctual degrees of any computably almost rigid structure, including $(\omega, <)$ and the linear order $(\mathbb{Z}, <)$, are dense. Surprisingly, the punctual degrees of $(\mathbb{Q}, <)$ are *not* dense [32]. It appears that primitive recursive procedures remain poorly understood, even for the well-studied class of computably presentable linear orders.

We wish to further understand the punctual degrees of linear orders. In particular, we are interested in whether any countable distributive lattice can be embedded into the punctual degrees of a linear order $\mathcal{L}$. Since the atomless Boolean algebra is universal for distributive lattices, we aim to show this by embedding the atomless Boolean algebra into the punctual degrees of $\mathcal{L}$, preserving the supremums and infimums. The first step towards this is to embed the *diamond*, which consists of presentations $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ of $\mathcal{L}$ where $\mathcal{A} <_{pr} \mathcal{B}$, $\mathcal{C} \mid_{pr} \mathcal{D}$, $\sup(\mathcal{C}, \mathcal{D}) = \mathcal{B}$ and $\inf(\mathcal{C}, \mathcal{D}) = \mathcal{A}$. Below is the first main result of the paper.

THEOREM 1.1 (The Diamond). *We can embed the diamond into the punctual degrees of $\mathcal{L}$ for the following linear orders:*

1. *Linear orders of the form $\mathcal{L} = \mathcal{L}_{-1} + \mathcal{L}_0 + \mathcal{L}_1$ where $\mathcal{L}_{-1}, \mathcal{L}_0, \mathcal{L}_1$ are punctually presentable linear orders and $\mathcal{L}_0$ is infinite and invariant in $\mathcal{L}$ (for example, rigid linear orders including ordinals).*
2. *The order $\mathbb{Z}$ of the integers.*
3. *The dense linear order $\mathbb{Q}$.*

We note that a linear order is computably presentable if and only if it is punctually presentable [29]. In this paper, we will use the term *punctually presentable*. We also note that in (1) above, we prove a stronger result: there is an embedding of the diamond between *any* presentations $\mathcal{A} <_{pr} \mathcal{B}$ such that $\mathcal{A}$ and $\mathcal{B}$ reveal elements outside $\mathcal{L}_0$ at the same stage. It follows that for any punctual rigid linear order (e.g., any computable ordinal), we can embed the diamond between any pair of punctual copies $\mathcal{A} <_{pr} \mathcal{B}$. Given the density of $\mathbf{PR}(\mathbb{Z})$ ([13]), this stronger property likely holds for $\mathbb{Z}$ as well, however, our proof does not show this. Given the non-density of $\mathbf{PR}(\mathbb{Q})$ ([32]), this stronger result cannot possibly hold in the case of $\mathbb{Q}$. In the first two cases, using different arguments, we can extend the construction to embed the atomless Boolean algebra. Below is the second main result of the paper.

THEOREM 1.2. *There is an embedding of the atomless Boolean algebra into the punctual degrees of $\mathcal{L}$ for the following linear orders:*

1. *Linear orders of the form $\mathcal{L} = \mathcal{L}_{-1} + \mathcal{L}_0 + \mathcal{L}_1$ where $\mathcal{L}_{-1}, \mathcal{L}_0, \mathcal{L}_1$ are punctually presentable linear orders and $\mathcal{L}_0$ is infinite and invariant in $\mathcal{L}$.*
2. *The order $\mathbb{Z}$ of the integers.*

As for the case of the diamond, in (1), we prove the stronger result: there is an embedding of the atomless Boolean algebra between $\mathcal{A}$ and $\mathcal{B}$ for *any* presentations $\mathcal{A} <_{pr} \mathcal{B}$ such that $\mathcal{A}$ and $\mathcal{B}$ reveal elements outside $\mathcal{L}_0$ at the same stage. Therefore,

as in the diamond case, it follows that for any rigid linear order, we can embed the diamond between any pair of punctual copies $\mathcal{A} <_{pr} \mathcal{B}$. We leave open whether the theorem above holds for $\mathbb{Q}$. The reader will see that our embedding of the diamond in the case of the rationals does not even seem to allow us to embed the 8-element Boolean algebra, let alone the dense atomless one. The reader may find this blockage both unusual and unexpected. However, given the unusually high combinatorial complexity of the proof of the non-density of $\mathbf{PR}(\mathbb{Q})$ mentioned above ([32]), there is perhaps no big surprise here.

In Section 2, we embed the diamond and the atomless Boolean algebra into the punctual degrees of invariant linear orders (proving Theorems 1.1 and 1.2 in the case of (1)). Following this, in Section 3, we embed the diamond into the punctual degrees of the rationals (proving Theorem 1.1 in the case of (3)). Lastly, in Section 4, we embed the diamond and the atomless Boolean algebra into the punctual degrees of the integers (proving Theorems 1.1 and 1.2 in the case of (2)).

§2. Invariant linear orders. Our first main result is concerned with linear orders $\mathcal{L}$ of the form $\mathcal{L} = \mathcal{L}_{-1} + \mathcal{L}_0 + \mathcal{L}_1$, where $\mathcal{L}_{-1}$ and $\mathcal{L}_1$ may be empty, and $\mathcal{L}_0$ is infinite and invariant. We say that $\mathcal{L}_0$ is invariant in $\mathcal{L}$ if for any $\varphi \in \text{Aut}(\mathcal{L})$, $\varphi \upharpoonright_{\mathcal{L}_0} = \text{id}_{\mathcal{L}_0}$. Note that it follows from the definition that $\mathcal{L}_0$ is rigid. Examples of linear orders that contain an infinite invariant $\mathcal{L}_0$ include ordinals and, more generally, infinite rigid linear orders. Of course, there are examples of such linear orders that are not rigid; e.g., consider $\mathbb{Q} + \mathbb{N} + \mathbb{N}^* + \mathbb{Q}$ in which $\mathbb{N} + \mathbb{N}^*$ is clearly invariant. Conversely, the existence of a rigid $\mathcal{L}_0$ is not sufficient. For example, each copy of $\mathbb{N}$ is rigid in $\mathbb{N} \times \mathbb{Z}$ but there is no $\mathcal{L}_0$ in $\mathbb{N} \times \mathbb{Z}$ that is invariant. The idea is that $\mathcal{L}$ has an identifiable infinite section that is rigid within the linear order, and we will use this section to apply the construction.

REMARK 2.1. We highlight that, in addition to requiring that $\mathcal{L}$ is punctually presentable, we require $\mathcal{L}_{-1}, \mathcal{L}_0, \mathcal{L}_1$ to be punctually presentable as well; note that this is automatically true if $\mathcal{L}_0$ is a (closed, open, or half-open) interval in $\mathcal{L}$. While this covers many linear orders which contain an invariant (convex) sub-order, it does not cover all such linear orders. In [12], Coles, Downey, and Khoussainov proved that there is a computable linear order $\mathcal{L} = \mathcal{A} + \omega^*$ such that $\mathcal{A}$ is not computably presentable. For such linear orders, we do not yet know that there exist punctual presentations $\mathcal{B} >_{pr} \mathcal{A}$ such that there is a primitive recursive $p : \mathcal{A} \rightarrow \mathcal{B}$ which is the identity outside an infinite invariant interval. A new construction is required for the case when $\mathcal{L} = \mathcal{A} + \omega^*$, where $\mathcal{A}$ has no computable presentation.

In this section, we prove the following theorem.

THEOREM 2.2. *Let $\mathcal{L}$ be a linear order such that there are $\mathcal{L} = \mathcal{L}_{-1} + \mathcal{L}_0 + \mathcal{L}_1$ with the following properties*

- $\mathcal{L}_0$ is infinite and invariant in $\mathcal{L}$
- $\mathcal{L}_{-1}, \mathcal{L}_0$, and $\mathcal{L}_1$ are punctually presentable linear orders.

For any punctual presentations $\mathcal{A}, \mathcal{B}$ of $\mathcal{L}$ such that $\mathcal{A} <_{pr} \mathcal{B}$ and $\mathcal{A}$ and $\mathcal{B}$ reveal elements outside $\mathcal{L}_0$ at the same stage, there exist punctual presentations $\mathcal{C}, \mathcal{D}$ such that $\mathcal{C}, \mathcal{D}$ are incomparable under $\leq_{pr}$, $\mathcal{A} = \inf(\mathcal{C}, \mathcal{D})$, and $\mathcal{B} = \sup(\mathcal{C}, \mathcal{D})$.

As mentioned, we allow $\mathcal{L}_{-1}$ or $\mathcal{L}_1$ (or both) to be empty.

The proof of Theorem 2.2 is delayed until Section 2.2. Before we prove the theorem, we need to establish a preliminary result that is contained in the section below. The result says that, under the assumptions of Theorem 1.1 (1), there exist two punctual copies of $\mathcal{L}$ with the properties required to serve as the ‘top’ and the ‘bottom’ of the diamond. Therefore, the combination of this result along with Theorem 2.2 proves Theorem 1.1 (1).

2.1. Constructing presentations $\mathcal{A}, \mathcal{B}$, of $\mathcal{L}$ with $\mathcal{B}$ above $\mathcal{A}$. Note that the theorem below does not require the existence of an invariant (convex) sub-order. The result is not difficult but it appears to be new. It will allow us to show that linear orders in Theorem 1.1 (1) have punctual presentations $\mathcal{A} <_{pr} \mathcal{B}$ with the properties in Theorem 2.2.

THEOREM 2.3. *If $\mathcal{L}$ is an infinite punctual linear order, then there exist punctual presentations $\mathcal{A}, \mathcal{B}$ of $\mathcal{L}$ such that $\mathcal{A} <_{pr} \mathcal{B}$.*

PROOF. We build punctual presentations of $\mathcal{L}$, $\mathcal{A}$, and $\mathcal{B}$ along with a primitive isomorphism $p : \mathcal{A} \rightarrow \mathcal{B}$. We need to satisfy the following requirements:

$$R_e : p_e : \mathcal{B} \rightarrow \mathcal{A} \text{ is not an isomorphism,}$$

where $(p_e)_{e \in \omega}$ is an effective enumeration of all primitive recursive functions. The proof is non-uniform. Recall that a right (left) limit point in a linear order $\mathcal{L}$ is a point l for which there exists a sequence $\langle l_i \rangle_{i \in \mathbb{N}}$ such that for all $i \in \mathbb{N}$, $l_i > l$ ($l_i < l$), the sequence is strictly decreasing (increasing), and for all $a > l$ ($a < l$), there is i such that $a > l_i$ ($a < l_i$).

Since $\mathcal{L}$ is punctual, within primitive recursively bounded time $t[s]$, we can decide $x <_{\mathcal{L}} y$ for all $x, y \leq s$. We call the time period $(t(s-1), t[s])$ stage s . $\mathcal{L}[s]$ denotes the substructure of $\mathcal{L}$ obtained by computing $<_{\mathcal{L}}$ on every pair of elements up to s . For structures we are building ($\mathcal{A}$ and $\mathcal{B}$), we consider $\mathcal{C}[s]$ to be the presentation we have built so far at stage s . Throughout the construction we will *add a new element to $\mathcal{C}$* , this means we add the $<_{\mathbb{N}}$ -least element $x \in \mathbb{N}$ that is not yet in $\mathcal{C}[s]$ and we define $<_{\mathcal{C}}$ as described in the construction. In this construction, we will ensure that $\mathcal{C}$ is punctual by ensuring that by the end of stage s , we have added the element s to $\mathcal{C}[s]$. We ensure that isomorphisms are primitive recursive by ensuring that it is defined on all elements in $\mathcal{C}[s]$ at stage s .

Case 1: $\mathcal{L}$ has a right limit point l . Fix some punctual presentation of $\mathcal{L}$. Non-uniformly fix the limit point l . Without loss of generality, we may assume that in this presentation $l = 0$. We will build punctual presentations $\mathcal{A}$ and $\mathcal{B}$ that are isomorphic to $\mathcal{L}$ such that $\mathcal{A} <_{pr} \mathcal{B}$. The idea is that $\mathcal{B}$ will always copy $\mathcal{A}$ so that it can naturally extend the definition of a primitive recursive $p : \mathcal{A} \rightarrow \mathcal{B}$, except we occasionally add an extra point in $\mathcal{B}$ which will ‘float’ just to the right of l while we diagonalise. We call this the *island technique*, where this extra point acts as a *floating island* an unknown distance away from the rest of the elements on the right of l . While diagonalising, we will grow $\mathcal{A}$ towards l and $\mathcal{B}$ towards the extra point to the right of l . We copy the other points in $\mathcal{L}$ after each diagonalisation.

In this construction, we will use the macro *grow $\mathcal{C}$ towards y* to mean, add a new element to $\mathcal{C}$ so that it is the least element greater than y under $<_{\mathcal{C}}$.

Construction. At stage 0, we introduce 0 into $\mathcal{A}$ and $\mathcal{B}$ which will be isomorphic to the limit point $l \in \mathcal{L}$. We will call this element l in all structures and define $p(l) = l$. Go to the next stage and begin to work for requirement R_0 .

At stage $s > 0$, we are working for some requirement R_e . The strategy for R_e as described below.

Strategy for R_e . We start by adding the floating point x into $\mathcal{B}$ so that it is the least element greater than l while we are working on this requirement. Wait for $p_e(x)$ to converge at some stage t and then wait for $p_e(y)$ to converge for all $y \in \mathcal{B}[t]$. While we wait for this, continue to grow $\mathcal{B}$ towards x and grow $\mathcal{A}$ towards l . Once we have seen $p_e(y)$ converge for all $y \in \mathcal{B}[t]$, add an element $\hat{x}$, so that it is the least element greater than l in $\mathcal{A}$ and define $p(\hat{x}) = x$. Note that at this stage, $\mathcal{A}[s]$ and $\mathcal{B}[s]$ are isomorphic. Go to recovery.

Recovery. Let $l_{\mathcal{L}}^+$, $l_{\mathcal{A}}^+$, and $l_{\mathcal{B}}^+$ be the least element greater than l in $\mathcal{L}[t]$, $\mathcal{A}[t]$, and $\mathcal{B}[t]$ respectively, where t is the stage that we started to work for the current requirement. We begin by adding elements to $\mathcal{A}$ and $\mathcal{B}$ so that they are isomorphic to $\mathcal{L}$ outside $[l, l^+]$. Then we wait for the interval $[l_{\mathcal{L}}, l_{\mathcal{L}}^+]$ to *catch up*. Let m be the size of $[l_{\mathcal{A}}, l_{\mathcal{A}}^+]$ at stage s (the current stage). We wait for $[l_{\mathcal{L}}, l_{\mathcal{L}}^+]$ to contain m elements. While we wait, we copy any elements outside the interval that appear into $\mathcal{A}$ and $\mathcal{B}$. Once we have seen m elements in $[l_{\mathcal{L}}, l_{\mathcal{L}}^+]$, $\mathcal{A}$ and $\mathcal{B}$ are both currently isomorphic to $\mathcal{L}$, so recovery has finished. Move to the next requirement on the list.

Verification. Notice that each R requirement is worked on for finitely many stages because we only wait for a primitive recursive function to halt on finitely many elements. During recovery, we wait for the presentation $\mathcal{L}$ to grow a finite amount towards the limit point l . Since l is a right limit point in $\mathcal{L}$, this must eventually happen and hence we stay in recovery for finitely many stages after each requirement. While we wait in recovery, $\mathcal{A}$ and $\mathcal{B}$ continue adding elements corresponding to those outside the $[l, l^+]$ interval. This means that while we wait for the interval in $\mathcal{L}$ to ‘catch up’, $\mathcal{A}$ and $\mathcal{B}$ stay punctual. Therefore $\mathcal{A}$ and $\mathcal{B}$ are punctual presentations of $\mathcal{L}$.

We now prove that all R requirements are met. We are in recovery for finitely many stages and diagonalisation ends after p_e halts on finitely many inputs, therefore every requirement is eventually worked on. Fix a requirement R_e , there is a stage s where we begin to work on requirement R_e . Once $p(x)$ converges, we have two cases, either $p_e(x) \leq l$ or $p_e(x) > l$.

If $p_e(x) \leq l$, then by our strategy, there are strictly more elements in $\mathcal{B}$ to the left of x than there are in $\mathcal{A}$ to the left of $p_e(x)$, as shown in Figure 1.

Similarly, if $p_e(x) > l$, there are strictly more elements in $\mathcal{B}$ to the right of x than there are in $\mathcal{A}$ to the right of $p_e(x)$, as shown in Figure 2.

We show that in either case we meet the requirement. Let t be the stage where we first see $p_e(x) \downarrow$ while we are working on requirement R_e . Let z be the least element greater than l at stage t . While we wait for $p_e(y)$ to converge for all $y \in \mathcal{B}[t]$, we only add elements into $\mathcal{A}$ between l and z . Note that there are finitely many $y \in \mathcal{B}[t]$, so eventually $p_e(y)$ converges for all $y \in \mathcal{B}[t]$. At this stage, p_e must have shown that either it is not order-preserving or it is not injective. Therefore requirement R_e is eventually met.

Case 2: $\mathcal{L}$ has a left limit point. Act exactly as in case 1 but swap left and right. Now *grow $\mathcal{C}$ towards y* means that we add the least element not yet in $\mathcal{C}[s]$ so that

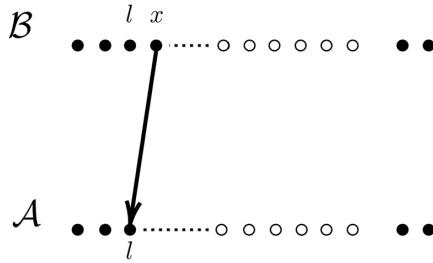


FIGURE 1. Solid elements are the old elements from before we started to work on requirement R_e , while hollow elements are new elements. The arrow represents p_e . The dashes show where we will continue to add points while we work on this requirement.

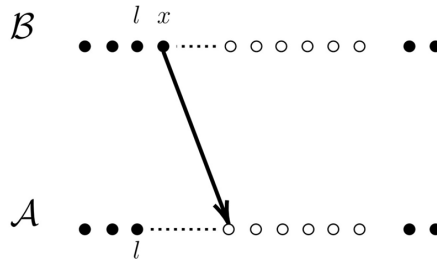


FIGURE 2. Solid elements are the old elements from before we started to work on requirement R_e started, while hollow elements are new elements. The arrow represents p_e . The dashes show where we will continue to add points while we work on this requirement.

it is the greatest element less than y . When we define l^+ in recovery, it is now the greatest element less than l .

Case 3: $\mathcal{L}$ has a sub-interval of order type $\mathbb{N} + \mathbb{N}^*$. Fix some punctual presentation of $\mathcal{L}$. Non-uniformly fix a, b in this presentation of $\mathcal{L}$ such that $[a, b] \cong \mathbb{N} + \mathbb{N}^*$. Without loss of generality we may assume that in this presentation $a = 0$ and $b = 1$. For any element x we can primitive recursively decide if x belongs to the interval $[a, b]$ in $\mathcal{L}$.

In this construction, we will build the interval $[a, b]$ by building the sequences a_i and b_i such that $a_0 = a$, $a_i <_C a_{i+1} <_C b$ and $b_0 = b$, $b_i >_C b_{i+1} >_C a_j$ for any $j \in \omega$. Eventually, the sequence a_i will be isomorphic $\mathbb{N}$ and b_i to $\mathbb{N}^*$. We use the macro *grow $\mathbb{N}(\mathbb{N}^*)$ in $[a, b]$ in $\mathcal{C}$* to mean, add a new element to $\mathcal{C}$ to extend the initial segment of a_i (b_i) that we have so far. Now we detail the formal construction.

Construction. At stage 0, add the elements $a = 0$ and $b = 1$ into $\mathcal{A}[0]$ and $\mathcal{B}[0]$. Define $p(0) = 0$ and $p(1) = 1$. Go to the next stage and begin to work towards meeting requirement R_0 .

At stage $s > 0$, we are working for some requirement R_e . The strategy for R_e is described below.

Strategy for R_e . At every stage, compute $p_e(x)$ on every $x \in [a, b]_{\mathcal{B}[s]}$ and wait for p_e to show it is not an isomorphism. While we wait, grow $\mathbb{N}$ and $\mathbb{N}^*$ in $\mathcal{B}$ and grow $\mathbb{N}$ in $\mathcal{A}$. Define p on all elements as they are added to $\mathcal{A}$ to the corresponding elements added to $\mathcal{B}$ while growing $\mathbb{N}$. Once p_e shows it is not an isomorphism, add corresponding elements to $\mathcal{A}$ that were added to $\mathbb{N}^*$ in $\mathcal{B}$ during this strategy and extend p . Note that at this stage, $\mathcal{A}[s]$ and $\mathcal{B}[s]$ are isomorphic and p is onto $\mathcal{B}[s]$. Finally add necessary elements so that $\mathcal{A}[s]$ and $\mathcal{B}[s]$ are isomorphic to $\mathcal{L}[s]$ outside $[a, b]$ and extend p naturally. Go to the next stage and begin to work on the strategy for R_{e+1} .

Verification. It is clear that $\mathcal{A}$ and $\mathcal{B}$ are punctual because at every stage we add at least one element. Now, let us show that p_e is not an isomorphism. Compute p_e on the elements of $\mathcal{B}$. Since $\mathcal{A}$ and $\mathcal{B}$ are not adding any elements outside $[a, b]$, eventually there is an element c in $\mathcal{B}$ that is part of $\mathbb{N}^*$ mapped to an element in $\mathcal{A}$ that is part of $\mathbb{N}$ (or we will see that p_e is not injective). After this point, we continue to add elements to the left of c in $\mathcal{B}$ however no elements are added to the left of $p_e(c)$, therefore eventually p_e will show that it is not order-preserving or not injective. Then each requirement is met at some stage. By construction, the interval $[a, b]$ is isomorphic to $\mathbb{N} + \mathbb{N}^*$ and at every recovery, we ensure that $\mathcal{A}$ and $\mathcal{B}$ are isomorphic to $\mathcal{L}$ outside $[a, b]$. Therefore $\mathcal{A}$ and $\mathcal{B}$ are punctual presentations of $\mathcal{L}$ as desired.

Case 4: Not case 1, 2, or 3. If $\mathcal{L}$ has no greatest or least element, then we can proceed as in case 1 or 2 with $\mathcal{L} \cup \{\infty\}$ or $\mathcal{L} \cup \{-\infty\}$, respectively. Otherwise, $\mathcal{L}$ is of the form $\mathbb{N} + \mathbb{Z} \cdot \mathbb{Q} + \mathbb{N}^*$ [42]. In this case, in a similar way to case 3, we build $\mathcal{A}$ and $\mathcal{B}$ in three sections, isomorphic to $\mathbb{N}$, $\mathbb{Z} \cdot \mathbb{Q}$, and $\mathbb{N}^*$, respectively. Let $\mathcal{P}$ be some punctual presentation of $\mathbb{Z} \cdot \mathbb{Q}$. As in case 3, while working towards meeting some R_e requirement we grow $\mathbb{N}$ and $\mathbb{N}^*$ in $\mathcal{B}$ and we grow $\mathbb{N}$ in $\mathcal{A}$ until we see a disagreement from p_e . Once we see this we recover by making $\mathcal{A}[s]$ and $\mathcal{B}[s]$ isomorphic and then add elements from $\mathcal{P}[s]$ to $\mathcal{A}[s]$ and $\mathcal{B}[s]$ in between the copies of $\mathbb{N}$ and $\mathbb{N}^*$ that we are building. Then move to the next requirement. The verification of the construction in case 3 applies in order to show that $\mathcal{A}$ and $\mathcal{B}$ are as desired. $\dashv$

In Theorem 2.2, we consider $\mathcal{A} <_{pr} \mathcal{B}$ and $\mathcal{A}$ and $\mathcal{B}$ reveal elements outside $\mathcal{L}_0$ at the same stage; the intuition is that it will allow us to restrict our strategies to the invariant part of $\mathcal{L}$. Note that in the case that $\mathcal{L}$ is rigid we can define $\mathcal{L}_{-1}$ and $\mathcal{L}_1$ to be empty. In this case, any punctual presentations $\mathcal{A}, \mathcal{B}$ of $\mathcal{L}$ such that $\mathcal{A} <_{pr} \mathcal{B}$ will suffice. The following lemma shows that in the case that $\mathcal{L}$ is not rigid, $\mathcal{A}$ and $\mathcal{B}$ with this property exist to prove Theorem 1.1 (1).

LEMMA 2.4. *Suppose $\mathcal{L} = \mathcal{L}_{-1} + \mathcal{L}_0 + \mathcal{L}_1$, where $\mathcal{L}_0$ is infinite and invariant, and $\mathcal{L}_{-1}, \mathcal{L}_0$, and $\mathcal{L}_1$ are punctually presentable linear orders. Then there exist punctual presentations $\mathcal{A}, \mathcal{B}$ of $\mathcal{L}$ such that $\mathcal{A} <_{pr} \mathcal{B}$ and $\mathcal{A}$ and $\mathcal{B}$ reveal elements outside $\mathcal{L}_0$ at the same stage.*

PROOF. Fix punctual presentations of $\mathcal{L}_{-1}, \mathcal{L}_0$, and $\mathcal{L}_1$, we will slightly abuse the notation and denote these presentations by $\mathcal{L}_{-1}, \mathcal{L}_0$, and $\mathcal{L}_1$, respectively. Apply Theorem 2.3 to $\mathcal{L}_0$ to obtain punctual presentations $\mathcal{L}_0 <_{pr} \hat{\mathcal{L}}_0$. Let N_{-1}, N_0, N_1 be a partition of $\mathbb{N}$ with $|N_i| = |\mathcal{L}_i|$ and for each infinite N_i , there is an injective primitive

recursive $r_i : \mathbb{N} \rightarrow N_i$ with primitive recursive inverse. We will treat r_i as a map on $\mathcal{L}_i$. Now let $\mathcal{A} = r_{-1}(\mathcal{L}_{-1}) + r_0(\mathcal{L}_0) + r_1(\mathcal{L}_1)$ and $\mathcal{B} = r_{-1}(\hat{\mathcal{L}}_{-1}) + r_0(\hat{\mathcal{L}}_0) + r_1(\mathcal{L}_1)$. Therefore, elements of $\mathcal{L}_{-1} \cup \mathcal{L}_1$ will appear in $\mathcal{A}$ at the same stage they appear in $\mathcal{B}$. Let p_0 be a primitive recursive isomorphism from $\mathcal{L}_0$ to $\hat{\mathcal{L}}_0$ given by Theorem 2.3, and let $p_{-1}(x) = x$ and $p_1(x) = x$. Define $q : \mathcal{A} \rightarrow \mathcal{B}$ as $q(x) = p_i(x)$ for $x \in N_i$. This is a primitive recursive isomorphism. Suppose there is a primitive recursive isomorphism $r : \mathcal{B} \rightarrow \mathcal{A}$. Then $r_0^{-1} \circ r \circ r_0$ is a primitive recursive isomorphism from $\hat{\mathcal{L}}_0$ to $\mathcal{L}_0$, which is a contradiction. So $\mathcal{A} <_{pr} \mathcal{B}$ and $\mathcal{A}$ and $\mathcal{B}$ reveal elements outside $\mathcal{L}_0$ at the same stage as desired. $\dashv$

2.2. Embedding the diamond. Now we prove Theorem 2.2 by construction. We build structures $\mathcal{C}$ and $\mathcal{D}$ so that they meet the following requirements:

- $R_e^0 : p_e : \mathcal{C} \rightarrow \mathcal{D}$ is not an isomorphism,
- $R_e^1 : p_e : \mathcal{D} \rightarrow \mathcal{C}$ is not an isomorphism,
- $P_{i,j,k} : \text{if } p_j : \mathcal{C} \rightarrow \mathcal{P}_i, p_k : \mathcal{D} \rightarrow \mathcal{P}_i \text{ are p.r. isomorphisms, then } \mathcal{B} \leq_{pr} \mathcal{P}_i,$
- $Q_{i,j,k} : \text{if } q_j : \mathcal{P}_i \rightarrow \mathcal{C}, q_k : \mathcal{P}_i \rightarrow \mathcal{D} \text{ are p.r. isomorphisms, then } \mathcal{P}_i \leq_{pr} \mathcal{A},$

where $\langle p_e \rangle_{e \in \mathbb{N}}$ and $\langle \mathcal{P}_i \rangle_{i \in \mathbb{N}}$ are effective enumerations of all primitive recursive functions and punctual presentations, respectively. We also build primitive recursive isomorphisms $\gamma_{\mathcal{C}} : \mathcal{C} \rightarrow \mathcal{B}$, $\gamma_{\mathcal{D}} : \mathcal{D} \rightarrow \mathcal{B}$, $\delta_{\mathcal{C}} : \mathcal{A} \rightarrow \mathcal{C}$, $\delta_{\mathcal{D}} : \mathcal{A} \rightarrow \mathcal{D}$. Let p be the primitive recursive isomorphism that witnesses $\mathcal{A} <_{pr} \mathcal{B}$.

We arrange an effective list of R requirements in order type ω and meet them one by one. While working towards an R requirement, we will copy $\mathcal{B}$ into one of our structures, while the other copies $\mathcal{A}$. $\mathcal{B} >_{pr} \mathcal{A}$ so $\mathcal{B}$ must infinitely often introduce elements that do not show up in $\mathcal{A}$ for unbounded lengths of time. These elements must lie within $\mathcal{L}_0$ since elements outside $\mathcal{L}_0$ appear at the same stage. Therefore when p_e converges on an element that is not yet in the range of p , p_e cannot possibly be an isomorphism and we can move to the next requirement. Notice that at any stage, $\mathcal{C}$ and $\mathcal{D}$ differ only within L_0 , this fact helps us meet P and Q requirements.

For structures we are building ($\mathcal{C}$ and $\mathcal{D}$), we consider $\mathcal{E}[s]$ to be the substructure that we have built so far at stage s . Throughout the construction, we will *add a new element to $\mathcal{E}$* , this means we add the $<_{\mathbb{N}}$ -least element $x \in \mathbb{N}$ that is not yet in $\mathcal{E}[s]$ and we define $\gamma_{\mathcal{E}}(x)$ immediately and then we define $x <_{\mathcal{E}} y$ by computing $\gamma_{\mathcal{E}}(x) <_{\mathcal{B}} \gamma_{\mathcal{E}}(y)$. Thus, as long as $s \in \mathcal{E}[s]$ by stage s , $\mathcal{E}$ is punctual.

We consider $\mathcal{A}[s]$ to be the structure obtained by computing $<_{\mathcal{A}}$ on $\{i : i < s\}$, while we consider $\mathcal{B}[s]$ to be the structure obtained by computing $<_{\mathcal{B}}$ on $\{i : i < \max\{p(i) : i < s\}\}$. Therefore we see that $\mathcal{B}$ contains all elements in $\mathcal{A}$ as well as maybe some extra elements. Note that we may assume that at stage s , $p(i)[s] \downarrow$ for all $i < s$.

We note that the above conventions described are different from those in the other sections because in this case we are following presentations $\mathcal{A}$ and $\mathcal{B}$ that we are not in control of. Now we describe the formal construction.

2.2.1. Construction. Fix an effective list of all R requirements. At stage 0, begin the strategy of the first requirement on the list. We now detail the strategies we enact

depending on which requirement we are working on. We continue this strategy at every stage until the strategy tells us to go to the next requirement.

Strategy for R_e^0 . For stage s , while we are working on this requirement, we define $\gamma_C, \gamma_D, \delta_C$, and δ_D as follows. First, we copy $\mathcal{A}$ into $\mathcal{C}$ and $\mathcal{D}$ as follows. For each $x \in \mathcal{A}[s] \setminus \mathcal{A}[s-1]$, do the following

- If $\gamma_{\mathcal{E}}^{-1}(p(x))[s] \uparrow$, then add a new element to $\mathcal{E}$, y , and define $\delta_{\mathcal{E}}(x) = y$ and $\gamma_{\mathcal{E}}(y) = p(x)$ (where $\mathcal{E} \in \{\mathcal{C}, \mathcal{D}\}$).
- If $\gamma_{\mathcal{E}}^{-1}(p(x))[s] \downarrow$, then define $\delta_{\mathcal{E}}(x) = \gamma_{\mathcal{E}}^{-1}(p(x))$ (where $\mathcal{E} \in \{\mathcal{C}, \mathcal{D}\}$).

Then we copy $\mathcal{B}$ into $\mathcal{C}$ as follows. For each $y \in \mathcal{B}[s] \setminus \mathcal{B}[s-1]$, if there is no $i \in \mathcal{A}[s]$ such that $p(i) = y$, then add a new element x to $\mathcal{C}$, and define $\gamma_C(x) = y$.

If there is $x \in \mathcal{C}[s]$ such that $p_e(x)[s] \downarrow$ and $\gamma_D^{-1}(\gamma_C(x))[s] \uparrow$, move to the next requirement, otherwise, go to the next stage and continue this strategy.

Strategy for R_e^1 . Do the same strategy but switch the roles of $\mathcal{C}$ and $\mathcal{D}$.

2.2.2. Verification. We show that all requirements are met.

LEMMA 2.5. $\gamma_C : \mathcal{C} \rightarrow \mathcal{B}$ and $\gamma_D : \mathcal{D} \rightarrow \mathcal{B}$ are primitive recursive isomorphisms.

PROOF. We first show that γ_C is surjective. Let $y \in \mathcal{B}[s] \setminus \mathcal{B}[s-1]$. If $\mathcal{C}$ is copying $\mathcal{B}$ at stage s , then y is added to the range of γ_C . If $\mathcal{C}$ is not copying $\mathcal{B}$ at stage s , then since p is an isomorphism from $\mathcal{A} \rightarrow \mathcal{B}$, there is a stage t such that y enters the range of p . At stage t , we add an element to $\mathcal{C}$ and include y to the range of γ_C . Therefore γ_C is surjective. By construction, γ_C is injective, order-preserving and γ_C is defined on x as soon as x enters $\mathcal{C}$. Therefore, γ_C is a primitive recursive isomorphism.

The exact same argument can be used for $\mathcal{D}$ in regards to defining γ_D . $\dashv$

LEMMA 2.6. $\delta_C : \mathcal{A} \rightarrow \mathcal{C}$ and $\delta_D : \mathcal{A} \rightarrow \mathcal{D}$ are primitive recursive isomorphisms.

PROOF. First, we show that δ_C is surjective. Let $x \in \mathcal{C}$, we will show that $x \in \text{range}(\delta_C)$. The preimage of δ_C for all elements in $\mathcal{C}$ is defined immediately unless $\mathcal{C}$ is copying $\mathcal{B}$ and $\mathcal{B}$ adds an element before $\mathcal{A}$ does. In this case, there exists i such that $p(i) = \gamma_C(x)$. At this stage, we will define $\delta_C(i) = x$. Therefore δ_C is surjective. By construction, δ_C is injective and order-preserving and δ_C is defined on x as soon as x enters $\mathcal{A}$. Therefore δ_C is a primitive recursive isomorphism.

The exact same argument can be used for $\mathcal{D}$ in regards to defining δ_D . $\dashv$

LEMMA 2.7. For all $e \in \mathbb{N}$, requirements R_e^0 and R_e^1 are met.

PROOF. Suppose at stage t we started working for some R_e^0 requirement. While we work on this requirement, $\mathcal{C}$ is currently copying $\mathcal{B}$ while $\mathcal{D}$ copies $\mathcal{A}$. Suppose there is no stage $s > t$ such that for some $x \in \mathcal{C}$, $p_e(x)[s] \downarrow$ and $\gamma_D^{-1}(\gamma_C(x))[s] \uparrow$. Then for all stages $s > t$, if $p_e(x)[s] \downarrow$ on $x \in \mathcal{C}[s]$ then $\gamma_D^{-1}(\gamma_C(x))[s] \downarrow$. Then the computation of p_e gives a primitive recursive time bound on the computation of $p^{-1}(\gamma_C(x))$; this is because $\mathcal{D}$ copies $\mathcal{A}$ during this stage. Therefore, p^{-1} is primitive recursive, a contradiction. Hence there is a stage s such that $p_e(x)[s] \downarrow$ on some $x \in \mathcal{C}$ such that $\gamma_D^{-1}(\gamma_C(x))[s] \uparrow$. Fix such an x .

We now show that R_e^0 is met. Notice that x must be in $\mathcal{L}_0$ and $\gamma_D^{-1}(\gamma_C)$ is an isomorphism from $\mathcal{C} \rightarrow \mathcal{D}$. Since $\mathcal{L}_0$ is invariant in $\mathcal{L}$, p_e cannot be an isomorphism.

For R_e^1 requirements, swap the roles of $\mathcal{C}$ and $\mathcal{D}$ (including the subscripts for γ and δ). Then for all e , there is a stage where R_e^0 and R_e^1 are met. $\dashv$

LEMMA 2.8. *For all $i, j, k \in \mathbb{N}$ requirement $P_{i,j,k}$ is met.*

PROOF. Let $\mathcal{P}_i \geq_{pr} \mathcal{C}, \mathcal{D}$, we will show that $\mathcal{P}_i \geq_{pr} \mathcal{B}$ by defining a primitive recursive isomorphism $\eta : \mathcal{B} \rightarrow \mathcal{P}_i$. Note that by Lemma 2.5, $\mathcal{B} >_{pr} \mathcal{C}, \mathcal{D}$.

Since $\mathcal{P}_i \geq_{pr} \mathcal{C}, \mathcal{D}$, there are primitive recursive isomorphisms $p_j : \mathcal{C} \rightarrow \mathcal{P}_i$ and $p_k : \mathcal{D} \rightarrow \mathcal{P}_i$. At any stage, exactly one of $\mathcal{C}$ or $\mathcal{D}$ is copying $\mathcal{B}$. Then for all stages s and all $x \in \mathcal{B}[s]$, at least one of $\gamma_{\mathcal{C}}^{-1}(x)[s]$ or $\gamma_{\mathcal{D}}^{-1}(x)[s]$ is defined. If it is $\gamma_{\mathcal{C}}^{-1}(x)$, then define $\eta(x) = p_j(\gamma_{\mathcal{C}}^{-1}(x))$. Otherwise, define $\eta(x) = p_k(\gamma_{\mathcal{D}}^{-1}(x))$.

Recall that elements outside $\mathcal{L}_0$ appear in $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ at the same stage. Then $\gamma_{\mathcal{C}}^{-1}(x)[s] \downarrow$ for all $x \in \mathcal{B}[s]$ outside $\mathcal{L}_0$. By definition of η , $\eta(x) = p_j(\gamma_{\mathcal{C}}^{-1}(x))$ for such x . For all $x \in \mathcal{L}_0$, $p_j(\gamma_{\mathcal{C}}^{-1}(x)) = p_k(\gamma_{\mathcal{D}}^{-1}(x))$ since $\mathcal{L}_0$ is invariant in $\mathcal{L}$. Therefore η is a primitive recursive isomorphism. $\dashv$

LEMMA 2.9. *For all $i, j, k \in \mathbb{N}$ requirement $Q_{i,j,k}$ is met.*

PROOF. Let $\mathcal{P}_i \leq_{pr} \mathcal{C}, \mathcal{D}$, we will show that $\mathcal{P}_i \leq_{pr} \mathcal{A}$ by defining a primitive recursive isomorphism $v : \mathcal{P}_i \rightarrow \mathcal{A}$. Note that by Lemma 2.6, $\mathcal{A} <_{pr} \mathcal{C}, \mathcal{D}$.

Since $\mathcal{P}_i \leq_{pr} \mathcal{C}, \mathcal{D}$, there are primitive recursive isomorphisms $q_j : \mathcal{P}_i \rightarrow \mathcal{C}$ and $q_k : \mathcal{P}_i \rightarrow \mathcal{D}$. Consider the least stage s such that $q_j(x)[s] \downarrow$ and $q_k(x)[s] \downarrow$. Define $v(x) = \delta_{\mathcal{C}}^{-1}(q_j(x))$. Since $\delta_{\mathcal{C}}^{-1}$ is an isomorphism and q_j is by assumption an isomorphism, all that is left is to show that v is primitive recursive.

For all x such that $q_j(x)[s] \downarrow$ is outside $\mathcal{L}_0$, $\delta_{\mathcal{C}}^{-1}(q_j(x))[s] \downarrow$. Otherwise, since $\mathcal{L}_0$ is invariant, $\gamma_{\mathcal{C}}(q_j(x)) = \gamma_{\mathcal{D}}(q_k(x))$. By construction, only one of $\mathcal{C}$ and $\mathcal{D}$ is currently copying the new elements in $\mathcal{B}$. Therefore it follows that once $q_j(x)$ and $q_k(x)$ converge, the corresponding element must be in $\mathcal{A}$. Then for each x at the least stage s such that $q_j(x)[s] \downarrow$ and $q_k(x)[s] \downarrow$, $\delta_{\mathcal{C}}^{-1}(q_j(x))[s] \downarrow$. Therefore v is a primitive recursive isomorphism. $\dashv$

All requirements are met; functions $\gamma_{\mathcal{C}}, \gamma_{\mathcal{D}}, \delta_{\mathcal{C}}$, and $\delta_{\mathcal{D}}$ are primitive recursive isomorphisms by construction. Hence we have built $\mathcal{C}, \mathcal{D}$ as desired.

REMARK 2.10. Notice that $p = \gamma_{\mathcal{C}} \circ \delta_{\mathcal{C}} = \gamma_{\mathcal{D}} \circ \delta_{\mathcal{D}}$ by construction, this will come in useful in the next section.

2.3. Embedding the atomless Boolean algebra. Recall that we can embed the diamond between any $\mathcal{A} <_{pr} \mathcal{B}$ that reveal elements outside $\mathcal{L}_0$ at the same stage. By construction, $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ reveal elements outside of $\mathcal{L}_0$ at the same stage. This means the diamond construction naturally iterates between $\mathcal{A}$ and $\mathcal{C}$, $\mathcal{C}$ and $\mathcal{B}$, etc. By applying this process, we obtain *generators* for an embedding of the atomless Boolean algebra between $\mathcal{A}$ and $\mathcal{B}$. In this section, we work towards proving the following theorem.

THEOREM 2.11. *Let $\mathcal{L}$ be a linear order such that there are $\mathcal{L}_{-1}, \mathcal{L}_0$, and $\mathcal{L}_1$ with the following properties.*

- $\mathcal{L}_{-1} + \mathcal{L}_0 + \mathcal{L}_1 = \mathcal{L}$.
- $\mathcal{L}_0$ is infinite and invariant in $\mathcal{L}$.
- $\mathcal{L}_{-1}, \mathcal{L}_0$, and $\mathcal{L}_1$ are punctual linear orders.

Then for any punctual presentations $\mathcal{A}, \mathcal{B}$ of $\mathcal{L}$ such that $\mathcal{A} <_{pr} \mathcal{B}$ and $\mathcal{A}$ and $\mathcal{B}$ reveal elements outside $\mathcal{L}_0$ at the same stage, we can embed the atomless Boolean algebra with $\mathcal{B} = 1$ and $\mathcal{A} = 0$.

A *generator* is a presentation that is built by some iteration of the diamond construction. We define the *complexity of a generator* $\mathcal{E}$ to be the number of times the diamond construction has been iterated in order to obtain $\mathcal{E}$. For example, first we obtain $\mathcal{E}_0, \mathcal{E}_1$ by iterating the diamond construction once between $\mathcal{A}$ and $\mathcal{B}$, then these presentations have complexity 1. We then iterate the diamond construction again between $\mathcal{E}_0$ and $\mathcal{A}$, between $\mathcal{E}_0$ and $\mathcal{B}$, between $\mathcal{E}_1$ and $\mathcal{A}$, and between $\mathcal{E}_1$ and $\mathcal{B}$; these presentations that we obtain have complexity 2. We consider the lattice obtained by finite combinations of meets and joins of generators. We will prove later that this indeed gives us a Boolean algebra with $\mathcal{A} = 0$ and $\mathcal{B} = 1$.

For this section, we use the following notation for the isomorphisms that are in play. Our notation naturally extends the notation from the previous section. We use $\gamma_{\mathcal{E}} : \mathcal{E} \rightarrow \mathcal{B}, \delta_{\mathcal{E}} : \mathcal{A} \rightarrow \mathcal{E}$, while $\eta_{\mathcal{E} \rightarrow \mathcal{G}} : \mathcal{E} \rightarrow \mathcal{G}$ is used where neither $\mathcal{E}$ nor $\mathcal{G}$ are $\mathcal{A}$ or $\mathcal{B}$. Recall that p is the primitive recursive isomorphism from $\mathcal{A}$ to $\mathcal{B}$.

LEMMA 2.12. *Let $\mathcal{E}$ be a generator. Then there are primitive recursive isomorphisms $\delta_{\mathcal{E}} : \mathcal{A} \rightarrow \mathcal{E}$ and $\gamma_{\mathcal{E}} : \mathcal{E} \rightarrow \mathcal{B}$, moreover $\gamma_{\mathcal{E}} \circ \delta_{\mathcal{E}} = p$.*

PROOF. We prove this by induction on the complexity of the generator. The base case is where $\mathcal{E}$ has complexity 1, then the lemma follows immediately from Lemmas 2.5 and 2.6.

Now suppose the lemma holds for complexity n , we will prove for complexity $n + 1$. $\mathcal{E}$ is a generator and has complexity $n + 1$, so $\mathcal{E}$ is constructed by iterating the diamond construction between some $\mathcal{G}$ and $\mathcal{H}$ that have complexity at most n . Without loss of generality, assume $\mathcal{G} >_{pr} \mathcal{H}$. By induction there are primitive recursive isomorphisms $\gamma_{\mathcal{G}}, \gamma_{\mathcal{H}}, \delta_{\mathcal{G}}$, and $\delta_{\mathcal{H}}$. By construction, $\eta_{\mathcal{E} \rightarrow \mathcal{G}}$ and $\eta_{\mathcal{H} \rightarrow \mathcal{E}}$ are primitive recursive isomorphisms. Then it follows that $\gamma_{\mathcal{E}} = \gamma_{\mathcal{G}} \circ \eta_{\mathcal{E} \rightarrow \mathcal{G}}$ and $\delta_{\mathcal{E}} = \eta_{\mathcal{H} \rightarrow \mathcal{E}} \circ \delta_{\mathcal{H}}$ are primitive recursive isomorphisms.

By induction, $\gamma_{\mathcal{G}} \circ \delta_{\mathcal{G}} = p$ and $\gamma_{\mathcal{H}} \circ \delta_{\mathcal{H}} = p$. By construction, $\eta_{\mathcal{E} \rightarrow \mathcal{G}} \circ \eta_{\mathcal{H} \rightarrow \mathcal{E}}$ is a primitive recursive isomorphism from $\mathcal{H}$ to $\mathcal{G}$. Then it follows that $p = \gamma_{\mathcal{G}} \circ \eta_{\mathcal{E} \rightarrow \mathcal{G}} \circ \eta_{\mathcal{H} \rightarrow \mathcal{E}} \circ \delta_{\mathcal{H}} = \gamma_{\mathcal{E}} \circ \delta_{\mathcal{E}}$. $\dashv$

LEMMA 2.13. *Let $\mathcal{E}, \mathcal{F}$ be presentations such that there are primitive recursive isomorphisms $\delta_{\mathcal{E}}, \gamma_{\mathcal{E}}, \delta_{\mathcal{F}}$, and $\gamma_{\mathcal{F}}$, and $\gamma_{\mathcal{E}} \circ \delta_{\mathcal{E}} = \gamma_{\mathcal{F}} \circ \delta_{\mathcal{F}} = p$. Then there is a punctual presentation $\mathcal{G}$ of $\mathcal{L}$ such that $\mathcal{G} = \sup(\mathcal{E}, \mathcal{F})$ and there are primitive recursive isomorphisms $\delta_{\mathcal{G}} : \mathcal{A} \rightarrow \mathcal{G}$ and $\gamma_{\mathcal{G}} : \mathcal{G} \rightarrow \mathcal{B}$, moreover $\gamma_{\mathcal{G}} \circ \delta_{\mathcal{G}} = p$.*

PROOF. We define a punctual presentation $\mathcal{G}$ that is exactly the elements that are in either $\mathcal{E}$ or $\mathcal{F}$. Below we give the formal definition, including the isomorphisms $\gamma_{\mathcal{G}}, \delta_{\mathcal{G}}, \eta_{\mathcal{E} \rightarrow \mathcal{G}}$, and $\eta_{\mathcal{F} \rightarrow \mathcal{G}}$.

Consider $\{\gamma_{\mathcal{E}}(i) : i < s\} \cup \{\gamma_{\mathcal{F}}(i) : i < s\}$ as a substructure of $\mathcal{B}$. We will denote this substructure as $\mathcal{B}_{\mathcal{E} \cup \mathcal{F}}[s]$. We build a punctual presentation $\mathcal{G}$ so that $\mathcal{G} \cong \mathcal{B}_{\mathcal{E} \cup \mathcal{F}}[s]$. This means that at each stage for each $x \in \mathcal{B}_{\mathcal{E} \cup \mathcal{F}}[s + 1] \setminus \mathcal{B}_{\mathcal{E} \cup \mathcal{F}}[s]$ we consider the least y with $\gamma_{\mathcal{G}}(y)[s] \uparrow$ and define $\gamma_{\mathcal{G}}(y) = x$. Define $a <_{\mathcal{G}} b$ if and only if $\gamma_{\mathcal{G}}(a) <_{\mathcal{B}} \gamma_{\mathcal{G}}(b)$. $\mathcal{E}$ and $\mathcal{F}$ are punctual and $\gamma_{\mathcal{G}}$ is a primitive recursive isomorphism, therefore $\mathcal{G}$ is punctual.

For x with $\gamma_{\mathcal{E}}(x)[s] \downarrow$, we define $\eta_{\mathcal{E} \rightarrow \mathcal{G}} = \gamma_{\mathcal{G}}^{-1}(\gamma_{\mathcal{E}}(x))$. For x with $\gamma_{\mathcal{F}}(x)[s] \downarrow$, we define $\eta_{\mathcal{F} \rightarrow \mathcal{G}} = \gamma_{\mathcal{G}}^{-1}(\gamma_{\mathcal{F}}(x))$. These are primitive recursive isomorphisms because at the stage that $\gamma_{\mathcal{F}}(x)[s] \downarrow$, $\gamma_{\mathcal{F}}(x) \in \mathcal{B}_{\mathcal{E} \cup \mathcal{F}}[s]$ by definition, therefore at stage s , $\gamma_{\mathcal{G}}^{-1}(\gamma_{\mathcal{F}}(x)) \downarrow$. The same argument shows that $\eta_{\mathcal{E} \rightarrow \mathcal{G}}$ is primitive recursive.

It follows that $\delta_{\mathcal{G}}(x) = \eta_{\mathcal{E} \rightarrow \mathcal{G}}(\delta_{\mathcal{E}}(x))$ is a primitive recursive isomorphism. Since $\gamma_{\mathcal{E}} \circ \delta_{\mathcal{E}} = \gamma_{\mathcal{F}} \circ \delta_{\mathcal{F}} = p$, it follows that $\gamma_{\mathcal{G}} \circ \delta_{\mathcal{G}} = p$.

Finally, it is left to prove that $\mathcal{G} = \sup(\mathcal{E}, \mathcal{F})$. Suppose there is $\mathcal{P}_i \geq_{pr} \mathcal{E}, \mathcal{F}$ witnessed by $p_j : \mathcal{E} \rightarrow \mathcal{P}_i$ and $p_k : \mathcal{F} \rightarrow \mathcal{P}_i$. We define $\eta : \mathcal{G} \rightarrow \mathcal{P}_i$ as follows. For each x , at the least stage s where $\gamma_{\mathcal{G}}(x)[s] \downarrow$, define $\eta(x) = p_j(\eta_{\mathcal{E} \rightarrow \mathcal{G}}^{-1}(x))$ if $\eta_{\mathcal{E} \rightarrow \mathcal{G}}^{-1}(x)[s] \downarrow$, otherwise define $\eta(x) = p_k(\eta_{\mathcal{F} \rightarrow \mathcal{G}}^{-1}(x))$.

By definition of $\mathcal{G}$, at the least stage s where $\gamma_{\mathcal{G}}(x)[s] \downarrow$, at least one of $\eta_{\mathcal{E} \rightarrow \mathcal{G}}^{-1}(x)[s]$ or $\eta_{\mathcal{F} \rightarrow \mathcal{G}}^{-1}(x)[s]$ is defined. Therefore η is primitive recursive.

Recall that for all $x \notin \mathcal{L}_0$, x is added to all presentations at the same stage. Therefore $\eta(x) = p_j(\eta_{\mathcal{E} \rightarrow \mathcal{G}}^{-1}(x))$ for all such x . For $x \in \mathcal{L}_0$, $p_j(\eta_{\mathcal{E} \rightarrow \mathcal{G}}^{-1}(x)) = p_k(\eta_{\mathcal{F} \rightarrow \mathcal{G}}^{-1}(x))$. Therefore η is an isomorphism since $p_j, p_k, \eta_{\mathcal{E} \rightarrow \mathcal{G}}, \eta_{\mathcal{F} \rightarrow \mathcal{G}}$ are all isomorphisms.

Therefore $\mathcal{P}_i \geq_{pr} \mathcal{G}$. Hence $\mathcal{G} = \sup(\mathcal{E}, \mathcal{F})$ as desired. $\dashv$

LEMMA 2.14. *Let $\mathcal{E}, \mathcal{F}$ be punctual presentations such that there are primitive recursive isomorphisms $\delta_{\mathcal{E}} : \mathcal{A} \rightarrow \mathcal{E}$, $\gamma_{\mathcal{E}} : \mathcal{E} \rightarrow \mathcal{B}$, $\delta_{\mathcal{F}} : \mathcal{A} \rightarrow \mathcal{F}$ and $\gamma_{\mathcal{F}} : \mathcal{F} \rightarrow \mathcal{B}$, and $\gamma_{\mathcal{E}} \circ \delta_{\mathcal{E}} = \gamma_{\mathcal{F}} \circ \delta_{\mathcal{F}} = p$. Then there is a punctual presentation $\mathcal{H}$ of $\mathcal{L}$ such that $\mathcal{H} = \inf(\mathcal{E}, \mathcal{F})$ and there are primitive recursive isomorphisms $\delta_{\mathcal{H}} : \mathcal{A} \rightarrow \mathcal{H}$ and $\gamma_{\mathcal{H}} : \mathcal{H} \rightarrow \mathcal{B}$, moreover $\gamma_{\mathcal{H}} \circ \delta_{\mathcal{H}} = p$.*

PROOF. We define a punctual presentation $\mathcal{H}$ that is exactly the elements that are in both $\mathcal{E}$ and $\mathcal{F}$. Below we give the formal definition, including the isomorphisms $\gamma_{\mathcal{H}}, \delta_{\mathcal{H}}, \eta_{\mathcal{H} \rightarrow \mathcal{E}}$, and $\eta_{\mathcal{H} \rightarrow \mathcal{F}}$.

Consider $\{\gamma_{\mathcal{E}}(i) : i < s\} \cap \{\gamma_{\mathcal{F}}(i) : i < s\}$ as a substructure of $\mathcal{B}$. We will denote this substructure as $\mathcal{B}_{\mathcal{E} \cap \mathcal{F}}[s]$. We build a punctual presentation $\mathcal{H}$ so that $\mathcal{H} \cong \mathcal{B}_{\mathcal{E} \cap \mathcal{F}}[s]$. This means that at each stage for each $x \in \mathcal{B}_{\mathcal{E} \cap \mathcal{F}}[s+1] \setminus \mathcal{B}_{\mathcal{E} \cap \mathcal{F}}[s]$ we consider the least y with $\gamma_{\mathcal{H}}(y)[s] \uparrow$ and define $\gamma_{\mathcal{H}}(y) = x$. Define $a <_{\mathcal{H}} b$ if and only if $\gamma_{\mathcal{H}}(a) <_{\mathcal{B}} \gamma_{\mathcal{H}}(b)$.

Now we define $\delta_{\mathcal{H}}(x)$. Let s be the least stage such that $p(x)[s] \downarrow$. By stage s , $\gamma_{\mathcal{E}}^{-1}(p(x))[s] \downarrow$, $\gamma_{\mathcal{F}}^{-1}(p(x))[s] \downarrow$ and by definition of $\mathcal{H}$, an element y has been added to $\mathcal{H}[s]$ such that $\gamma_{\mathcal{H}}(y) = p(x)$. Then at stage s , we define $\delta_{\mathcal{H}}(x) = y$. Hence, $\delta_{\mathcal{H}}$ is primitive recursive. It follows that $\mathcal{H}$ is punctual and $\gamma_{\mathcal{H}} \circ \delta_{\mathcal{H}} = p$ by definition.

Define $\eta_{\mathcal{H} \rightarrow \mathcal{E}}(x) = \gamma_{\mathcal{E}}^{-1}(\gamma_{\mathcal{H}}(x))$ and $\eta_{\mathcal{H} \rightarrow \mathcal{F}}(x) = \gamma_{\mathcal{F}}^{-1}(\gamma_{\mathcal{H}}(x))$. These are isomorphisms by composition and are primitive recursive because at the stage s that $\gamma_{\mathcal{H}}(x)[s] \downarrow$, $\gamma_{\mathcal{H}}(x) \in \mathcal{B}_{\mathcal{E} \cap \mathcal{F}}[s]$ by definition, therefore $\gamma_{\mathcal{E}}^{-1}(\gamma_{\mathcal{H}}(x))[s] \downarrow$ and $\gamma_{\mathcal{F}}^{-1}(\gamma_{\mathcal{H}}(x))[s] \downarrow$.

Now we show that $\mathcal{H} = \inf(\mathcal{E}, \mathcal{F})$. Suppose there is $\mathcal{P}_i \leq_{pr} \mathcal{E}, \mathcal{F}$ witnessed by $q_j : \mathcal{P}_i \rightarrow \mathcal{E}$ and $q_k : \mathcal{P}_i \rightarrow \mathcal{F}$. We define $\eta : \mathcal{P}_i \rightarrow \mathcal{H}$ as follows. For each x , at the least stage s such that $q_j(x)[s] \downarrow$ and $q_k(x)[s] \downarrow$, define $\eta(x) = \eta_{\mathcal{H} \rightarrow \mathcal{E}}^{-1}(q_j(x))$. Now we show that η is primitive recursive. Recall that elements outside L_0 appear at the same time in every structure. So for x outside L_0 , $q_j(x)$ is also outside L_0 and so $\eta_{\mathcal{H} \rightarrow \mathcal{E}}^{-1}(q_j(x))$ converges once $q_j(x)$ converges. For x in L_0 , $\gamma_{\mathcal{E}}(q_j(x)) = \gamma_{\mathcal{F}}(q_k(x)) \in \mathcal{B}_{\mathcal{E} \cap \mathcal{F}}[s]$. Therefore, the corresponding element must be in $\mathcal{H}[s]$ and so $\eta_{\mathcal{H} \rightarrow \mathcal{E}}^{-1}(q_j(x))[s] \downarrow$. Thus η is a primitive recursive isomorphism.

Hence $\mathcal{P}_i \leq_{pr} \mathcal{H}$ and $\mathcal{H} = \inf(\mathcal{E}, \mathcal{F})$ as desired. $\dashv$

With the above lemmas, we are now ready to prove Theorem 2.11.

PROOF OF THEOREM 2.11. Apply Theorem 2.2 between $\mathcal{A}$ and $\mathcal{B}$. Repeat the application of Theorem 2.2 between every pair of presentations built so far with the property that $\mathcal{E} <_{pr} \mathcal{F}$. We call the presentations built in this way the *generators*. We consider the structure obtained by taking finite combinations of meets and joins of these generators. We show that this gives us the atomless Boolean algebra by showing that complements, meets, and joins of generators exist as well as showing that the lattice embedding is distributive.

Existence of meets and joins. Follows from Lemmas 2.12–2.14.

Existence of complements. If complements of the generators exist then it follows that every element has a complement.

We use induction on the complexity of the generator. The base case is generators of complexity 1. In this case, the complement of a generator $\mathcal{G}$ with complexity 1 is the other generator created between $\mathcal{A}$ and $\mathcal{B}$. Let $\mathcal{G}$ be a generator with complexity $n > 1$. Then $\mathcal{G}$ was built between some other generators $\mathcal{E} <_{pr} \mathcal{F}$ of lower complexity. Let $\mathcal{H}$ be the other generator that was built between $\mathcal{E}$ and $\mathcal{F}$. By induction, $\mathcal{E}$ and $\mathcal{F}$ have complements, $\mathcal{E}^c$ and $\mathcal{F}^c$, respectively. Then we show that $(\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c$ is the complement of $\mathcal{G}$.

To do this, we need to show that $\mathcal{G} \vee (\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c \cong_{pr} \mathcal{B}$ and $\mathcal{G} \wedge ((\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c) \cong_{pr} \mathcal{A}$. By construction, at any stage, generators are either copying $\mathcal{A}$ or copying $\mathcal{B}$. Recall that in Lemmas 2.13 and 2.14, the join and meet of two generators $\mathcal{C}$ and $\mathcal{D}$ exist. It follows that the join (meet) of two presentations copies $\mathcal{A}$ ($\mathcal{B}$) only if both presentations copy $\mathcal{A}$ ($\mathcal{B}$) and otherwise copies $\mathcal{B}$ ($\mathcal{A}$). Also notice that given some presentation, the complement of this presentation has the property that, at any stage, if one copies $\mathcal{A}$, the other must be copying $\mathcal{B}$.

First, we show that at any stage $\mathcal{G} \wedge ((\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c)$ copies $\mathcal{A}$. Suppose $\mathcal{G}$ is currently copying $\mathcal{B}$, then we show that $(\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c$ is copying $\mathcal{A}$. Since $\mathcal{G}$ is copying $\mathcal{B}$ and $\mathcal{G} <_{pr} \mathcal{F}$, $\mathcal{F}$ also copies $\mathcal{B}$ and $\mathcal{F}^c$ is copying $\mathcal{A}$. $\mathcal{E} = \mathcal{G} \wedge \mathcal{H}$ and $\mathcal{G}$ copies $\mathcal{B}$, then $\mathcal{E}$ and $\mathcal{H}$ are either both copying $\mathcal{A}$ or both copying $\mathcal{B}$. Therefore, $\mathcal{G} \wedge ((\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c)$ copies $\mathcal{A}$ as desired.

Finally we show that at any stage $\mathcal{G} \vee (\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c$ copies $\mathcal{B}$. Suppose $\mathcal{G}$ and $\mathcal{F}^c$ are both copying $\mathcal{A}$ and we will prove that $\mathcal{H} \wedge \mathcal{E}^c$ is copying $\mathcal{B}$. Since $\mathcal{F}^c$ is copying $\mathcal{A}$, $\mathcal{F}$ must be copying $\mathcal{B}$. Notice if $\mathcal{E}$ is copying $\mathcal{B}$, then $\mathcal{G}$ must be copying $\mathcal{B}$ as well, which is a contradiction. So $\mathcal{E}$ is copying $\mathcal{A}$ and $\mathcal{E}^c$ is copying $\mathcal{B}$. Since $\mathcal{F} = \mathcal{G} \vee \mathcal{H}$, it must be that $\mathcal{H}$ is also copying $\mathcal{B}$. Therefore, $\mathcal{G} \vee (\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c$ copies $\mathcal{B}$.

Hence $(\mathcal{H} \wedge \mathcal{E}^c) \vee \mathcal{F}^c$ is the complement of $\mathcal{G}$ and therefore all generators have a complement.

Distributivity. To show a lattice is distributive, it is sufficient to show that it is distributive as a join-semilattice [24]. Therefore, it is sufficient to show that for any presentations $\mathcal{E}, \mathcal{F}$, and $\mathcal{G}$ in our Boolean algebra, if $\mathcal{G} \leq_{pr} \sup(\mathcal{E}, \mathcal{F})$, there are $\mathcal{E}'$ and $\mathcal{F}'$ such that $\mathcal{G} = \sup(\mathcal{E}', \mathcal{F}')$, $\mathcal{E}' \leq_{pr} \mathcal{E}$ and $\mathcal{F}' \leq_{pr} \mathcal{F}$.

Let $\mathcal{E}' = \inf(\mathcal{G}, \mathcal{E})$ and $\mathcal{F}' = \inf(\mathcal{G}, \mathcal{F})$. We show that $\mathcal{G} = \sup(\mathcal{E}', \mathcal{F}')$. Let $\mathcal{K} \geq_{pr} \mathcal{E}', \mathcal{F}'$. We will show that $\mathcal{K} \geq_{pr} \mathcal{G}$ by defining $\alpha : \mathcal{G} \rightarrow \mathcal{K}$. For convenience, let $\sup(\mathcal{E}, \mathcal{F}) = \mathcal{M}$.

At the least stage t such that $\eta_{\mathcal{G} \rightarrow \mathcal{M}}(x)[t] \downarrow$, define $\alpha(x)[t] = \eta_{\mathcal{E}' \rightarrow \mathcal{K}}(\eta_{\mathcal{E}' \rightarrow \mathcal{G}}^{-1}(x))$ if $\eta_{\mathcal{E}' \rightarrow \mathcal{G}}^{-1}(x)[t] \downarrow$, otherwise define $\alpha(x)[t] = \eta_{\mathcal{F}' \rightarrow \mathcal{K}}(\eta_{\mathcal{F}' \rightarrow \mathcal{G}}^{-1}(x))$.

Recall that elements outside $\mathcal{L}_0$ appear at the same time in $\mathcal{E}', \mathcal{F}', \mathcal{G}$, then $\eta_{\mathcal{E}' \rightarrow \mathcal{G}}^{-1}(x)$ and $\eta_{\mathcal{F}' \rightarrow \mathcal{G}}^{-1}(x)$ converge at stage t for elements outside $\mathcal{L}_0$. Therefore

$\alpha(x) = \eta_{\mathcal{E}' \rightarrow \mathcal{K}}(\eta_{\mathcal{E}' \rightarrow \mathcal{G}}^{-1}(x))$ for all $x \notin L_0$. If x is in $\mathcal{L}_0$, then $\eta_{\mathcal{E}' \rightarrow \mathcal{K}}(\eta_{\mathcal{E}' \rightarrow \mathcal{G}}^{-1}(x)) = \eta_{\mathcal{F}' \rightarrow \mathcal{K}}(\eta_{\mathcal{F}' \rightarrow \mathcal{G}}^{-1}(x))$. Therefore, α is an isomorphism.

Now we prove that at stage t , then either $\eta_{\mathcal{E}' \rightarrow \mathcal{G}}^{-1}(x)[t] \downarrow$ or $\eta_{\mathcal{F}' \rightarrow \mathcal{G}}^{-1}(x)[t] \downarrow$ for any $x \in \mathcal{G}[t]$. Recall that if $\eta_{\mathcal{G} \rightarrow \mathcal{M}}(x)[t] \downarrow$, then by definition of supremum, $\eta_{\mathcal{E} \rightarrow \mathcal{M}}(\eta_{\mathcal{G} \rightarrow \mathcal{M}}(x)[t] \downarrow)$ or $\eta_{\mathcal{F} \rightarrow \mathcal{M}}(\eta_{\mathcal{G} \rightarrow \mathcal{M}}(x)[t] \downarrow)$. By definition of $\mathcal{E}'$ and $\mathcal{F}'$, it follows that either $\eta_{\mathcal{E}' \rightarrow \mathcal{G}}^{-1}(x)[t] \downarrow$ or $\eta_{\mathcal{F}' \rightarrow \mathcal{G}}^{-1}(x)[t] \downarrow$. $\dashv$

§3. The dense linear order. In this section, we prove that we can embed the diamond into the punctual degrees of the dense linear order.

3.1. Embedding the diamond.

THEOREM 3.1 (Diamond in $PR(\mathbb{Q})$). *There exist punctual presentations $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ of $(\mathbb{Q}, <)$ such that $\mathcal{C}$ and $\mathcal{D}$ are incomparable under $\leq_{pr}$, $\mathcal{B} = \sup(\mathcal{C}, \mathcal{D})$, and $\mathcal{A} = \inf(\mathcal{C}, \mathcal{D})$.*

We build structures $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ so that they meet the following requirements¹:

$R_e^0 : p_e : \mathcal{C} \rightarrow \mathcal{D}$ is not an isomorphism,

$R_e^1 : p_e : \mathcal{D} \rightarrow \mathcal{C}$ is not an isomorphism,

$P_{i,j,k} : \text{if } p_j : \mathcal{C} \rightarrow \mathcal{P}_i, p_k : \mathcal{D} \rightarrow \mathcal{P}_i \text{ are p.r. isomorphisms, then } \mathcal{B} \leq_{pr} \mathcal{P}_i,$

$Q_{i,j,k} : \text{if } q_j : \mathcal{P}_i \rightarrow \mathcal{C}, q_k : \mathcal{P}_i \rightarrow \mathcal{D} \text{ are p.r. isomorphisms, then } \mathcal{P}_i \leq_{pr} \mathcal{A},$

where $\langle p_e \rangle_{e \in \mathbb{N}}$ and $\langle \mathcal{P}_i \rangle_{i \in \mathbb{N}}$ are effective enumerations of all primitive recursive functions and punctual presentations, respectively. We also build primitive recursive isomorphisms $\gamma_{\mathcal{C}} : \mathcal{C} \rightarrow \mathcal{B}, \gamma_{\mathcal{D}} : \mathcal{D} \rightarrow \mathcal{B}, \delta_{\mathcal{C}} : \mathcal{A} \rightarrow \mathcal{C}, \delta_{\mathcal{D}} : \mathcal{A} \rightarrow \mathcal{D}$.

The idea for this construction is that $\mathcal{C}$ always grows to the right, while $\mathcal{D}$ always grows to the left. $\mathcal{B}$ adds elements that are in $\mathcal{C}$ or $\mathcal{D}$, while $\mathcal{A}$ adds elements only once they are in both $\mathcal{C}$ and $\mathcal{D}$. To meet P requirements, we build an isomorphism from $\mathcal{B}$ to $\mathcal{P}_i$ by copying p_j to the left and p_k to the right. The *switch* between p_j and p_k is simple when they line up. But in the case that they do not line up, which is entirely possible in the case of the dense linear order, we use a *special* interval $[z, 1]$ which will *patch* the gap between the isomorphisms p_j and p_k . In the case of Q requirements, we can diagonalise to force q_j and q_k to line up. We highlight that these special elements z and 1 are the same elements for all requirements.

3.2. Construction. Order all R, P , and Q requirements in an effective list of order type ω . At stage 0, let $|A[0]| = |B[0]| = |C[0]| = |D[0]| = 2$. Call these elements 1 and z with $1 > z$. These elements are special elements used for our construction. Define $\gamma_{\mathcal{C}}(z) = \delta_{\mathcal{C}}(z) = \gamma_{\mathcal{D}}(z) = \delta_{\mathcal{D}}(z) = z$, similarly for 1 . This means that z and 1 are the *same* elements in $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$. For each stage s , let z_s be the element such that it is the greatest element $z < 1$ at stage $s - 1$. In the construction, $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ are the same in the interval $[z, 1]$, therefore z_s is the *same* element in all structures.

¹Note that these requirements are the same as in the invariant linear order case, they are restated here for convenience.

We note that the follow conventions are the same as used in Section 2.1 but is restated here for convenience. For structures we are building $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$, we consider $\mathcal{E}[s]$ to be the presentation we have built so far at stage s . Throughout the construction, we will *add a new element to $\mathcal{E}$* , this means we add the $<_{\mathbb{N}}$ -least element $x \in \mathbb{N}$ that is not yet in $\mathcal{E}[s]$ and we define $<_{\mathcal{E}}$ as described in the construction. In this construction, we will determine the punctuality of $\mathcal{E}$ by adding the element s to $\mathcal{E}[s]$ by the end of stage s . To ensure that the isomorphisms are primitive recursive, we define them on all elements in $\mathcal{E}[s]$ at stage s .

We will use the macro *grow $\mathcal{L}$ to the right* (at stage s) to mean add one new element x to $\mathcal{L}$ so that $x > y$ for all $y \in \mathcal{L}[s]$. Similarly, we will use the macro *grow $\mathcal{L}$ to the left* to mean add one new element x to $\mathcal{L}$ so that $x < y$ for all $y \in \mathcal{L}[s]$. We will use *grow the interval $[z_s, 1]$ in $\mathcal{L}$* to mean, add $|\mathcal{B}[s - 1]|$ many new elements between z_s and 1 to $\mathcal{L}[s]$.

At stage 0, no requirement requires attention. R and Q requirements will require attention at stage s as follows:

- *Requirement R_e^0 requires attention at stage s* if it has not been met and there is some $c \in \mathcal{C}[s]$ such that $p_e(c) = z$.
- *Requirement R_e^1 requires attention at stage s* if it has not been met and there is some $d \in \mathcal{D}[s]$ such that $p_e(d) = 1$.
- *Requirement $Q_{i,j,k}$ requires attention at stage s* if it has not been considered yet and there are $a, b \in \mathcal{P}_i[s]$ such that $q_j(a) = q_k(b) = 1$.

Every requirement after being considered, never requires attention again. At stage s wait until there is a requirement that requires attention, while we wait, grow the interval $[z_s, 1]$ in all structures. Note that P requirements never require attention; we will show that they are met in the verification.

Requirement $Q_{i,j,k}$. Let a, b be the witnesses of $q_j(a) = q_k(b) = 1$. If $a = b$ then declare $Q_{i,j,k}$ good. Otherwise, grow only the interval $(z_t, 1)$ in all structures. Wait for a stage t where either $\mathcal{P}_i$ shows it is not a linear order, or either q_j or q_k shows it is not an isomorphism. Declare that $Q_{i,j,k}$ no longer requires attention, then move to the recovery stage.

Requirement R_e^0 . Let c be the witness of $p_e(c) = z$. At every stage t while we work on this requirement, grow $\mathcal{C}$ and $\mathcal{B}$ to the left and right, grow $\mathcal{D}$ and $\mathcal{A}$ to the right, and grow the interval $(z_t, 1)$ in all structures. Compute $p_e(x)$ for all $x \in \mathcal{C}$ to the left of c . Eventually, there must be $x, y \in \mathcal{C}$ such that $x <_{\mathcal{C}} y <_{\mathcal{C}} c$ and $p_e(x) \not\prec_{\mathcal{D}} p_e(y)$. Declare that R_e^0 is met and no longer requires attention, then move to the recovery stage.

Requirement R_e^1 . Let d be the witness of $p_e(d) = 1$. At every stage t while we work on this requirement, grow $\mathcal{D}$ and $\mathcal{B}$ to the left and right, grow $\mathcal{C}$ and $\mathcal{A}$ to the left, and grow the interval $(z_t, 1)$ in all structures. Compute $p_e(x)$ for all $x \in \mathcal{D}$ to the right of d . Eventually, there must be an $x, y \in \mathcal{D}$ such that $d <_{\mathcal{D}} x <_{\mathcal{D}} y$ and $p_e(x) \not\prec_{\mathcal{C}} p_e(y)$. Declare that R_e^1 no longer requires attention, then move to the recovery stage.

Recovery stage. At a recovery stage s , we ensure that all of our structures have the same number of elements. Ensure that $\gamma_{\mathcal{C}}, \gamma_{\mathcal{D}}, \delta_{\mathcal{C}}, \delta_{\mathcal{D}}$ are all surjective by adding any required elements. In all structures, first grow $(z_s, 1)$, then add an element between any two adjacent elements.

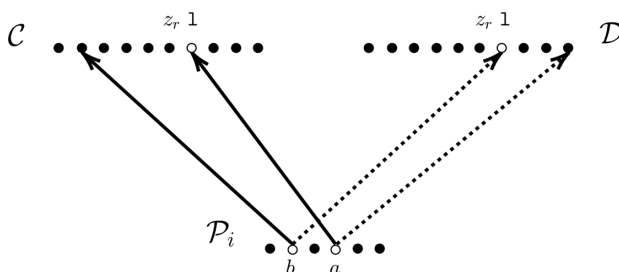


FIGURE 3. A snapshot of a Q requirement that requires attention and is not declared to be good. The dashed arrows are q_k and solid arrows are q_j .

3.2.1. *Verification.* Now we show that all requirements are met.

LEMMA 3.2. *For all $e \in \mathbb{N}$, if requirement R_e^0 is given attention, then R_e^0 is met.*

PROOF. Let c be such that $p_e(c) = z$. While we working on this requirement we do not grow $\mathcal{D}$ to the left of z . Hence, there are only some finite number m elements to the left of z in $\mathcal{D}$. Since we grow $\mathcal{C}$ to the left, it must be that eventually we see some $x <_C y <_C c$ but $p_e(x) \not\leq_{\mathcal{D}} p_e(y)$, hence we have met R_e^0 . $\dashv$

LEMMA 3.3. *For all $e \in \mathbb{N}$, if requirement R_e^1 is given attention then R_e^1 is met.*

PROOF. Let d be such that $p_e(d) = 1$. While we are working on this requirement we do not grow $\mathcal{C}$ to the right of 1 . Hence there are only some finite number elements to the right of 1 in $\mathcal{C}$. Since we grow $\mathcal{D}$ to the right, it must be that eventually we see some $x, y \in \mathcal{D}$ such that $d <_{\mathcal{D}} x <_{\mathcal{D}} y$ and $p_e(x) \not\leq_{\mathcal{C}} p_e(y)$, hence we have met R_e^1 . $\dashv$

LEMMA 3.4. *For all $i, j, k \in \mathbb{N}$, if requirement $Q_{i,j,k}$ is given attention and not declared to be good, then requirement $Q_{i,j,k}$ is met.*

PROOF. If at any stage $\mathcal{P}_i$ shows it is not a linear order, then the requirement is met. So we suppose that $\mathcal{P}_i$ is a linear order. For this requirement to require attention, there must be $a, b \in \mathcal{P}_i$ such that $q_j(a) = q_k(b) = 1$. Since this requirement is not declared good by assumption, we have $a \neq b$. Either $q_j(b) < q_j(a)$ or $q_k(a) < q_k(b)$, otherwise, either q_j or q_k would not be an isomorphism and the requirement is met. Consider the former case. See Figure 3 for a diagram of this case. Wait for $q_j(b)$ to converge at some stage r . Then $q_j(b) \leq z_r$. From now on, while we are working on this requirement, our structures grow only the interval $[z_r, 1]$. Suppose that there are less than n elements in the structures that we build at stage r . Wait for $2n$ new elements to appear in $\mathcal{P}_i$ and for q_j and q_k to converge on them. If at least n of these new elements are to the left of b , we will see a disagreement since there are less than n elements to the left of $q_j(b)$. Otherwise, there are at least n new elements to the right of b , and we will see a disagreement, since there are less than n elements to the right of $q_k(b)$. Therefore, there must be a stage where q_j or q_k shows that it is not an isomorphism, and at this stage, the requirement is met.

In the case that $q_j(b) > q_j(a)$, then $q_k(a) < q_k(b)$. Then by the same argument as above $q_k(a) \leq z_r$ and elements are only added between $q_k(a)$ and $q_j(a)$ while diagonalising and there must be a stage where this requirement is met. $\dashv$

LEMMA 3.5. *For all $i, j, k \in \mathbb{N}$, if requirement $Q_{i,j,k}$ is declared to be good, then requirement $Q_{i,j,k}$ is met.*

PROOF. $Q_{i,j,k}$ is met if q_j or q_k are not isomorphisms, so suppose q_j and q_k are isomorphisms. $Q_{i,j,k}$ is good, so there is $a \in \mathcal{P}_i$ such that $q_j(a) = q_k(a) = 1$. Define $\theta : \mathcal{P}_i \rightarrow \mathcal{A}$ as follows: for $x \leq a$ define $\theta(x) = \delta_{\mathcal{D}}^{-1}(q_k(x))$ and for $x > a$ define $\theta(x) = \delta_{\mathcal{C}}^{-1}(q_j(x))$. θ is an isomorphism because $\delta_{\mathcal{D}}^{-1}(q_k)$ and $\delta_{\mathcal{C}}^{-1} \circ q_j$ are isomorphisms by assumption and $q_j(a) = q_k(a) = 1$.

We now prove that θ is primitive recursive. First, compute $x <_{\mathcal{P}_i} a$, if true, then compute $q_k(x)$. Now $q_k(x) \leq_{\mathcal{D}} 1$ as $q_j(a) = q_k(a) = 1$. Notice that by construction, any element that appears in $\mathcal{A}$ on the left of 1 appears in $\mathcal{D}$ at the same time. Then $\delta_{\mathcal{D}}^{-1}(q_k(x))[s] \downarrow$ where s is the least stage when $x <_{\mathcal{P}_i} a$ and $q_k(x)$ are computed. In the case that $x \leq_{\mathcal{P}_i} a$ is false, then compute $q_j(x)$ and then the same argument (swapping left to right) can be used to show why $\delta_{\mathcal{C}}^{-1}(q_j(x))[s] \downarrow$ at the stage s such that these computations have converged. $\dashv$

LEMMA 3.6. *For all $e, i, j, k \in \mathbb{N}$ requirements $Q_{i,j,k}$, R_e^0 and R_e^1 are met.*

PROOF. If a requirement never requires attention, then p_e, p_j , or p_k are not surjective and therefore the requirement is met. Suppose that the requirement requires attention at some stage s . By Lemmas 3.2–3.5, once a requirement is given attention the requirement is met and never requires attention again. The action of each requirement is finite and therefore we give attention to infinitely many requirements. Hence the requirement is eventually given attention and met. $\dashv$

LEMMA 3.7. *For all $i, j, k \in \mathbb{N}$, requirement $P_{i,j,k}$ is met.*

PROOF. If p_j and p_k are not isomorphisms then we have met the requirement, so assume p_j and p_k are isomorphisms.

Suppose $p_j(\gamma_{\mathcal{C}}^{-1}(b_0)) = p_k(\gamma_{\mathcal{D}}^{-1}(b_0))$ for some $b_0 \in \mathcal{B}$. Non-uniformly fix b_0 , then define $\eta(b) = p_j(\gamma_{\mathcal{C}}^{-1}(b))$ for $b \leq b_0$ and $\eta(b) = p_k(\gamma_{\mathcal{D}}^{-1}(b))$ for $b > b_0$. By the same argument as the proof in Lemma 3.5, $\eta : \mathcal{B} \rightarrow \mathcal{P}_i$ is a primitive recursive isomorphism. Informally, this case is that because p_j and p_k ‘line up’ somewhere, we can then use this to make a primitive recursive isomorphism by copying p_j to the left and p_k to the right of the place where they line up.

Otherwise $p_j(\gamma_{\mathcal{C}}^{-1}(b_0)) \neq p_k(\gamma_{\mathcal{D}}^{-1}(b_0))$ for all $b_0 \in \mathcal{B}[s]$. In particular there are $c \in \mathcal{C}$, $d \in \mathcal{D}$ such that $p_j(c) = p_k(z)$ and $p_k(d) = p_j(z)$. Then $c <_{\mathcal{C}} z$ or $d <_{\mathcal{D}} z$ since p_j and p_k are isomorphisms. See Figures 4 and 5 for a diagram of each case.

Suppose that $c < z$, as shown in Figure 4. We define a primitive recursive isomorphism $v : (\gamma_{\mathcal{C}}(c), 1) \subseteq \mathcal{B} \rightarrow (z, 1) \subseteq \mathcal{D}$ as follows.

Suppose t is the least stage such that $p_j(c)[t] \downarrow$ and $p_k(d)[t] \downarrow$. At stage t , we can start defining v . Notice that at every stage $s \geq t$, $(\gamma_{\mathcal{C}}(c), 1)$ contains more elements than $(z, 1)$ does. We define v so that at each stage s it maps some initial segment of $(\gamma_{\mathcal{C}}(c), 1)$ onto $(z, 1)$ and we will show that v will be defined on the rest of the elements in $(\gamma_{\mathcal{C}}(c), 1)$ at the next stage. Note that v is injective and order-preserving by construction. At stage t , let $(\gamma_{\mathcal{C}}(c), x)$ be the initial segment in $(\gamma_{\mathcal{C}}(c), 1)[t]$ of size $|(z, 1)[t]|$. Map $(\gamma_{\mathcal{C}}(c), x)$ onto $(z, 1)[t]$. See Figure 6 for a diagram of v .

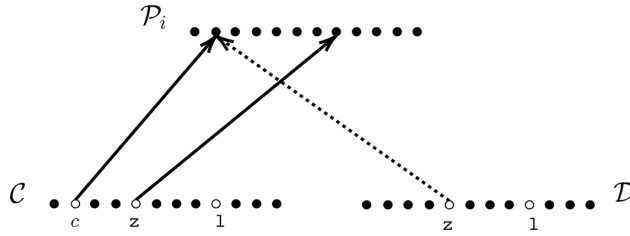


FIGURE 4. A snapshot of a P requirement where $c < z$. The dashed arrows represent p_k and solid arrows represent p_j .

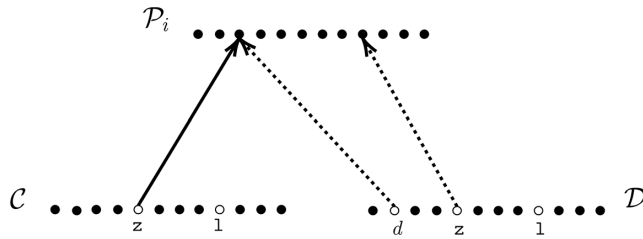


FIGURE 5. A snapshot of a P requirement where $d < z$. The dashed arrows represent p_k and solid arrows represent p_j .

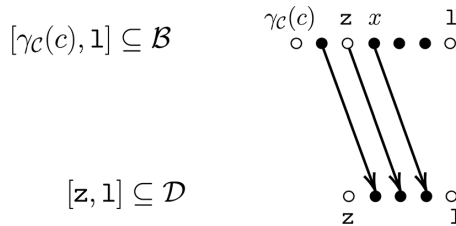


FIGURE 6. A diagram showing $v : (\gamma_C(c), 1) \subseteq \mathcal{B} \rightarrow (z, 1) \subseteq \mathcal{D}$ at some stage s .

At stage $s > t$, we define v as follows. Let $x \in (\gamma_C(c), 1)$ be the greatest element such that $v(x)[s-1]$ is defined. By construction of v , it is defined on all elements in $(\gamma_C(c), x]$ at stage $s-1$ and therefore at the end of each stage v is as shown in Figure 6.

If s is not a recovery stage, let (x, y) be the initial segment in $(x, 1)[s]$ of size $|(v(x), 1)[s]|$. Map (x, y) onto $(v(x), 1)[s]$.

If s is a recovery stage, then recall that the recovery stage has two steps, first we grow $(z_s, 1)$, then we add densifying elements (which means an element is added between every pair of elements that are currently adjacent). As soon as we grow $(z_s, 1)$, let (x, y) be the initial segment in $(x, 1)$ of size $|(v(x), 1)|$. Map (x, y) onto

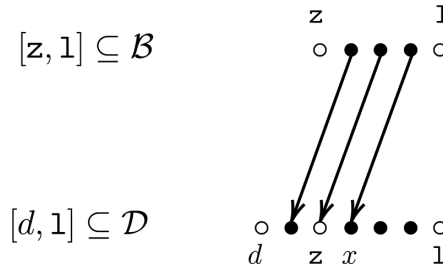


FIGURE 7. A diagram showing $v' : [z, 1] \subseteq \mathcal{B} \rightarrow [d, 1] \subseteq \mathcal{D}$ at stage s .

$(v(x), 1)$. Now we add the densifying elements as follows. Let a be an element added between some b_0, b_1 where $v(b_0)$ and $v(b_1)$ are defined. Then define v to map a to the densifying element between $v(b_0)$ and $v(b_1)$.

Now we show that v is a primitive recursive isomorphism on domain $(\gamma_C(c), 1)$. v is surjective because at each stage s , v maps an initial segment of $(z, 1)$, (z, y) , onto $(z, 1)[s]$. Recall that at each stage we add $|\mathcal{B}[s-1]|$ many elements between z_s and 1 in $\mathcal{D}$. Since there are less than $|\mathcal{B}[s-1]|$ many elements between z and z_s at stage $s-1$ in $\mathcal{B}$, $y \geq z_s$. Therefore, every element in $(z, 1)$ gets an image either immediately or at the next stage. By construction, notice that the size of $\mathcal{B}[s]$ is bounded by 4^{s+1} and hence the size of $\mathcal{B}[s]$ is bounded by a primitive recursive function. Then it follows that v is defined on elements either immediately or at the next stage. Therefore, v is primitive recursive on domain $(\gamma_C(c), 1)$.

Otherwise, $d < z$, as shown in Figure 5. We define a primitive recursive isomorphism $v' : (z, 1) \subseteq \mathcal{B} \rightarrow (d, 1) \subseteq \mathcal{D}$ as follows.

Suppose t is the least stage such that $p_j(c)[t]$ and $p_k(d)[t]$. At stage t , we can start defining v . Notice that at every stage $s \geq t$, $(z, 1)$ contains fewer elements than $(d, 1)$ does. Note that v is injective and order-preserving by construction. Let x be such that (d, x) has size $|(z, 1)|$. Then we define v' to map $(z, 1)$ onto (d, x) . See Figure 7.

At stage $s > t$, we define v' as follows. Let $x = v'(z_s)$, i.e., the greatest value in the image of v' defined at the previous stage. Notice that in this case, in contrast to the previous case, we always map all elements in $(z, 1)$ immediately and at the end of each stage v' is as shown in Figure 7.

To extend the definition of v' when we grow $(z_s, 1)$, we do as follows. Let (x, y) be the initial segment in $(x, 1)$ of size $|(z_s, 1)|$. Map $(z_s, 1)$ onto (x, y) . We define v' on densifying elements at recovery stages as in the previous case.

Notice that in this case, all elements in $(z, 1)[s]$ get defined immediately. Therefore, v' is primitive recursive on domain $(z, 1)$. At every stage, elements are added between z_s and 1 , so every element in $(d, 1)$ eventually gets a preimage. Therefore, v' is a primitive recursive isomorphism on domain $(z, 1)$.

Recall that for each element $x \in \mathcal{B}[s]$, at least one of $\gamma_C^{-1}(x)[s]$ and $\gamma_D^{-1}(x)[s]$ converges. The important fact is that if there is a new element, $x \in \mathcal{B}[s]$, to the left of all other elements so far, then by construction $\gamma_C^{-1}(x)[s]$. Similarly, if this new element is to the right, then $\gamma_D^{-1}(x)[s]$. New elements added in the interval $[z, 1]$ are

added to all structures at the same stage, so for every element in this interval in $\mathcal{B}[s]$, both $\gamma_C^{-1}(x)[s]$ and $\gamma_D^{-1}(x)[s]$ converge. Recall that $\gamma_C(z) = z$, $\gamma_C(1) = 1$, $\gamma_D(z) = z$, and $\gamma_D(1) = 1$.

If $c < z$, define $\eta : \mathcal{B} \rightarrow \mathcal{P}_i$ as $\eta(b) = p_j(\gamma_C^{-1}(b))$ for $b \leq \gamma_C(c)$, $\eta(b) = p_k(\gamma_D^{-1}(b))$ for $b \geq 1$, and $\eta(b) = p_k(v(b))$ for $\gamma_C(c) < b < 1$. Then this is a primitive recursive isomorphism from $\mathcal{B}$ to $\mathcal{P}_i$ as desired.

If $d < z$, define $\eta' : \mathcal{B} \rightarrow \mathcal{P}_i$ as $\eta'(b) = p_j(\gamma_C^{-1}(b))$ for $b \leq z$, $\eta'(b) = p_k(\gamma_D^{-1}(b))$ for $b \geq 1$, and $\eta'(b) = p_k(v'(b))$ for $z < b < 1$. Then this is a primitive recursive isomorphism from $\mathcal{B}$ to $\mathcal{P}_i$ as desired. $\dashv$

LEMMA 3.8. $\mathcal{A} \cong \mathcal{B} \cong \mathcal{C} \cong \mathcal{D}$ are punctual and isomorphic to $\mathbb{Q}$.

PROOF. By the previous lemmas, infinitely many requirements are worked on. Before we start to work on a new requirement, we enter a recovery stage where we ensure all structures are isomorphic and an element is added between every pair of elements that are currently adjacent, hence the structure we get is dense. We have infinitely many R requirements that we work on; hence, the structures have no end points. Then it is the case that $\mathcal{A} \cong \mathcal{B} \cong \mathcal{C} \cong \mathcal{D} \cong \mathbb{Q}$. $\dashv$

§4. The integers. In this section, we prove Theorems 1.1 and 1.2 in the case of the integers.

4.1. Embedding the diamond.

THEOREM 4.1 (Diamond for $\mathbb{Z}$). *There exist punctual presentations $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ of $(\mathbb{Z}, <)$ such that $\mathcal{C}$ and $\mathcal{D}$ are incomparable under $\leq_{pr}$, $\mathcal{B} = \sup(\mathcal{C}, \mathcal{D})$ and $\mathcal{A} = \inf(\mathcal{C}, \mathcal{D})$.*

We build structures $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ so that they meet the following requirements²:

$R_e^0 : p_e : \mathcal{C} \rightarrow \mathcal{D}$ is not an isomorphism,

$R_e^1 : p_e : \mathcal{D} \rightarrow \mathcal{C}$ is not an isomorphism,

$P_{i,j,k} : \text{if } p_j : \mathcal{C} \rightarrow \mathcal{P}_i, p_k : \mathcal{D} \rightarrow \mathcal{P}_i \text{ are p.r. isomorphisms, then } \mathcal{B} \leq_{pr} \mathcal{P}_i,$

$Q_{i,j,k} : \text{if } q_j : \mathcal{P}_i \rightarrow \mathcal{C}, q_k : \mathcal{P}_i \rightarrow \mathcal{D} \text{ are p.r. isomorphisms, then } \mathcal{P}_i \leq_{pr} \mathcal{A},$

where $\langle p_e \rangle_{e \in \mathbb{N}}$ and $\langle \mathcal{P}_i \rangle_{i \in \mathbb{N}}$ are effective enumerations of all primitive recursive functions and punctual presentations, respectively. We also build primitive recursive isomorphisms $\gamma_C : \mathcal{C} \rightarrow \mathcal{B}$, $\gamma_D : \mathcal{D} \rightarrow \mathcal{B}$, $\delta_C : \mathcal{A} \rightarrow \mathcal{C}$, $\delta_D : \mathcal{A} \rightarrow \mathcal{D}$.

We meet R requirements using the *island technique*, as used in Section 2.1, where we introduce an element x to the right of all elements. This element acts as a floating *island* an unknown ‘distance’ away. We compute $p_e(x)$ and use this image to diagonalise. We note that the macros used in this section were used in previous section, but we restate them here for convenience and completeness.

We note that the following conventions are the same as used in Sections 2.1 and 3 but are restated here for convenience. For structures we are building $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$, we consider $\mathcal{E}[s]$ to be the presentation we have built so far at stage s . Throughout the

²Note that these requirements are the same as in the previous sections, they are restated here for convenience.

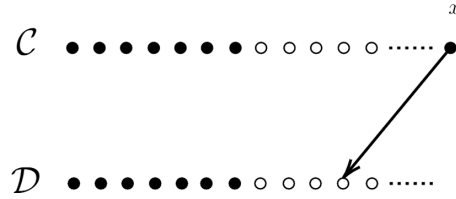


FIGURE 8. A diagram showing the extra element x in $\mathcal{C}$ and the way $\mathcal{C}$ and $\mathcal{D}$ grow during this requirement. The hollow circles represent new elements added during working on this requirement. The arrow represents p_e .

construction, we will *add a new element to $\mathcal{E}$* , this means we add the $<_{\mathbb{N}}$ -least element $x \in \mathbb{N}$ that is not yet in $\mathcal{E}[s]$ and we define $<_{\mathcal{E}}$ as described in the construction. In this construction, we will determine the punctuality of $\mathcal{E}$ by adding the element s to $\mathcal{E}[s]$ by the end of stage s . To ensure that the isomorphisms are primitive recursive, we define them on all elements in $\mathcal{E}[s]$ at stage s .

We will use the macro *grow $\mathcal{L}$ to the right* to mean, add one new element x to $\mathcal{L}$ so that $x > y$ for all $y \in \mathcal{L}[s]$. We use *grow $\mathcal{L}$ towards x* for some element x to mean, add one new element z to $\mathcal{L}$ so that $z < x$ and $z > y$ for all $y \in \mathcal{L}[s]$ such that $y < x$. We now detail the formal construction.

4.1.1. Construction. We arrange all requirements in some effective list and meet them one by one. All R and Q requirements require attention from stage 0. In contrast, we say that $P_{i,j,k}$ requires attention at stage s if there are $c \in \mathcal{C}[s]$ and $d \in \mathcal{D}[s]$ such that $p_j(c)[s] \downarrow = p_k(d)[s] \downarrow$. At every stage, we define isomorphisms $\gamma_{\mathcal{C}} : \mathcal{C} \rightarrow \mathcal{B}$, $\gamma_{\mathcal{D}} : \mathcal{D} \rightarrow \mathcal{B}$, $\delta_{\mathcal{C}} : \mathcal{A} \rightarrow \mathcal{C}$, $\delta_{\mathcal{D}} : \mathcal{A} \rightarrow \mathcal{D}$. We note that we will abuse notation and say that $c \leq d$ for $c \in \mathcal{C}$ and $d \in \mathcal{D}$, by this, we mean that $\gamma_{\mathcal{C}}(c) \leq_{\mathcal{B}} \gamma_{\mathcal{D}}(d)$.

At stage 0, begin working on the first requirement on the list. We now detail the strategies for each requirement.

Requirement R_e^0 . Add an extra element, x , into $\mathcal{B}$ and $\mathcal{C}$ and declare that x is larger than any element added so far. At every stage, while we work on this requirement, grow $\mathcal{C}$ and $\mathcal{B}$ towards x , while $\mathcal{A}$ and $\mathcal{D}$ grow to the right. See Figure 8 for a diagram of this. Compute p_e on all elements enumerated into $\mathcal{C}$. Eventually there must be $z \in \mathcal{C}$ such that $z <_{\mathcal{C}} x$ and $p_e(z) >_{\mathcal{D}} p_e(x)$. Once we see this element, we have met the requirement. Now add an element y to the right in $\mathcal{A}$ and $\mathcal{D}$ and define $\delta_{\mathcal{C}}(y) = x$, $\delta_{\mathcal{D}}(y) = y$, $\gamma_{\mathcal{D}}(y) = x$.

Requirement R_e^1 . Do the same as in the previous requirement but switch the roles of $\mathcal{C}$ and $\mathcal{D}$.

Requirement $Q_{i,j,k}$. If at any stage during this strategy q_j or q_k is seen to not be an isomorphism, halt the stage and move to recovery. Wait for some $l \in \mathcal{P}_i$ to appear and q_j and q_k to be computed on l such that $q_j(l) = c$ and $q_k(l) = d$ for some new elements $c \in \mathcal{C}$, $d \in \mathcal{D}$, where a new element is an element that was added since we began working on this requirement. While we wait, grow all the structures to the

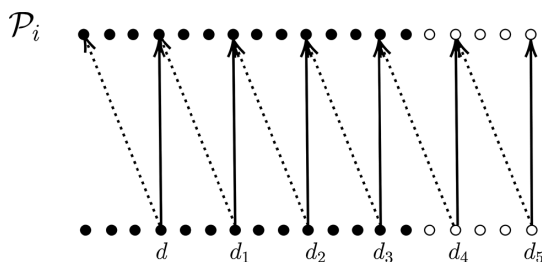


FIGURE 9. A diagram showing the sequence d_i . The hollow circles represent new elements added during working on this requirement, as specified in the construction, in the case of this diagram $m = 4$. The solid arrows represent p_j and the dashed arrows represent p_k . Note that we do not represent $\mathcal{C}$ and $\mathcal{D}$ separately in this diagram since they are isomorphic during this requirement and are identified with each other using $\gamma_{\mathcal{C}}$ and $\gamma_{\mathcal{D}}$.

right. Once we find l , if $c = d$ then $Q_{i,j,k}$ is good, move to recovery. Otherwise, if $c < d$ then grow all structures towards d . In the case that $d < c$, grow all structures towards c .

Requirement $P_{i,j,k}$. Recall that if we are working on $P_{i,j,k}$, this means it requires attention, so there are $c \in \mathcal{C}$ and $d \in \mathcal{D}$ such that $p_j(c) \downarrow = p_k(d) \downarrow$. If at any stage during this strategy p_j or p_k is seen to not be an isomorphism, halt the stage and move to recovery. If $c = d$, then we say that $P_{i,j,k}$ is good, halt the stage and move to the next R or Q requirement on the list. Otherwise consider the following two cases:

- $c < d$. Suppose there are n elements between c and d . We find a sequence $d_0 = d, d_1, \dots, d_{m+1}$ such that $p_j(d_i) = p_k(d_{i+1})$ and m is such that d_m is a new element since the stage we started to work on this requirement, as can be seen in Figure 9. While we work to define this sequence, grow all structures to the right. Let i be the least such that $m + 1 \geq i > 0$ and we have not yet found d_i . If there is an x such that $p_k(x) = p_j(d_{i-1})$ then let $x = d_i$. If there is a stage where there is z such that $p_k(z) > p_j(d_{i-1})$ and there is no $x < z$ such that $p_k(x) = p_j(d_{i-1})$, then move to the next requirement. If we have not moved on from this requirement, then we have found the sequence $d_0, d_1, \dots, d_{m+1}$. Add $n + 1$ elements between d_m and d_{m+1} .
- $d < c$. Do the same as in the previous case, swapping the roles of $\mathcal{C}$ and $\mathcal{D}$ as well as p_j and p_k .

Halt the stage and move to the least requirement which requires attention and has not yet been worked on.

Recovery. Add a new element on the left in each structure. Note that at the end of this stage, the structures all have the same number of elements and $\gamma_{\mathcal{C}}[s], \gamma_{\mathcal{D}}[s], \delta_{\mathcal{C}}[s], \delta_{\mathcal{D}}[s]$ are all isomorphisms. Halt the stage, move to the least requirement which requires attention and has not yet been worked on.

4.1.2. *Verification.* Now we verify that all requirements are met.

LEMMA 4.2. *For all $e \in \mathbb{N}$, if requirement R_e^0 is given attention then R_e^0 is met.*

PROOF. Compute $p_e(x)$. Suppose there are n elements on the left of $p_e(x) \in \mathcal{D}$ at this stage. Wait for $n + 1$ elements to appear on the left of $x \in \mathcal{C}$. Compute p_e on these elements. Since we grow $\mathcal{D}$ only to the right, there are still n elements on the left of $p_e(x)$, then p_e will show it is not an isomorphism. $\dashv$

LEMMA 4.3. *For all $e \in \mathbb{N}$, if requirement R_e^1 is given attention then R_e^1 is met.*

PROOF. The proof is the same to the previous one with the $\mathcal{C}$ and $\mathcal{D}$ swapped. $\dashv$

LEMMA 4.4. *For all $i, j, k \in \mathbb{N}$, if requirement $Q_{i,j,k}$ is given attention then $Q_{i,j,k}$ is met.*

PROOF. If q_j and q_k are not injective or order-preserving, then $Q_{i,j,k}$ is met. Suppose q_j and q_k are injective and order-preserving.

First we prove that we eventually find l such that $q_j(l) = c$ and $q_k(l) = d$ for some new elements $c \in \mathcal{C}$, $d \in \mathcal{D}$. Suppose not, then we show that q_j or q_k is not injective. Indeed, suppose that there are n elements in each structure at the beginning of the stage. When $2n + 1$ new elements appear in $\mathcal{P}_i$, compute q_j and q_k on them. Suppose that at least $n + 1$ of these new elements were mapped into new elements in $\mathcal{C}$, then these elements were mapped into old elements in $\mathcal{D}$. In this case, q_k is not injective. Now suppose that there were less than $n + 1$ elements mapped into new ones in $\mathcal{C}$, then at least $n + 1$ elements were mapped into old elements in $\mathcal{C}$ and q_j is not injective. Thus, we eventually find l such that $q_j(l) = c$ and $q_k(l) = d$ for some new elements $c \in \mathcal{C}$, $d \in \mathcal{D}$.

If $c = d$ and q_j and q_k are isomorphisms, then $\delta_C^{-1}(q_j(l)) = \delta_D^{-1}(q_k(l))$ for all $l \in \mathcal{P}_i$. Define an isomorphism $\theta(l) = \delta_C^{-1}(q_j(l))$. This is an isomorphism since q_j and δ_C^{-1} are isomorphisms. θ is primitive recursive because at the least stage that both $q_j(l)$ and $q_k(l)$ converge, $\delta_C^{-1}(q_j(l))$ converges.

Consider the case $c \neq d$. Let t be the stage when there is $l \in \mathcal{P}_i[t]$ such that $q_j(l)[t] = c$ and $q_k(l)[t] = d$ for new elements $c \in \mathcal{C}[t]$, $d \in \mathcal{D}[t]$. Let n be the number of elements in $\mathcal{C}$ at stage t (note that all structures have the same size at this stage). By construction, in this case, we grow the interval between c and d in all structures while we wait for a disagreement. We show that this eventually happens. Let $x = \min(c, d)$ and $y = \max(c, d)$. Notice that, while we wait there are no more than n elements on the left of x and no more than n elements on the right of y . Once $2n + 1$ new elements to appear in $\mathcal{P}_i$, we see disagreement because there must be at least $n + 1$ new elements on the left or the right of l . In the former case, there are not enough elements to the left of x , hence we will see that q_j or q_k is not an isomorphism (depending whether $x = c$ or d). In the latter case, there are not enough elements to the right of y , hence we will see that q_j or q_k is not an isomorphism, a contradiction. $\dashv$

LEMMA 4.5. *For all $i, j, k \in \mathbb{N}$, if requirement $P_{i,j,k}$ is given attention then $P_{i,j,k}$ is met.*

PROOF. If p_j and p_k are not injective or order-preserving, then $P_{i,j,k}$ is met. Suppose p_j and p_k are injective and order-preserving.

If $P_{i,j,k}$ is good, then for all $b \in \mathcal{B}$, $p_j(\gamma_{\mathcal{C}}^{-1}(b)) = p_k(\gamma_{\mathcal{D}}^{-1}(b))$. Define $\eta : \mathcal{B} \rightarrow \mathcal{P}_i$ as follows. At the least stage s such that $b \in \mathcal{B}[s]$, define $\eta(b) = p_j(\gamma_{\mathcal{C}}^{-1}(b))$ if $\gamma_{\mathcal{C}}^{-1}(b)[s] \downarrow$, otherwise define $\eta(b) = p_k(\gamma_{\mathcal{D}}^{-1}(b))$. By construction, at stage s at least one of $\gamma_{\mathcal{D}}^{-1}(b)$ and $\gamma_{\mathcal{C}}^{-1}(b)$ is defined. Therefore η is a primitive recursive isomorphism as desired.

Now suppose $P_{i,j,k}$ is not good. We show that it cannot be the case that both p_j and p_k are isomorphisms. First, we notice that when we begin to work on any requirement at some stage s , any interval of $\mathcal{A}[s-1]$, $\mathcal{B}[s-1]$, $\mathcal{C}[s-1]$, and $\mathcal{D}[s-1]$ is fixed, i.e., no new elements will ever appear in any old interval. Therefore, in the case that we were able to find the sequence $d_0, \dots, d_{m+1}$, the interval $(p_j(c), p_j(d))$ has n elements while the interval $(p_j(d_m), p_j(d_{m+1}))$ has $n+1$ elements; hence p_j or p_k is not an isomorphism. If it is the case that there was some i such that we could not find d_i , then there is z such that $p_k(z) > p_j(d_{i-1})$. Since there is no d_i , there is no $x < z$ such that $p_k(x) = p_j(d_{i-1})$. Then p_k is not surjective since we will never add any elements between d_{i-1} and z . Therefore, we have met requirement $P_{i,j,k}$. $\dashv$

LEMMA 4.6. For all $e, i, j, k \in \mathbb{N}$, requirement R_e^0 , R_e^1 , $Q_{i,j,k}$ and $P_{i,j,k}$ is met.

PROOF. By Lemmas 4.2–4.5, every requirement is eventually met if given attention. By construction and these lemmas, it follows that the construction works on each requirement for finitely many stages. Therefore, every requirement that requires attention is eventually given attention. In the case of P requirements, if $P_{i,j,k}$ does not require attention, then it is met because p_j or p_k is not surjective.

Therefore, all requirements are met. $\dashv$

LEMMA 4.7. $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ are presentations of $(\mathbb{Z}, <)$.

PROOF. Notice that when we start working on a new strategy, we do not place elements in between any old elements. When we do add elements between some new elements we place only finitely many. Infinitely often we add an element to the left and to the right. Therefore, $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ are presentations of $(\mathbb{Z}, <)$. $\dashv$

4.2. Embedding the atomless Boolean algebra. In this section, we prove the integers case of Theorem 1.2. We use the strategies from the diamond for integers, naturally extending the construction to the full atomless Boolean algebra.

Let β denote the atomless Boolean algebra. We will detail how to extend the above techniques to build a presentation $\mathcal{A}_m$ of $\mathcal{L}$ for each $m \in \beta$. Let $\preceq_\beta$ be the partial order defined as follows $a \preceq_\beta b$ if $a \wedge b = a$. We build the presentations $\mathcal{A}_a$ for each $a \in \beta$ so that they meet the following requirements.

$R_e^{a,b}$: if $a \not\preceq_\beta b$, then $p_e : \mathcal{A}_a \rightarrow \mathcal{A}_b$ is not an isomorphism,

$P_{i,j,k}^{a,b}$: if $p_j : \mathcal{A}_a \rightarrow \mathcal{P}_i$, $p_k : \mathcal{A}_b \rightarrow \mathcal{P}_i$ are p.r. isomorphisms, then $\mathcal{A}_{a \vee b} \leq_{pr} \mathcal{P}_i$,

$Q_{i,j,k}^{a,b}$: if $q_j : \mathcal{P}_i \rightarrow \mathcal{A}_a$, $q_k : \mathcal{P}_i \rightarrow \mathcal{A}_b$ are p.r. isomorphisms, then $\mathcal{P}_i \leq_{pr} \mathcal{A}_{a \wedge b}$,

where $\langle p_e \rangle_{e \in \mathbb{N}}$ and $\langle \mathcal{P}_i \rangle_{i \in \mathbb{N}}$ are effective enumerations of all primitive recursive functions and punctual presentations, respectively.

We also build primitive recursive isomorphisms $\gamma_{a,b} : \mathcal{A}_a \rightarrow \mathcal{A}_b$ for all $a, b \in \beta$ with $a \preceq_\beta b$.

4.2.1. Construction. We arrange all requirements in some effective list and meet them one by one. At each stage, we are building finitely many presentations and we will slowly introduce more presentations to build throughout the construction. Let E_s be the indices of presentations we have started building by stage s . Arrange all elements of β in some effective list; we will use this list to introduce new structures throughout the construction.

The requirements *require attention at stage s* as follows:

- $R_e^{a,b}$ requires attention at stage s if $a, b \in E_s$.
- $Q_{i,j,k}^{a,b}$ requires attention at stage s if $a, b, c \in E_s$ where $c = a \wedge b$.
- $P_{i,j,k}^{a,b}$ requires attention at stage s if $a, b, c \in E_s$ for $c = a \vee b$ and there are $x \in \mathcal{A}_a[s]$ and $y \in \mathcal{A}_b[s]$ such that $p_j(x)[s] \Vdash p_k(y)[s]$.

At every stage, we define isomorphisms $\gamma_{a,b} : \mathcal{A}_a \rightarrow \mathcal{A}_b$ such that $a \preceq_\beta b$ for $a, b \in E_s$. As in the diamond, we will abuse notation and say that $x = y$ for $x \in \mathcal{A}_a$ and $y \in \mathcal{A}_b$, this means that $\gamma_{a,c}(x) = \gamma_{b,c}(y)$ for some $c \succeq_\beta a, b$. We note that by construction, if this is true for some $c \succeq_\beta a, b$ then it is actually true for all $c \succeq_\beta a, b$.

At stage 0, let $E_0 = \{a, b\}$, where a and b are the first elements on the list of elements in β . Begin working on the first requirement on the list that requires attention.

At stage $s > 0$, enact the strategy of the current requirement we are working on, if no requirement currently requires attention, then go to recovery. We now detail the strategies for each requirement.

Requirement $R_e^{a,b}$. Do as in strategy for requirement R_e^0 from the diamond construction, but replace $\mathcal{C}$ with $\mathcal{A}_a$ and $\mathcal{D}$ with $\mathcal{A}_b$. All presentations $\mathcal{A}_c$ with $c \succ_\beta a$ and $c \in E_s$, copy $\mathcal{A}_a$. All other presentations in E_s copy $\mathcal{A}_b$. All $\gamma_{d,d'}$ are naturally extended.

Requirement $Q_{i,j,k}^{a,b}$. Do as in strategy $Q_{i,j,k}$ from the diamond construction, replacing c, d with x, y , respectively, and replacing $\mathcal{C}, \mathcal{D}$ with $\mathcal{A}_a, \mathcal{A}_b$, respectively. As in the diamond construction, all structures grow in the same way during this requirement, so $\mathcal{A}_d$ copies $\mathcal{A}_a$ for all $d \in E_s$. All $\gamma_{d,d'}$ are naturally extended.

Requirement $P_{i,j,k}^{a,b}$. Do as in strategy $P_{i,j,k}$ from the diamond construction, replacing c, d with x, y , respectively, and replacing $\mathcal{C}, \mathcal{D}$ with $\mathcal{A}_a, \mathcal{A}_b$, respectively. As in the diamond construction, all structures grow in the same way during this requirement, so $\mathcal{A}_d$ copies $\mathcal{A}_a$ for all $d \in E_s$. All $\gamma_{d,d'}$ are naturally extended.

Recovery. Add a new element on the left in each structure in E_s . Note that at the end of this stage, all established structures have the same number of elements, and for all $a \preceq_\beta b$, $\gamma_{a,b} : \mathcal{A}_a \rightarrow \mathcal{A}_b$ are all isomorphisms. Let c be the least element on the list of elements of β that has not yet been introduced. Build $\mathcal{A}_c$ so that it is isomorphic to all $\mathcal{A}_a$ ($a \in E_s$), define isomorphisms $\gamma_{c,b}$ and $\gamma_{a,c}$ for all $a, b \in E_s$, $a \preceq_\beta c$ and $c \preceq_\beta b$, and define $E_{s+1} = E_s \cup \{c\}$. Halt the stage, move to the highest priority requirement that requires attention.

4.2.2. Verification. Notice that at the start of working on a new requirement, all structures are isomorphic (i.e., they have the same number of elements).

LEMMA 4.8. *For all $a, b \in \beta$ with $a \preceq_\beta b$, $\gamma_{a,b} : \mathcal{A}_a \rightarrow \mathcal{A}_b$ is a primitive recursive isomorphism.*

PROOF. While working on Q and P , all structures grow in the same way, hence it is clear that during these stages $\gamma_{c,d}$ are naturally extended for all $c, d \in \beta$ with $c \preceq_\beta d$. During $R_e^{a,b}$ requirements, we have the following cases for $c, d \in \beta$ with $c \preceq_\beta d$:

- For any presentations $\mathcal{A}_c, \mathcal{A}_d$ with $c, d \succeq_\beta a$, act as $\mathcal{A}_a$ and therefore grow the same way, so $\gamma_{c,d}$ are naturally extended.
- For any presentations $\mathcal{A}_c, \mathcal{A}_d$ with $c, d \not\succeq_\beta a$, act as $\mathcal{A}_b$ and therefore grow the same way, so $\gamma_{c,d}$ naturally extended.
- If we are not in any of the previous cases, then we have $c \not\succeq_\beta a$ and $d \succeq_\beta a$. In this case, $\mathcal{A}_c$ acts as $\mathcal{A}_b$ and $\mathcal{A}_d$ acts as $\mathcal{A}_a$. Then $\gamma_{c,d}$ is naturally extended while leaving only the special element out of its range. Note that this element enters the range in recovery.

As above, by construction, at every stage s , we define $\gamma_{c,d}$ immediately on every element of $\mathcal{A}_c[s]$, hence $\gamma_{c,d}$ is primitive recursive. When we start working on a new requirement at stage s , $\gamma_{c,d}$ is an isomorphism on domain $\mathcal{A}_c[s]$ onto $\mathcal{A}_d[s]$. $\dashv$

LEMMA 4.9. *For all $a, b \in \beta$ and $e, i, j, k \in \mathbb{N}$, requirements $R_e^{a,b}$, $Q_{i,j,k}^{a,b}$, and $P_{i,j,k}^{a,b}$ are met.*

PROOF. This follows immediately from Lemmas 4.2 and 4.4–4.6 by swapping roles of $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ to $\mathcal{A}_{a \vee b}, \mathcal{A}_{a \wedge b}, \mathcal{A}_a, \mathcal{A}_b$, respectively. $\dashv$

Therefore, all requirements are met and hence we have embedded the atomless Boolean algebra into the punctual degrees of $(\mathbb{Z}, <)$.

Acknowledgements. The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

REFERENCES

- [1] P. E. ALAEV, *Existence and uniqueness of structures computable in polynomial time*. *Algebra and Logic*, vol. 55 (2016), no. 1, pp. 72–76.
- [2] C. J. ASH and J. KNIGHT, *Computable Structures and the Hyperarithmetical Hierarchy*, Studies in Logic and the Foundations of Mathematics, 144, North-Holland Publishing Co., Amsterdam, 2000.
- [3] R. BAGAVIEV, I. I. BATYRSHIN, N. BAZHENOV, D. BUSHTETS, M. DORZHEVA, and H. T. KOH, et al., *Computably and punctually universal spaces*. *Annals of Pure and Applied Logic*, vol. 176 (2025), no. 1, p. 103491.
- [4] N. A. BAZHENOV and I. S. KALIMULLIN, *Punctual categoricity spectra of computably categorical structures*. *Algebra and Logic*, vol. 60 (2021), no. 3, pp. 223–228, Cited by: 1.
- [5] N. BAZHENOV, R. DOWNEY, I. KALIMULLIN, and A. G. MELNIKOV, *Foundations of online structure theory*. *The Bulletin of Symbolic Logic*, vol. 25 (2019), no. 2, pp. 141–181.
- [6] N. BAZHENOV, I. KALIMULLIN, A. G. MELNIKOV, and K. M. NG, *Online presentations of finitely generated structures*. *Theoretical Computer Science*, vol. 844 (2020), pp. 195–216.
- [7] D. CENZER, R. G. DOWNEY, J. B. REMMEL, and Z. UDDIN, *Space complexity of abelian groups*. *Archive for Mathematical Logic*, vol. 48 (2009), no. 1, pp. 115–140.
- [8] D. CENZER and J. REMMEL, *Polynomial-time versus recursive models*. *Annals of Pure and Applied Logic*, vol. 54 (1991), no. 1, pp. 17–58.
- [9] ———, *Polynomial-time abelian groups*. *Annals of Pure and Applied Logic*, vol. 56 (1992), no. 1, pp. 313–363.
- [10] ———, *Feasibly Categorical Abelian Groups*, Birkhäuser, Boston, MA, 1995, pp. 91–153.
- [11] D. CENZER and J. B. REMMEL, *Polynomial-Time Versus Computable Boolean Algebras*, De Gruyter, Berlin and Boston, 1999, pp. 15–54.

- [12] R. J. COLES, R. DOWNEY, and B. KHOUSSAINOV, *On initial segments of computable linear orders*. **Order**, vol. 14 (1997), no. 2, pp. 107–124.
- [13] M. DORZHIEVA, R. DOWNEY, E. HAMMATT, A. G. MELNIKOV, and K. M. NG, *Punctually presented structures II: comparing presentations*. **Archive for Mathematical Logic**, vol. 64 (2025), pp. 159–184.
- [14] M. DORZHIEVA and A. G. MELNIKOV, *Punctually presented structures I: Closure theorems*. **Computability**, vol. 12 (2023), no. 4, pp. 323–337.
- [15] R. DOWNEY, N. GREENBERG, A. G. MELNIKOV, K. M. NG, and D. TURETSKY, *Punctual categoricity and universality*. **The Journal of Symbolic Logic**, vol. 85 (2020), no. 4, pp. 1427–1466.
- [16] R. DOWNEY, A. G. MELNIKOV, and K. M. NG, *Foundations of online structure theory II: The operator approach*. **Logical Methods in Computer Science**, vol. 17 (2021), no. 3, pp. 1–35.
- [17] R. DOWNEY, M. HARRISON-TRAINOR, I. KALIMULLIN, A. G. MELNIKOV, and D. TURETSKY, *Graphs are not universal for online computability*. **Journal of Computer and System Sciences**, vol. 112 (2020), pp. 1–12.
- [18] R. G. DOWNEY, A. M. KACH, S. LEMPP, A. E. M. LEWIS-PYE, A. MONTALBÁN, and D. D. TURETSKY, *The complexity of computable categoricity*. **Advances in Mathematics**, vol. 268 (2015), pp. 423–466.
- [19] S. GONCHAROV, S. LEMPP, and R. SOLOMON, *The computable dimension of ordered abelian groups*. **Advances in Mathematics**, vol. 175 (2003), no. 1, pp. 102–143.
- [20] S. S. GONCHAROV, *The quantity of nonautoequivalent constructivizations*. **Algebra and Logic**, vol. 16 (1977), no. 3, pp. 169–185.
- [21] ———, *Autostability of models and abelian groups*. **Algebra and Logic**, vol. 19 (1980), no. 1, pp. 13–27.
- [22] S. S. GONCHAROV and V. D. DZGOEV, *Autostability of models*. **Algebra and Logic**, vol. 19 (1980), no. 1, pp. 28–37.
- [23] S. S. GONCHAROV, **Countable Boolean Algebras and Decidability**, Siberian School of Algebra and Logic, Kluwer Academic/Plenum, New York, 1997.
- [24] G. A. GRATZER, **Lattice Theory: Foundation**, Springer, Basel, 2011.
- [25] S. GRIGORIEFF, *Every recursive linear ordering has a copy in $dtime\text{-}space(n, \log(n))$* . **The Journal of Symbolic Logic**, vol. 55 (1990), no. 1, pp. 260–276.
- [26] I. S. KALIMULLIN, A. G. MELNIKOV, and K. M. NG, *The diversity of categoricity without delay*. **Algebra and Logic**, vol. 56 (2017), no. 2, pp. 171–177.
- [27] I. S. KALIMULLIN, *Constructing punctually categorical semigroups*. **Algebra and Logic**, vol. 59 (2020), no. 5, pp. 408–411.
- [28] I. KALIMULLIN, A. G. MELNIKOV, and A. MONTALBÁN, *Punctual definability on structures*. **Annals of Pure and Applied Logic**, vol. 172 (2021), no. 8, Paper No. 102987. 18.
- [29] I. KALIMULLIN, A. G. MELNIKOV, and K. M. NG, *Algebraic structures computable without delay*. **Theoretical Computer Science**, vol. 674 (2017), pp. 73–98.
- [30] I. KALIMULLIN, A. G. MELNIKOV, and M. ZUBKOV, **Punctual Degrees and Lattice Embeddings**, World Scientific, Singapore, 2023, pp. 315–334.
- [31] D. KALOCIŃSKI, L. S. MAURO, and M. WROCLAWSKI, *Punctual presentability in certain classes of algebraic structures*, **49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)** (R. Kráľovič and A. Kučera, editors), Leibniz International Proceedings in Informatics (LIPIcs), 306, Schloss Dagstuhl—Leibniz-Zentrum für Informatik, Dagstuhl, 2024, pp. 65:1–65:15.
- [32] H. T. KOH, A. G. MELNIKOV, and K. M. NG, *A non-density aspect of the rationals*, to appear, 2024.
- [33] P. LAROCHE, *Recursively presented Boolean algebras*. **Notices AMS**, vol. 24 (1977), pp. 552–553.
- [34] S. LEMPP, C. MCCOY, R. MILLER, and R. SOLOMON, *Computable categoricity of trees of finite height*. **The Journal of Symbolic Logic**, vol. 70 (2005), no. 1, pp. 151–215.
- [35] A. I. MAL'CEV, *Constructive algebras. I*, **The metamathematics of algebraic systems** (A. Ivanovič Mal'cev, editors), Studies in Logic and the Foundations of Mathematics, 66, Elsevier, Amsterdam, Netherlands, 1971, pp. 148–214.
- [36] A. G. MELNIKOV and K. M. NG, *Computable torsion abelian groups*. **Advances in Mathematics**, vol. 325 (2018), pp. 864–907.
- [37] ———, *The back-and-forth method and computability without delay*. **Israel Journal of Mathematics**, vol. 234 (2019), no. 2, pp. 959–1000.

- [38] ———, *A structure of punctual dimension two*. *Proceedings of the American Mathematical Society*, vol. 148 (2020), no. 7, pp. 3113–3128.
- [39] A. MONTALBÁN, *Computable Structure Theory: Beyond the Arithmetic*, 2023, to appear.
- [40] A. MONTALBÁN, *Computable Structure Theory: Within the Arithmetic*, Cambridge University Press, Cambridge, 2021.
- [41] J. B. REMMEL, *Recursively categorical linear orderings*. *Proceedings of the American Mathematical Society*, vol. 83 (1981), no. 2, pp. 387–387.
- [42] J. G. ROSENSTEIN, *Linear Orderings*, Pure and Applied Mathematics, 98, Academic Press, Inc., [Harcourt Brace Jovanovich, Publishers], New York, NY, 1982.
- [43] V. SELIVANOV and S. SELIVANOVA, *Primitive recursive ordered fields and some applications*, *Computer Algebra in Scientific Computing* (F. Boulier, M. England, T. M. Sadykov, and E. V. Vorozhtsov, editors), Springer International Publishing, Cham, 2021, pp. 353–369.
- [44] R. L. SMITH, *Two Theorems on Autostability in p -Groups*, Springer, Berlin, 1981, pp. 302–311.

SCHOOL OF MATHEMATICS AND STATISTICS
 VICTORIA UNIVERSITY OF WELLINGTON
 WELLINGTON 6012
 NEW ZEALAND

E-mail: dm-3004@inbox.ru

INSTITUTE OF DISCRETE MATHEMATICS AND GEOMETRY,
 TECHNISCHE UNIVERSITÄT WIEN
 VIENNA 1040
 AUSTRIA

E-mail: ellen.hammatt@tuwien.ac.at