Online and Feasible Presentability: From Trees to Modal Algebras

Nikolay Bazhenov
Novosibirsk State University, Novosibirsk, Russia

Dariusz Kalociński
Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Michał Wrocławski
Faculty of Philosophy, University of Warsaw, Warsaw, Poland

Abstract

We investigate whether every computable member of a given class of structures admits a fully primitive recursive (also known as punctual) or fully P-TIME copy. A class with this property is referred to as punctually robust or P-TIME robust, respectively. We present both positive and negative results for structures corresponding to well-known representations of trees, such as binary trees, ordered trees, sequential (or prefix) trees, and partially ordered (poset) trees. A corollary of one of our results on trees is that semilattices and lattices are not punctually robust. In the main result of the paper, we demonstrate that, unlike Boolean algebras, modal algebras – that is, Boolean algebras with modality – are not punctually robust. The question of whether distributive lattices are punctually robust remains open. The paper contributes to a decades-old program on effective and feasible algebra, which has recently gained momentum due to rapid developments in punctual structure theory and its connections to online presentations of structures.

2012 ACM Subject Classification Theory of computation → Computability

Keywords and phrases Algebraic structure, computable structure, fully primitive recursive structure, punctual structure, polynomial-time computable structure, punctual robustness, tree, semilattice, lattice, Boolean algebra, modal algebra

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.142

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: `https://arxiv.org/abs/2504.16663`

Funding *Nikolay Bazhenov*: The work of N. Bazhenov is supported by the Mathematical Center in Akademgorodok under the agreement No. 075-15-2025-349 with the Ministry of Science and Higher Education of the Russian Federation.
Dariusz Kalociński: The work of D. Kalociński is supported by the National Science Centre Poland under the agreement No. 2023/49/B/HS1/03930.

Acknowledgements We would like to thank Luca San Mauro and the reviewers for helpful comments.

1 Introduction

Suppose an infinite graph $G = (\mathbb{N}, E)$ is given via a computer program P, which, for any pair $x, y \in \mathbb{N}$, determines whether $E(x, y)$ holds. This graph is an infinite structure represented in a finitary manner. Such structures are called computable.[1] However, without further restrictions on P, the presentation of G via P may be inefficient. Does there always exist a more efficient representation of G – that is a computer program P' such that P' computes E', $G' = (\mathbb{N}, E')$ is isomorphic to G, and E' is primitive recursive or even P-TIME computable?

[1] An infinite countable structure $\mathcal{A}$ is computable if the domain of $\mathcal{A}$ is a computable subset of $\mathbb{N}$ and all the relations and functions from the signature of $\mathcal{A}$ are uniformly computable [2].

The answer to the question posed above is negative [27]. However, if we impose certain structural restrictions on G – such as requiring G to be strongly locally finite or to be a tree – the answer becomes positive [14]. A more general condition that guarantees the existence of a nice copy of G has been presented in [28].

The question raised in the opening paragraph admits several natural generalizations. For instance, one could replace graphs with other structures or explore alternative effectiveness conditions beyond primitive recursiveness or polynomial-time computability. Various instances of this question have drawn interest from researchers at the intersection of automata theory, complexity theory, and computable algebra. Notable examples include studies on algebraic structures represented by finite-state automata [30, 9, 22] and investigations into polynomial-time algebra [36, 13, 1]. Recently, Kalimullin, Melnikov, and Ng [27] initiated a research program at the intersection of computability and complexity, where primitive recursiveness serves as a fundamental tool for representing structures. Henceforth, we adopt this paradigm in the paper.

▶ **Definition 1** (punctual structure [27]). *An infinite structure over a finite signature is* punctual *if its domain is* $\mathbb{N}$ *and all of its relations and functions are primitive recursive.*

In the literature, punctual structures are also referred to as fully primitive recursive (fpr).

The purpose of the above definition was to capture the notion of an infinite structure which can be presented in an online fashion, or "without delay". The requirement that the domain be $\mathbb{N}$ (or any *fixed* primitive recursive subset of $\mathbb{N}$) is significant. If, instead, we require the domain can be *any* primitive recursive subset of $\mathbb{N}$, a distinct notion arises (cf. Theorem 1.2 in [13] and the associated discussion). More importantly, structures satisfying the relaxed definition may lack the typical online characteristics [27, 5].

Henceforth, all structures, unless stated otherwise, are countably infinite.

A more detailed explanation of the underlying nature of punctual structures follows. This will be useful later, as many subsequent constructions adhere to this general pattern.

Essentially, a punctual structure $\mathcal{A}$ may be given by a primitive recursive algorithm P which, on input $\overline{x} = x_1, x_2, \ldots, x_n$, determines $R^{\mathcal{A}}(\overline{x})$ and $f^{\mathcal{A}}(\overline{x})$, for every R, f from the signature of $\mathcal{A}$. Intuitively, for arguments $x_i \leq s$ (where $s \in \mathbb{N}$), all the relations and function values are determined by the primitive recursive computation $P(\overline{x})$ which should converge by the computational stage s. In a manner of speaking, our decision whether $R(\overline{x})$, or what is $f(\overline{x})$, must be *quick*, or made *without delay*. Here *quick* and *without delay* means *now*, that is at most at the stage s. In fact, Definition 1 allows for a delay, but the delay itself must be primitive recursive – any instance of a truly unbounded search is prohibited.

As observed above, a punctual presentation of a structure possesses the following core property: *input is received and processed incrementally, and decisions about incoming data must be made without access to the complete set of problem data.* This property is a defining characteristic of online procedures in general, both in finite and infinite settings: in the finite setting, online algorithmics comprises a distinctive subfield of computer science, where performances of online algorithms are compared, relative to the offline performance, using competitive analysis (see, e.g., [10]); in the infinite setting, online graph coloring procedures from infinite combinatorics serve as a pertinent example (see, e.g., [31]). Due to the core property punctual structures share with online algorithms, it has been suggested that they encapsulate, at least partially, what may intuitively be described as a truly online presentation of a countably infinite structure; the justification of this approach is a delicate task and has been accomplished elsewhere [27, 5]. For alternative theoretical approaches, see [18, 3].

The theory of online structures has rapidly emerged as an intriguing subfield of computable structure theory, yielding an impressive amount of results across various topics, including universality [17, 20], degree structures [7, 23, 35], and definability [26], among others [6, 16, 34]. One of the central topics in online structure theory concerns punctual presentability.

▶ **Definition 2.** *A structure is* punctually presentable *if it is isomorphic to a punctual one.*

This definition formalizes the property we asked about G in the opening paragraph of this paper. More generally, given a class $\mathfrak{K}$ of structures, we ask whether every computable member of $\mathfrak{K}$ has a punctual copy.

▶ **Definition 3** (punctual robustness [28])**.** *We say that a class of structures $\mathfrak{K}$ is* punctually robust*, if every computable member of $\mathfrak{K}$ is punctually presentable.*

Turning to more feasible computations, similar notions naturally arise in the polynomial-time setting. As usual, our underlying model for P-TIME computations is provided by Turing machines.

▶ **Definition 4** (P-TIME structure and P-TIME robustness)**.** *By $\mathbb{B}$ we denote $\{0,1\}^{<\omega}$, i.e., the set of all finite binary strings. A structure $\mathcal{S}$ in a finite signature L is* P-TIME *(or polynomial-time computable) if the domain of $\mathcal{S}$ is a polynomial-time computable subset of $\mathbb{B}$, and all the L-relations and L-functions of $\mathcal{S}$ are polynomial-time computable. A P-TIME structure $\mathcal{S}$ is* fully P-TIME *if the domain of $\mathcal{S}$ is equal to $\mathbb{B}$. A structure is fully P-TIME presentable if it is isomorphic to a fully P-TIME structure.*

We say that a class of structures $\mathfrak{K}$ is P-TIME robust *if every computable member of $\mathfrak{K}$ is fully P-TIME presentable.*

We note that for this framework, in place of the binary alphabet $\{0,1\}$ in the definition of P-TIME robustness, one can choose an arbitrary alphabet Σ which contains at least two symbols (see, e.g., Chapter 3 in [11]).

A proof of punctual/P-TIME non-robustness for a given isomorphism-closed class $\mathfrak{K}$ of L-structures (where L indicates the signature) usually requires specific strategies tailored to the type of structure we are dealing with. However, the general scheme is an example of diagonalization and can be framed in game-theoretic terms as follows. We imagine playing a game against infinitely many adversaries $\mathcal{A}_1, \mathcal{A}_2, \ldots$. Adversaries are given via a computable enumeration. Each $\mathcal{A}_i$ is a punctual/P-TIME L-structure and $\mathcal{A}_1, \mathcal{A}_2, \ldots$ contains all isomorphism-types of punctual/P-TIME copies of the structures from $\mathfrak{K}$. The proof must give a *computable* strategy for building $\mathcal{B} \in \mathfrak{K}$ such that $\forall i(\mathcal{A}_i \in \mathfrak{K} \Rightarrow \mathcal{B} \not\cong \mathcal{A}_i)$.

The following classes of structures have been shown to be punctually robust: equivalence structures [13, 27], (relational) successor trees [14], linear orders [24, 27], torsion-free abelian groups [27], Boolean algebras [27], abelian p-groups [27], graphs with infinite semitransversals [28]; on the other hand, there are computable undirected graphs [27], computable torsion abelian groups [12], computable Archimedean ordered abelian groups [27] and functional predecessor trees [28] with no punctual copy.

Contributions

In this work we provide new results on punctual and P-TIME robustness of trees, (semi)lattices and modal algebras. Note that if a class is not punctually robust then it is not P-TIME robust either. Hence, P-TIME robustness is attested only for punctually robust classes. In the following, we briefly describe each of the three contributions.

The first group of results concerns trees. It is a continuation of the work by Cenzer and Remmel on (variations of) relational successor trees [14], as well as of Kalociński, San Mauro and Wrocławski on functional predecessor trees [28]. Our results extend this line of research to a few other well-known tree structures.

One of the fundamental types of tree-like structures is an ordered tree ([32], p. 306). This concept arises naturally by imposing an ordering on the children of each node in the tree. To avoid confusion with poset trees (to be defined), we use the term *relational predecessor trees with ordering*. The definition could have used the successor relation instead, without affecting Theorem 6 (see below).

▶ **Definition 5** (r.p.o. tree). *$\mathcal{T} = (T, P, <, r)$ is a relational predecessor tree with ordering (abbreviated r.p.o. tree) if P is a binary relation and $<$ is a partial ordering satisfying:*

1. *(T, P) is a directed graph such that all edges are oriented towards the root r, and the underlying undirected graph is connected and acyclic,*
2. *for every $x, y \in T$, $x \leq y \vee y \leq x$ iff x and y have the same parent.*

The relation P is the immediate predecessor relation of the tree. In Section 2 we prove:

▶ **Theorem 6.** *The class of r.p.o. trees is punctually robust and P-TIME robust.*

Theorem 6 unifies and expands on both the results of Cenzer and Remmel [14] on trees, and of Grigorieff [24] on linear orderings.

Another type of the tree that we consider is a poset tree. Given a partial order $(P, \leq)$, we define $P_{\leq x} = \{y \in P : y \leq x\}$ and $P_{\geq x} = \{y \in P : y \geq x\}$.

▶ **Definition 7** (poset tree). *A partial order $(P, \leq)$ is a poset tree if $(P, \leq)$ has the greatest element (the root) and, for every $x \in P$, $P_{\geq x}$ is a finite linear order.*

Such trees are well-known in set theory, where a tree is a partially ordered set $(P, <)$, usually with the least element, such that for each $x \in P$, the set $P_{<x}$ is well-ordered (see, e.g., Definition 9.10 in [25]). The inessential difference between the two definitions is the direction in which the tree grows. A more important difference is that, in set theory, a node may be infinitely far apart from the root. Our definition does not allow it. However, the following theorem of ours, which we prove in Section 3, works even for the class of trees in the set-theoretic sense (provided we reverse the growth direction):

▶ **Theorem 8.** *Poset trees are not punctually robust.*

While the underlying combinatorial idea behind the proof is quite intuitive, and some variants of it have been used elsewhere for a stronger notion of a tree [14], the present case requires much more care, as nodes in a poset tree do not carry information about their distances from the root. Even more importantly, Theorem 8 yields non-robustness of structures that lie intermediate between trees and Boolean algebras:

▶ **Corollary 9.** *The following classes of structures are not punctually robust:*

- **(i)** *join semilattices and meet semilattices,*
- **(ii)** *lattices, complemented lattices, and non-distributive lattices.*

The result holds under both order-theoretic and algebraic interpretation.

The remaining open question is whether the class of distributive lattices is punctually robust. The strategy used to obtain Theorem 8 is closely related to the "non-distributivity" of the computable poset tree we construct, and is therefore unlikely to provide a solution. To resolve the question, a different approach would be required, if distributive lattices are punctually robust at all.

The third contribution deals with modal algebras which are obtained from Boolean algebras by adding a modality operator. Boolean algebras constitute a classical object in mathematical logic. It is well-known that the class of Boolean algebras provides algebraic semantics for the classical propositional calculus (see, e.g., [37]). On the other hand, countable Boolean algebras are well-studied in computable structure theory – we refer to the monographs [21, 19] for a detailed exposition of results in this area.

Intuitively speaking, here we treat a *modality* as a formal logical operator which expresses the possibility of formulas: $\Diamond p$ means that the statement p is possible. More formally, one typically considers additional logical connectives $\Diamond$ (the possibility operator) and $\Box$ (the necessity operator), and defines an appropriate modal propositional calculus which extends the classical propositional calculus (see, e.g., the monograph [15]). Modal algebras provide algebraic semantics for (normal) modal logics. The formal definition of a modal algebra is given in Section 4. In contrast to Boolean algebras, there are only few known results about computable presentations of modal algebras – here we cite [29, 4].

It turns out that adding modality $\Diamond$ to the language of Boolean algebras enriches the computational content of the class: while (pure) Boolean algebras are punctually robust (informally, they are computationally "tame") [27], modal algebras are not punctually robust (i.e., some of them can exhibit a "wilder" subrecursive behavior). In Section 4 we prove:

▶ **Theorem 10.** *The class of modal algebras is not punctually robust.*

Last but not least, we provide some elementary results about robustness of infinite binary trees and prefix trees – the details about them can be found in the related version.

2 Relational Predecessor Trees with Ordering

If $P(x, y)$ holds and $x \neq r$, then we say that x is a *child* of y (and y is a parent of x). Notice that here we assume that the root r does not have parents. If x is a child of y and y is a child of z, then we say that x is a *grandchild* of z. An analogous convention holds for all representations of trees under consideration in this article.

▶ **Definition 11.** *If $\mathcal{T}$ is a tree (r.p.o. tree or a different type) and a is a node of that tree, then we define the depth of a in the following way: for the root of the tree $d^{\mathcal{T}}(r) = 0$ and whenever b is a child of a, then $d^{\mathcal{T}}(b) = d^{\mathcal{T}}(a) + 1$. In case of representations of trees with an empty node e we also define $d^{\mathcal{T}}(e) = -1$. We also define $\mathcal{T}^{\leq n}$ to be the part of $\mathcal{T}$ consisting only of nodes with a depth less or equal n.*

Because of the space limit, here we only prove a specific case of Theorem 6. The full proof of Theorem 6 appears in the related version.

▶ **Theorem 12.** *The class of r.p.o. trees of unbounded depth is P-TIME robust.*

Proof. We fix a computable $\mathcal{T} = (\mathbb{B}, P^{\mathcal{T}}, <^{\mathcal{T}}, r^{\mathcal{T}})$. We carry out the construction in stages $s = 0, 1, \ldots$. During each stage we perform a fixed number l of steps of some algorithm calculating $\mathcal{T}$. We can assume that during each stage the algorithm at most once returns a true statement of the type either $P^{\mathcal{T}}(a, b)$ or $a <^{\mathcal{T}} b$ for some $a, b \in \mathbb{B}$ and that in each such case a and b are connected to the root of the current approximation of $\mathcal{T}$.

We are going to construct a P-TIME copy $\mathcal{R} \cong \mathcal{T}$. At the end of each stage s structures $\mathcal{T}_s$ and $\mathcal{R}_s$ are going to be our current approximations of $\mathcal{T}$ and $\mathcal{R}$ respectively. We observe that according to the construction described below both of them will be rooted at every stage but at some stages they could be not isomorphic to each other. We are also going to construct an isomorphism $\varphi : \mathcal{T} \to \mathcal{R}$.

To ensure that the construction is P-TIME, if for some time we do not discover any new relations in $\mathcal{T}$, we can delay establishing positive cases of relations in $\mathcal{R}$ by putting new fresh strings into a reservoir set A. The strings in that set will not be immediately put in a fixed place in the tree $\mathcal{R}$ but we will guarantee that none of them can be in either relation with each other. Thus, while waiting to discover positive cases of relations in $\mathcal{T}$, we will keep adding negative cases of relations to $\mathcal{R}$. Since the tree has infinite depth, we are guaranteed to be able to add all the strings from the reservoir to the P-TIME tree at some point.

We order $\mathbb{B}$ in the following way: $\alpha \sqsubset \beta$ if either $lh(\alpha) < lh(\beta)$ or $lh(\alpha) = lh(\beta)$ and additionally on the first position that they differ, α has 0 and β has 1.

We construct the reservoir set A. Initially $A = \emptyset$. At the end of each stage s we take the $\sqsubset$-least sequence α from outside $\mathcal{R}_s$, add α to A and declare that α is short.

We want to ensure that during infinitely many stages $\mathcal{T}_s \cong \mathcal{R}_s$. We only intend to update $\mathcal{R}_s$ to satisfy this condition if we discover $a, b, c \in \mathcal{T}$ such that $P^{\mathcal{T}}(b, a)$ and $P^{\mathcal{T}}(c, b)$ and the isomorphism $\varphi(a) = \alpha$ is defined while $\varphi(b)$ and $\varphi(c)$ are not defined. Then we define $\varphi(c) = \beta$ where β is the $\sqsubset$-least element of A and we remove β from A.

For every other $d \in \mathcal{T}_s \setminus \varphi^{-1}[\mathcal{R}_s]$: $d_1, \cdots, d_j$, we take the $\sqsubset$-least sequences of length at least s: $\delta_1, \cdots, \delta_j$ and define $\varphi(d_i) = \delta_i$ for $1 \leq i \leq j$. We declare that all δ_i are long.

Verification

▶ **Lemma 13.** *$\mathcal{R}_s \cong \mathcal{T}_s$ infinitely many times.*

Proof. Suppose that the depth of $\mathcal{T}_s$ is d_s at the end of some stage s, $\mathcal{R}_s$ is isomorphic to $\mathcal{T}_s$ and $\mathcal{R}_s$ does not change later.

Then the depth of $\mathcal{T}_t$ is at most $d_s + 1$ for all $t \geq s$. This is because otherwise $\mathcal{T}_t$ would have an element a of depth $d_t > d_s + 1$ such that $a_0, \cdots, a_{d_t}$ is a path in $\mathcal{T}_t$ starting from the root $a_0 = r$ until $a_{d_t} = a$. In this path some a_i is the last element such that $\varphi(a_i) = \alpha_i$ is defined. Since this path has not been prolonged in $\mathcal{R}_t$ beyond α_i we conclude that $i = d_t$ or $i = d_t - 1$. Hence the depth of α_i is either d_t or $d_t - 1$. Also since α_i is in $\mathcal{R}_s$ we conlude that the depth of α_i is less or equal d_s. Hence $d_t - 1 \leq d_s$ which is impossible.

Since the depth of $\mathcal{T}_t$ is less or equal $d_s + 1$ we obtain a contradiction with the assumption that this tree is unbounded. ◀

The above lemma implies that $\mathcal{T} \cong \mathcal{R}$.

▶ **Lemma 14.** *The domain of $\mathcal{R}$ is $\mathbb{B}$.*

Proof. We observe that $\sqsubset$ is a well-ordering on $\mathbb{B}$. We suppose that α is the $\sqsubset$-least sequence not in the domain of $\mathcal{R}$.

Then α was never added to A because every sequence from A is eventually added to $\mathcal{R}$ (since $\mathcal{R}$ gets extended infinitely many times and each time the $\sqsubset$-least sequence from A is used to perform such extension). The sequence α can only avoid getting added to A if α is long and was earlier added to $\mathcal{R}$ but this is also a contradiction. ◀

▶ **Lemma 15.** *$P^{\mathcal{R}}$ and $<^{\mathcal{R}}$ are P-TIME.*

Proof. We observe that if α and β are in either relation from the signature, then either of them is long (maybe both of them). We assume that β is long and that $lh(\alpha) \leq lh(\beta) = s$.

Then to discover that α and β are in a relation we perform the construction until stage s and check if $\mathcal{R}_s$ confirms that they are. We observe that if α and β do not enter a relation at the latest during stage s, then they do not later.

We show that the above procedure is P-TIME. To perform the construction until stage s we perform $l \times s$ steps of the algorithm calculating $\mathcal{T}$ and we add at most s new sequences to $\mathcal{R}$, the length of each of them is at most s. Hence the algorithm is P-TIME. ◀

Theorem 12 is proven. ◀

3 Poset Trees

Let $(T, \leq)$ be a poset tree. We say that elements $x, y \in T$ are adjacent, $Adj(x, y)$, if and only if $x < y \wedge \neg\exists z \in T\,(x < z < y)$, where $<$ is the strict partial order induced by $\leq$. We say $x \in T$ is a *branching node* of T (or x branches in T) if it has at least two children in T. Such an x induces a unique subtree of T, called a *branching* and defined as $br(x, T) = \{y \in T : Adj(y, x)\} \cup T_{\geq x}$. If x is a branching node, we define $|br(x, T)|$, the length of $br(x, T)$, as the length of $T_{\geq x}$. By a *binary branching* we mean a tree with exactly two leaves sharing a parent. We say that $(T, \leq)$ is *uniquely branching* if for every $n \in \mathbb{N}$, there exists at most one branching node $x \in T$ such that $|T_{\geq x}| = n$. We say that a *branching* node $x \in T$ belongs to the *level* n of T if the set $T_{>x}$ contains precisely n branching nodes. We denote the level n of T by $T[n]$ and call its members n-level nodes. Keep in mind that we number levels $0, 1, \ldots$. Therefore, the first level is level 0. Level $T[n]$ may be empty but if $T[n] \neq \emptyset$ then $T[k] \neq \emptyset$, for $k < n$. Let $(T, \leq)$ be a tree such that its level i, $i \in \mathbb{N}$, is nonempty. We define $T_{[\leq i]}$, the *i-level subtree of* T, as the least subtree of T containing all nodes at levels $\leq i$ together with their children. Notice that $T_{[\leq i]}$ is the sum of the branchings $br(x, T)$ for all $x \in T[j]$ such that $j \leq i$. By $r(T)$ we denote the root of the tree T.

▶ **Lemma 16.** *Let $(T, \leq)$ be a finite poset tree in which every internal node has exactly two children and let $F \subseteq T$ be such that, at every level of T except the first, at most one node is in F. Then there exists a leaf $y \in T$ such that $T_{\geq y} \cap F = \emptyset$.*

Proof. By assumption, $r(T) \notin F$. Let $x \in T$ be such that $T_{\geq x} \cap F = \emptyset$. The node x has two children which are at the same level, so one of them, denote it by x', is outside F. Therefore, $T_{\geq x'} \cap F = \emptyset$. This way we find a leaf y such that $T_{\geq y} \cap F = \emptyset$. ◀

Let $T, \hat{T}$ be disjoint finite trees and let $z \in T$ be a leaf. T' is obtained from T by attaching $\hat{T}$ to z in T if $dom(T') = dom(T) \cup (dom(\hat{T}) - \{r(\hat{T})\})$ and $x, y \in dom(T')$ satisfy $Adj_{T'}(x, y)$ iff $Adj_T(x, y) \vee Adj_{\hat{T}}(x, y) \vee Adj_{\hat{T}}(x, r(\hat{T})) \wedge y = z$.

We are ready to start the proof of Theorem 8. We build a computable poset tree $\mathcal{T} = (\mathbb{N}, \leq^T)$ such that for every $i \in \mathbb{N}$:

$$\mathcal{P}_i = (\mathbb{N}, \phi_i^{(2)}) \text{ is a poset tree } \implies \mathcal{T} \not\cong \mathcal{P}_i. \tag{R_i}$$

Here, $(\phi_i^{(2)})_{i \in \mathbb{N}}$ is a computable list of all binary primitive recursive functions.

Intuitively, the strategy will work as follows. We wait (in a dovetail fashion) until $\mathcal{P}_i$ shows large enough fragment P of itself (all nodes of $\mathcal{P}_i$ up to level i) isomorphic to the corresponding fragment of $\mathcal{T}$ that we construct (otherwise, we are done with $\mathcal{P}_i$ in the limit). We enumerate fresh elements into $\mathcal{P}_i$ so that the current $\mathcal{P}_i$ outnumbers the current approximation of $\mathcal{T}$. These fresh elements form subtrees of $\mathcal{P}_i$ attached to the leaves of P. We now apply the pigeonhole principle: one of the subtrees must have more elements than the corresponding subtree in $\mathcal{T}$ – we block the growth of that part in $\mathcal{T}$ and make $\mathcal{P}_i$ non-isomorphic to $\mathcal{T}$.

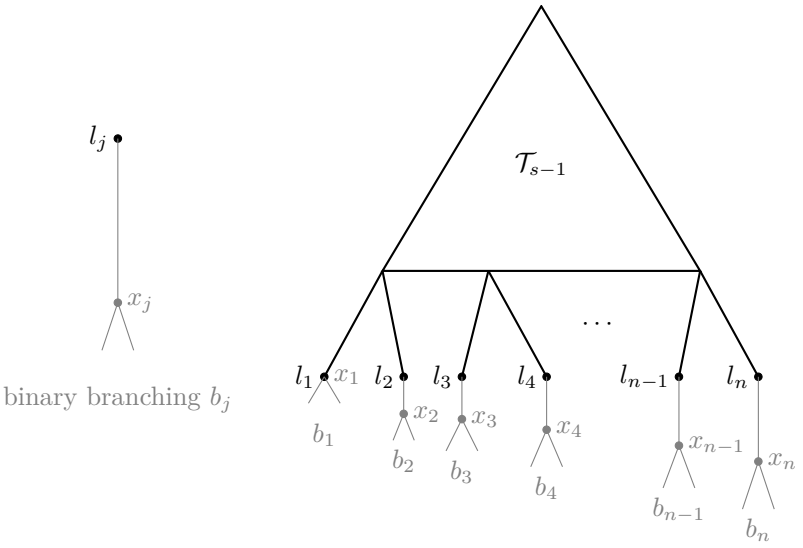

Figure 1 Obtaining $\mathcal{T}_s$ (black and gray) from $\mathcal{T}_{s-1}$ (black) at odd stage s by attaching binary branchings b_j (left) to l_j in $\mathcal{T}_{s-1}$, for $1 \leq j \leq n$.

$\mathcal{T}$ will be a uniquely branching poset tree with infinitely many levels. Alongside, we maintain a dynamic set F of elements of $\mathcal{T}$ whose role is to block the growth of the tree. At odd stages the tree grows wherever F permits. At even stages, the strategies for $\mathcal{P}_i$ monitor the relationship between $\mathcal{T}$ and $\mathcal{P}_i$ and put or withdraw elements from F to satisfy R_i. At any given stage s' we are dealing with an approximation $\mathcal{P}_{i,s'}$ of $\mathcal{P}_i$ defined as follows: $\mathcal{P}_{i,s'} = (N_{i,s'}, \leq_{i,s'})$, where $N_{i,s'} = \{0, 1, \ldots, s'-1\}$ and for all $x, y \in N_{i,s'}$, we have $x \leq_{i,s'} y \Leftrightarrow \phi_i^{(2)}(x, y) = 1$.

A node $y \in \mathcal{T}$ is *closed* (at a given stage) if there exists x such that $y \leq^T x$ and $x \in F$ at that stage. A node y is *open* if y is not closed.

Construction. We proceed in stages. At the end of stage s, we have a finite tree $\mathcal{T}_s$. We set $\mathcal{T} = \bigcup_s \mathcal{T}_s$.

At stage $s = 0$, the domain of $\mathcal{T}_0$ is $T_0 = \{0\}$ and $0 \leq^T 0$. No R_i is satisfied at stage 0, $F = \emptyset$. At each subsequent stage $s > 0$, we start with a finite tree $\mathcal{T}_{s-1}$.

Stage $s = 2t + 1$ (expansionary). Let $l_1, l_2, \ldots, l_n$ be the open leaves of $\mathcal{T}_{s-1}$. Construct, for $1 \leq j \leq n$, a binary branching b_j that branches at x_j, with the root l_j and other elements fresh (i.e., the domain of $\mathcal{T}$ is extended by an interval $[|T_s|; b)$, for a least b sufficient to construct the branchings), such that after attaching b_j to l_j in $\mathcal{T}_{s-1}$, the length of the branching in the new tree starting at x_j is equal to the height of $\mathcal{T}_{s-1}$ plus $j - 1$. Declare that tree as $\mathcal{T}_s$ (cf. Figure 1). Observe that if $\mathcal{T}_{s-1}$ is uniquely branching, then $\mathcal{T}_s$ is also uniquely branching.

Stage $s = 2t > 0$ (non-expansionary). Set $\mathcal{T}_s = \mathcal{T}_{s-1}$. For each $\mathcal{P}_i$, $0 \leq i \leq s$, apply the strategy for $\mathcal{P}_i$. Structurally $\mathcal{T}_s$ and $\mathcal{T}_{s-1}$ are the same, but they may differ with respect to open nodes (that is, F at stage $s-1$ may be different from F at stage s).

$\mathcal{P}_i$-strategy at stage $s = 2t$. Let $s' = \max(s, |T_s| + 1)$. We consider the approximation $\mathcal{P}_{i,s'}$ of $\mathcal{P}_i$. Let $s'' = \max(s-2, |T_{s-2}| + 1)$. Therefore, $\mathcal{P}_{i,s''}$ is the approximation of $\mathcal{P}_i$ that we considered at the previous non-expansionary stage (if $i \leq s''$). We say that the strategy is ready if:

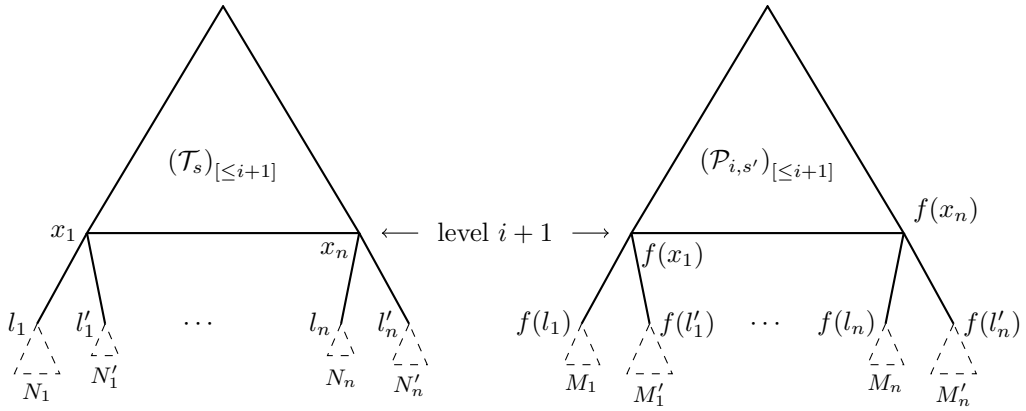

Figure 2 Schematic representation of $\mathcal{T}_s$ and $\mathcal{P}_{i,s'}$ from the point of view of the strategy for $\mathcal{P}_i$ at stage s. Solid lines correspond to the tree $(\mathcal{T}_s)_{[\leq i+1]}$ (left) and $(\mathcal{P}_{i,s'})_{[\leq i+1]}$ (right). The strategy for $\mathcal{P}_i$ monitors the cardinalities of the subtrees, which are represented by dashed triangles.

$$i \leq s-2, \text{ that is, } \mathcal{P}_i \text{ was considered at the previous non-expansionary stage} \tag{1}$$

$$\mathcal{P}_{i,s'} \text{ is a poset tree,} \tag{2}$$

$$\mathcal{T}_s[i+1] \neq \emptyset \text{ and } \mathcal{P}_{i,s'}[i+1] \neq \emptyset, \tag{3}$$

$$(\mathcal{P}_{i,s'})_{[\leq i+1]} = (\mathcal{P}_{i,s''})_{[\leq i+1]}, (\mathcal{T}_s)_{[\leq i+1]} = (\mathcal{T}_{s-2})_{[\leq i+1]}, \text{ and} \tag{4}$$

$$(\mathcal{T}_s)_{[\leq i+1]} \cong (\mathcal{P}_{i,s'})_{[\leq i+1]}. \tag{5}$$

If the strategy is not ready, withdraw all $(i+1)$-level nodes from F. If the strategy is ready and some node from level $i+1$ is in F, we do nothing.

The remaining case is when the strategy is ready and nodes from level $i+1$ are not in F. The situation is illustrated in Figure 2. Let f be an isomorphism from $(\mathcal{T}_s)_{[\leq i+1]}$ to $(\mathcal{P}_{i,s'})_{[\leq i+1]}$. Let $x_1, \ldots, x_n$ be the $(i+1)$-level nodes of $(\mathcal{T}_s)_{[\leq i+1]}$. Each x_j has two children which we denote by l_j and l'_j. Each l_j and l'_j is the root of a subtree of $\mathcal{T}_s$ with cardinality N_j and N'_j, respectively. In $\mathcal{P}_{i,s'}$, we have the corresponding subtrees with roots $f(l_j), f(l'_j)$ and cardinalities M_j, M'_j. Since $(\mathcal{P}_{i,s'})_{[\leq i+1]} \cong (T_s)_{[\leq i+1]}$ but $\mathcal{P}_{i,s'}$ has more elements than $\mathcal{T}_s$, the surplus of elements must be placed in the subtrees just described. By the pigeonhole principle, it follows that there exists j, $1 \leq j \leq n$, such that $M_j + M'_j > N_j + N'_j$. Take the least such j and put x_j into F.

Verification

By construction, $\mathcal{T}$ is a binary poset tree and that $\mathcal{T}$ is uniquely branching. It is also easy to see that the domain of $\mathcal{T}$ is $\mathbb{N}$ and that $x \leq^T y$ is computable: it is sufficient to wait in an unbounded loop for a stage s at which $x, y \in \mathcal{T}_s$ and respond how they are related in $\mathcal{T}_s$.

▶ **Lemma 17.** *$\mathcal{T}$ is infinite. Hence, it has infinitely many levels.*

Proof. It is sufficient to prove that at every stage s, $\mathcal{T}_s$ has at least one open leaf (and thus, the tree is extended at each expansionary stage). It is easy to see that $\mathcal{T}_3$ has levels 0 and 1 (it looks like two binary branchings attached to the root). Let $s \geq 3$. Consider $\mathcal{T}'_s$ obtained from $\mathcal{T}_s$ as follows: for every branching nodes $x, y \in T_s$ such that $x <_T y$ and with no branching nodes between them, the path $\{z \in T_s : x <_T z \leq^T y\}$ is collapsed to the single element y (visually speaking, the nodes connecting two branching nodes from adjacent levels

are erased). Observe that $\mathcal{T}'_s$ is a finite proper binary tree. By construction, $\mathcal{T}'_s$ and F satisfy the premise of Lemma 16. Therefore, $\mathcal{T}'_s$ has an open leaf. It is immediate that $\mathcal{T}_s$ has an open leaf as well. It is easy to see that $\mathcal{T}$ has infinitely many branching nodes and infinitely many levels. ◀

▶ **Lemma 18.** *Every requirement is eventually satisfied.*

Proof. Fix a requirement R_i and assume $\mathcal{P}_i$ is a poset tree. We may additionally assume that $\mathcal{P}_i$ is binary and that $\mathcal{P}_i[i+1]$ is nonempty (that is, $\mathcal{P}_i$ has level $i+1$). If it does not, then R_i is satisfied, because $\mathcal{T}$ has infinitely many levels by Lemma 17.

Each of the properties (1), (2), (3) and (4) corresponds to a computable predicate of two variables, i and s. For $\mathcal{P}_i$ satisfying the assumptions from the previous paragraph, the corresponding predicates hold in the limit – there is s_0 such that they are true when $s > s_0$:

Property (1)**.** Immediate.

Property (2)**.** Recall that $s' = max(s, |T_s|+1)$. For large enough s, $r(\mathcal{P}_i)$ becomes a member of $\mathcal{P}_{i,s'}$. Once this happens, $\mathcal{P}_{i,s'}$ is always a poset tree.

Property (3)**.** The level $i+1$ of $\mathcal{T}$ is nonempty by Lemma 17, it is present in $\mathcal{T}_s$ for large enough s by construction. Above we assumed that the level $i+1$ of $\mathcal{P}_i$ is nonempty.

Property (4)**.** For large enough s, $\mathcal{T}_{[\leq i+1]} \subseteq \mathcal{T}_s$ and $(\mathcal{P}_i)_{[\leq i+1]} \subseteq \mathcal{P}_{i,s'}$.

Let s_0 be the least stage such that (1)-(4) hold for all stages $s \geq s_0$. In particular, it means that starting from stage s_0, we have $(\mathcal{P}_{i,s})_{[\leq i+1]} = (\mathcal{P}_i)_{[\leq i+1]}$ and $(\mathcal{T}_s)_{[\leq i+1]} = (\mathcal{T})_{[\leq i+1]}$.

If (5) does not hold at stage s_0, then it will not hold anymore, and thus R_i is satisfied. So suppose (5) always holds starting from stage s_0 and let f be the isomorphism from $(\mathcal{T}_{s_0})_{[\leq i+1]}$ to $(\mathcal{P}_{i,s_0})_{[\leq i+1]}$. Note that this isomorphism is unique because $\mathcal{T}$ is uniquely branching.

Observe that, at stage s_0-2, the strategy for $\mathcal{P}_i$ was not ready. If it were, then $s'_0 = s_0-2$ would be the least stage such that (1), (2), (3) and (4) hold for all even stages $s \geq s'_0$, which would contradict our choice of s_0. Therefore, at stage s_0-2 all $(i+1)$-level nodes were withdrawn from F. Hence, at stage s_0, the strategy for $\mathcal{P}_i$ is ready and nodes from level $i+1$ are not in F. It means that we put some x_j from level $i+1$ of $\mathcal{T}$ into F. At stage s_0, the subtree of $\mathcal{T}$ with the root x_j has fewer elements than the corresponding subtree of $\mathcal{P}_i$ with root $f(x_j)$. And this property will hold forever because at any subsequent stage, the strategy is ready and $x_j \in F$, and in that case we do nothing. Therefore, at any subsequent expansionary stage, the subtree of $\mathcal{T}$ with root x_j does not grow. It remains to observe that if there were an isomorphism g from $\mathcal{T}$ to $\mathcal{P}_i$, it would have to map x_j to $f(x_j)$, because $\mathcal{T}$ and $\mathcal{P}_i$ are isomorphic up to level $i+1$ and $\mathcal{T}$ is uniquely branching so f has only one legitimate choice for the value $f(x_j)$. ◀

We recall that a partial ordering $\mathcal{L} = (L, \leq)$ is called a join semilattice, meet semilattice, or lattice if, for any two elements a, b in $\mathcal{L}$, there exists a least upper bound (denoted $a \cup b$), a greatest lower bound (denoted $a \cap b$), or both, respectively. Join semilattices, meet semilattices, and lattices are also represented as algebraic structures, namely $(L, \cup)$, $(L, \cap)$, and $(L, \cup, \cap)$, respectively, and are characterized by appropriate axioms. For further details on lattices, we refer the reader to the standard literature [8].

We reserve the term *(semi)lattice* to refer specifically to these algebraic structures, while the term *order-theoretic (semi)lattice* is used for structures of the form $(L, \leq)$. For (semi)lattices, the order $\leq$ and the operations $\cup$ and $\cap$ are mutually definable. Specifically, $x \leq y$ if and only if $x \cup y = y$, and $x \leq y$ if and only if $x \cap y = x$.

▶ **Corollary 9.** *The following classes of structures are not punctually robust:*

(i) *join semilattices and meet semilattices,*

(ii) *lattices, complemented lattices, and non-distributive lattices.*

The result holds under both order-theoretic and algebraic interpretation.

Proof of Corollary 9(i). Let $\mathcal{T} = (\mathbb{N}, \subseteq)$ be the poset tree from Theorem 8. Since $\subseteq$ is a partial ordering of $\mathbb{N}$ and the join of $x, y \in \mathbb{N}$ is well defined, $\mathcal{T}$ is an order-theoretic join semilattice and, by Theorem 8, it has no punctual presentation. By defining $x \subseteq^* y \Leftrightarrow y \subseteq x$, and arguing as above, $\mathcal{T}^* = (\mathbb{N}, \subseteq^*)$ is a computable order-theoretic meet semilattice with no punctual copy.

Let $x \cup y$ denote the join of x and y in $\mathcal{T}$. Consider the join semilattice $(\mathbb{N}, \cup)$. The join $x \cup y$ is a computable function of x, y: we run the construction from the proof of Theorem 8 and wait for a stage s at which both x and y enter T_s. Then we have two cases:

1. For $\subseteq$-comparable x, y, we return the maximum of x, y with respect to $\subseteq$.
2. For $\subseteq$-incomparable x, y, the construction guarantees that $T_{\supseteq x} \cup T_{\supseteq y} \subset T_s$, and thus $x \cup y$ can be computed from T_s.

It follows that $(\mathbb{N}, \cup)$ is a computable join semilattice.

To show that $(\mathbb{N}, \cup)$ is not punctually presentable, assume otherwise. Let $(\mathbb{N}, \cup')$ be a punctual copy of $(\mathbb{N}, \cup)$, and let $x \subseteq' y \Leftrightarrow x \cup' y = y$. Hence, the relation $x \subseteq' y$ is primitive recursive. But then $(\mathbb{N}, \subseteq')$ is a punctual copy of $\mathcal{T}$ which is impossible by Theorem 8.

An analogous proof works for $(\mathbb{N}, \cap^*)$, the meet semilattice induced by $\mathcal{T}^* = (\mathbb{N}, \subseteq^*)$. ◀

Proof of Corollary 9(ii). Let $\mathcal{L}_- = (\mathbb{N} \setminus \{0\}, \sqsubseteq)$ be a punctual poset tree satisfying the following condition: for all $x, y \in \mathbb{N} \setminus \{0\}$, $x \sqsubseteq y \Leftrightarrow x - 1 \subseteq y - 1$. We observe that $\mathcal{T} \cong \mathcal{L}_-$ via primitive recursive isomorphism $\varphi : \mathbb{N} \to \mathbb{N} \setminus \{0\}$. The root of $\mathcal{L}_-$ is equal to 1.

Let $\mathcal{L} = (\mathbb{N}, \sqsubseteq)$ be such that $\mathcal{L}_-$ is a submodel of $\mathcal{L}$ and $0 \sqsubseteq x$, for all $x \in \mathbb{N}$ and $x \not\sqsubseteq 0$, for all $x \in \mathbb{N} \setminus \{0\}$. It is easy to see that $\mathcal{L}$ is a computable partial order. Moreover, $\mathcal{L}$ is a lattice. The join of $x, y \in \mathbb{N} \setminus \{0\}$ in $\mathcal{L}$ is the same as in $\mathcal{L}_-$. The join of $0, x$, where $x \in \mathbb{N}$, is x. The meet of x, y such that $x \sqsubseteq y$ is x. The meet of x, y such that $x \not\sqsubseteq y, y \not\sqsubseteq x$ is 0. It is clear that the corresponding algebraic structure $(\mathbb{N}, \sqcup, \sqcap)$ where $\sqcup, \sqcap$ are the join and the meet functions of $\mathcal{L}$, respectively, is computable.

Observe that $\mathcal{L}$ is a complemented lattice. The integer 1 is the greatest element of $\mathcal{L}$ and 0 is the least. We show that each $a \in \mathcal{L}$ has a complement, that is, b such that $a \cup b = 1$ and $a \cap b = 0$. The construction of Theorem 8 guarantees that 0 is a branching node in $\mathcal{T}$, and thus 1 is a branching node in $\mathcal{L}$. Let l, r be the two children of 1 in $\mathcal{L}$. 0 and 1 are complements of each other. Assume that $a \notin \{0, 1\}$. Either $a \sqsubseteq l$ or $a \sqsubseteq r$. If $a \sqsubseteq l$, $a \cup r = 1$ and $a \cap r = 0$. Similarly, if $a \sqsubseteq r$, $a \cup l = 1$ and $a \cap l = 0$.

Next, we prove that $(\mathbb{N}, \sqcup, \sqcap)$ is not punctually presentable. Towards a contradiction, let $\mathcal{L}' = (\mathbb{N}, \sqcup', \sqcap')$ be a punctual copy of $\mathcal{L} = (\mathbb{N}, \sqcup, \sqcap)$. Let $(\mathbb{N}, \sqsubseteq')$ be the order-theoretic counterpart of $\mathcal{L}'$, i.e., $\sqcup'$ and $\sqcap'$ are the join and meet of $\mathcal{L}'$, respectively. Let $0_{\mathcal{L}'}$ be the least element of $\mathcal{L}'$. It is clear that $(\mathbb{N} \setminus \{0_{\mathcal{L}'}\}, \sqsubseteq') \cong \mathcal{T}$. Now, given a primitive recursive bijection $f : \mathbb{N} \to \mathbb{N} \setminus \{0_{\mathcal{L}'}\}$, we define a punctual structure $\mathcal{T}' = (\mathbb{N}, \subseteq')$ as follows: $x \subseteq' y \Leftrightarrow f(x) \sqsubseteq' f(y)$. The primitive recursiveness of $\subseteq'$ follows from the following equivalences: $x \subseteq' y \Leftrightarrow f(x) \sqsubseteq' f(y) \Leftrightarrow f(x) \cup' f(y) = f(y)$ and the fact that $\cup'$ is primitive recursive. Hence $\mathcal{T}'$ is a punctual copy of $\mathcal{T}$ which contradicts Theorem 8. ◀

4 Modal Algebras

In this section, we mainly follow notations from the monographs [21, 33]. Boolean algebras are viewed as algebraic structures in the signature $\{\cup, \cap, \mathrm{C}, 0, 1\}$. Informally speaking, one can think of the signature functions as the usual set-theoretic operations: union, intersection, and complement. In addition, 0 is a least element, and 1 is a greatest element.

Let $\mathcal{B}$ be a Boolean algebra. A function $f\colon \mathcal{B} \to \mathcal{B}$ is called a *modality* if f satisfies the following two properties:

- $f(a \cup b) = f(a) \cup f(b)$ for all $a, b \in \mathcal{B}$,
- $f(0_{\mathcal{B}}) = 0_{\mathcal{B}}$.

Informally, one can view $f(x)$ as the result of applying the possibility operator $\Diamond = f$ to the "statement" x. If f is a modality, then the structure $(\mathcal{B}, f)$ is called a *modal algebra*. Here we prove the following result:

▶ **Theorem 10.** *The class of modal algebras is not punctually robust.*

The proof idea is similar to the one for non-punctual robustness of torsion abelian groups (Theorem 3.2 in [27]). We build a computable modal algebra $\mathcal{A}^*$. Given a punctual "adversary" structure $\mathcal{P}_e$ (in the signature of modal algebras), we want $\mathcal{P}_e$ to show an element c_e with some specific properties. Namely, either c_e generates (via the function $f_{\mathcal{P}_e}$) an infinite substructure inside $\mathcal{P}_e$, or c_e is a part of an $f_{\mathcal{P}_e}$-cycle of some finite size N_e. In the first case, we will win "automatically", since our $\mathcal{A}^*$ will not contain elements generating infinite substructures. In the second case, we try to prevent some fixed prime factor p of N_e from appearing in the possible sizes of the $f_{\mathcal{A}^*}$-cycles. Preventing such p from appearing is quite a delicate technical task which is achieved via careful "algebraic-flavored" arrangements. In each of the two cases, we ensure that $\mathcal{A}^* \not\cong \mathcal{P}_e$. As usual, an appropriately organized priority construction allows to successfully deal with a whole computable sequence of adversaries.

We note that the construction for torsion abelian groups (from [27]) is a finite injury construction, while our construction of a modal algebra is injury-free. There are also some important differences related to the algebraic properties: for example, in the implemented construction, our R_e-strategy has to work with a specifically selected tuple $\bar{a} = a_0, a_1, \ldots, a_M$ of witnesses (in place of just one element c_e that was used in the proof idea), and this $\vec{a}$ is selected based on the Boolean algebra specifics.

Before proceeding to the formal proof of Theorem 10, we give the necessary algebraic preliminaries. For a Boolean algebra $\mathcal{B}$, its ordering $\leq_{\mathcal{B}}$ is defined in a standard way: $x \leq_{\mathcal{B}} y$ if and only if $x \cup y = y$. An element $a \in \mathcal{B}$ is an *atom* if a is a minimal non-zero element in $\mathcal{B}$, i.e., $0 <_{\mathcal{B}} a$ and there is no b with $0 <_{\mathcal{B}} b <_{\mathcal{B}} a$.

By $Atom(\mathcal{B})$ we denote the set of atoms of $\mathcal{B}$. The *Fréchet ideal* $Fr(\mathcal{B})$ is the ideal of $\mathcal{B}$ generated by the set $Atom(\mathcal{B})$. Notice that the ideal $Fr(\mathcal{B})$ contains precisely the finite sums of atoms.

Let a be an element from a Boolean algebra $\mathcal{B}$. We put $a^0 = \mathrm{C}(a)$ and $a^1 = a$.

Let $\bar{a} = a_0, a_1, \ldots, a_n$ be a tuple of elements from $\mathcal{B}$, and let $\bar{\varepsilon} = (\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_n) \in \{0,1\}^{n+1}$. We define $\bar{a}^{\bar{\varepsilon}} = a_0^{\varepsilon_0} \cap a_1^{\varepsilon_1} \cap \cdots \cap a_n^{\varepsilon_n}$. The following fact is well-known (see, e.g., Exercise 6 in § 1.2 of [21]):

▶ **Lemma 19.** *Let $\mathcal{B}$ be a Boolean algebra, and let $\bar{a} = a_0, \ldots, a_n$ be a tuple from $\mathcal{B}$. Then the subalgebra $gr_{\mathcal{B}}(\bar{a})$ generated by the set $\{a_0, \ldots, a_n\}$ contains precisely the finite sums of the elements $\bar{a}^{\bar{\varepsilon}}$, $\varepsilon \in \{0,1\}^{n+1}$.*

In addition, if $a_i \notin \{0_{\mathcal{B}}, 1_{\mathcal{B}}\}$ and $a_i \neq a_j$ for $i \neq j$, then the finite subalgebra $gr_{\mathcal{B}}(\bar{a})$ has at least $n+2$ atoms. (Notice that an atom of the subalgebra $gr_{\mathcal{B}}(\bar{a})$ is not necessarily an atom of the original algebra $\mathcal{B}$.)

It is not hard to prove the following ancillary result:

▶ **Lemma 20.** *Let $\mathcal{B}$ be an infinite Boolean algebra, and let g be an arbitrary map from the set $Atom(\mathcal{B})$ to $\mathcal{B}$. Then the function*

$$F_{[g]}(x) = \begin{cases} 0_{\mathcal{B}}, & \text{if } x = 0_{\mathcal{B}}, \\ g(a_0) \cup g(a_1) \cup \cdots \cup g(a_n), & \text{if } x = a_0 \cup a_1 \cup \cdots \cup a_n,\ a_i \in Atom(\mathcal{B}), \\ 1_{\mathcal{B}}, & \text{if } x \notin Fr(\mathcal{B}), \end{cases}$$

is a modality. In addition, if the algebra $\mathcal{B}$ is computable, and the sets $Atom(\mathcal{B})$ and $Fr(\mathcal{B})$ are computable, and the function g is computable, then the modality $F_{[g]}$ is also computable.

▶ **Definition 21** (forward orbit)**.** *Let $(\mathcal{B}, f)$ be a modal algebra. For an element $a \in \mathcal{B}$, the* forward orbit *of a (with respect to f) is the set $FOrb(a) = \{f^{(n)}(a) : n \in \mathbb{N}\}$.*

Now we are ready to start the proof of Theorem 10. By $B(\mathbb{N})$ we denote the Boolean algebra of all finite and cofinite subsets of $\mathbb{N}$. Beforehand, we choose our underlying computable Boolean algebra $\mathcal{B}$ as follows. The algebra $\mathcal{B}$ is a computable isomorphic copy of the direct product $B(\mathbb{N}) \times B(\mathbb{N})$ such that the sets $Atom(\mathcal{B})$ and $Fr(\mathcal{B})$ are computable.

In what follows, we identify $\mathcal{B}$ with $B(\mathbb{N}) \times B(\mathbb{N})$, and we use the following notations:

- $\top_0 = (1_{B(\mathbb{N})}, 0_{B(\mathbb{N})})$ and $\top_1 = (0_{B(\mathbb{N})}, 1_{B(\mathbb{N})})$.
- Let $\{w_i : i \in \mathbb{N}\}$ be a computable list of all atoms of $B(\mathbb{N})$. We put $u_i = (w_i, 0_{B(\mathbb{N})})$ and $v_i = (0_{B(\mathbb{N})}, w_i)$.

It is clear that $\{u_i, v_i : i \in \mathbb{N}\}$ is a computable list of all atoms of $\mathcal{B}$. In addition, we have $u_i <_{\mathcal{B}} \top_0$ and $v_i <_{\mathcal{B}} \top_1$ for all $i \in \mathbb{N}$.

By Lemma 20, it is sufficient for us to construct a computable map $g\colon Atom(\mathcal{B}) \to \mathcal{B}$. Then the desired computable modal algebra $\mathcal{A}^*$ is defined as

$$\mathcal{A}^* = (\mathcal{B}, F_{[g]}). \tag{6}$$

4.1 Preparations for the Construction of g

Firstly, we describe the *basic module* of the construction of the map g, we call this module:

Adding a p-cycle to the map g. Given an odd prime number p, we find the least $i \in \mathbb{N}$ such that the value $g(u_i)$ is not yet defined. We also choose the least $j \in \mathbb{N}$ such that $g(v_j)$ is undefined.

Let $k = \lfloor \frac{p}{2} \rfloor$. For $0 \leq m < k$, we put: $g(u_{i+m}) = v_{j+m}$, $g(v_{j+m}) = u_{i+m+1}$, and $g(u_{i+k}) = u_i$.

We also say that the set $\{u_{i+m} : m \leq k\} \cup \{v_{j+\ell} : \ell < k\}$ is a *p-cycle*. This concludes the description of the basic module.

We will ensure the following property of our (future) construction:

(#) For each odd prime p, we add at most one p-cycle to g.

Property (#) is enough to prove the following useful lemma about the cardinalities of forward orbits.

▶ **Lemma 22.** *Suppose that the modal algebra $\mathcal{A}^*$ from Eq. (6) satisfies Property (#). In addition, assume that there are infinitely many primes p such that we have added a p-cycle to g. Let $a \in \mathcal{A}^*$ and $a \neq 0_{\mathcal{B}}$. Then the element a satisfies precisely one of the following three cases:*

1. $a \notin Fr(\mathcal{B})$ *and* $F_{[g]}(a) = 1_{\mathcal{B}}$.
2. $a \in Fr(\mathcal{B})$ *and* $F_{[g]}(a) = a$.
3. $a \in Fr(\mathcal{B})$ *and* $\mathrm{card}(FOrb(a)) = q_1 \cdot q_2 \cdot \ldots \cdot q_n$ *for some* $n \geq 1$ *and some odd primes* q_i *such that* $q_i \neq q_j$ *for* $i \neq j$. *(Note that here a* q_i*-cycle has been added to* g *at some stage of the construction.) In addition, if* $0 \leq i < j < \mathrm{card}(FOrb(a))$, *then* $F^{(i)}_{[g]}(a) \neq F^{(j)}_{[g]}(a)$.

Furthermore, if $a \in Fr(\mathcal{B})$, $a \neq 0_{\mathcal{B}}$ *and* $a \leq_{\mathcal{B}} \top_k$ *for some* $k \in \{0, 1\}$, *then the element* a *satisfies Case 3.*

Proof. If $a \notin Fr(\mathcal{B})$, then Lemma 20 defines $F_{[g]}(a) = 1_{\mathcal{B}}$. Thus, assume that $a \in Fr(\mathcal{B})$ and $F_{[g]}(a) \neq a$.

By Property (#), for each prime p the algebra $\mathcal{A}^*$ contains at most one p-cycle. We choose all prime numbers $q_1, q_2, \ldots, q_n$ such that:

- there exists an atom $w \in Atom(\mathcal{B})$ such that $w \leq_{\mathcal{B}} a$ and w belongs to the (unique) q_i-cycle; and
- there exists $w' \in Atom(\mathcal{B})$ such that $w' \not\leq_{\mathcal{B}} a$ and w' belongs to the q_i-cycle.

Notice that (at least one) such prime q_i exists. Indeed, if there are no such primes q_i, then the element a satisfies the following condition:

if an atom w belongs to a q-cycle and $w \leq_{\mathcal{B}} a$, then *every* atom from this q-cycle lies $\leq_{\mathcal{B}}$-below a.

This condition implies that $F_{[g]}(a)$ must be equal to a (which contradicts our assumption).

Define $M = q_1 \cdot q_2 \cdot \ldots \cdot q_n$. Then a straightforward computation shows the following: $F^{(M)}_{[g]}(a) = a$ and for all i, j we have: if $0 \leq i < j < M$, then $F^{(i)}_{[g]}(a) \neq F^{(j)}_{[g]}(a)$. Hence, $\mathrm{card}(FOrb(a)) = M$. We deduce that every element $a \neq 0_{\mathcal{B}}$ satisfies one of the three cases of the lemma.

Now suppose that $a \in Fr(\mathcal{B}) \setminus \{0_{\mathcal{B}}\}$ and $a \leq_{\mathcal{B}} \top_0$. Then for some $i \in \mathbb{N}$, we have $u_i \in \mathrm{Atom}(\mathcal{B})$ and $u_i \leq_{\mathcal{B}} a$. On the other hand, for all $j \in \mathbb{N}$, we have $v_j \in Atom(\mathcal{B})$ and $v_j \not\leq_{\mathcal{B}} a$. By choosing v_j from the q-cycle of the atom u_i, we deduce that a must satisfy Case 3. (If $a \in Fr(\mathcal{B}) \setminus \{0_{\mathcal{B}}\}$ and $a \leq_{\mathcal{B}} \top_1$, then one can make a similar argument.) Lemma 22 is proven. ◀

4.2 Requirements and the Construction of g

Observe the following: the modal algebra $\mathcal{A}^*$ from Eq. (6) has a punctual copy if and only if the structure $(\mathcal{A}^*, \top_0, \top_1)$ has a punctual copy. Thus, we fix a uniformly computable list $(\mathcal{P}_e)_{e \in \mathbb{N}}$ containing all punctual structures in the signature $\{\cup, \cap, \mathrm{C}, 0, 1, f\} \cup \{\top_0, \top_1\}$. We satisfy the following requirements:

$\boldsymbol{R_e}$ **:** The structure $(\mathcal{A}^*, \top_0, \top_1)$ is not isomorphic to $\mathcal{P}_e$.

In what follows, we will abuse the notations: we identify the structures $\mathcal{A}^*$ and $(\mathcal{A}^*, \top_0, \top_1)$.

At a stage s, we will have a finite list of *active requirements*. Each active requirement R_e possesses a corresponding witness $c_e \in \mathcal{P}_e$. Active requirements may be (forever) *deactivated*. In addition, at a stage s we will have at most one requirement R_{e_0} *on the alert*. The intended (full) life-cycle of a given requirement R_e is as follows:

inactive $\mapsto$ on the alert $\mapsto$ active $\mapsto$ deactivated.

Along the construction, we always do the following background *monitoring procedure*. At a stage s, for each $\mathcal{P}_e$ with $e \leq s$, we consider the finite set $S_{e,s} = \{0_{\mathcal{P}_e}, 1_{\mathcal{P}_e}, \top_{0,\mathcal{P}_e}, \top_{1,\mathcal{P}_e}\} \cup \{x : x \leq_{\mathbb{N}} s\}$. Assume that at the stage s we have witnessed one of the following conditions:

(a) Some of the elements from the set

$$X_{e,s} = \left\{\bar{a}^{\bar{\varepsilon}} : \bar{a} = (a_1, \ldots, a_n),\ 1 \leq n \leq \mathrm{card}(S_{e,s}),\ a_i \in S_{e,s},\ \bar{\varepsilon} \in \{0,1\}^n\right\}$$

do not satisfy the axioms of Boolean algebras or the axioms of modal algebras. (Notice that here the set $X_{e,s}$ is the "potential" Boolean subalgebra $gr_{\mathcal{P}_e}(S_{e,s})$ generated by the set $S_{e,s}$ inside $\mathcal{P}_e$.)

(b) $\top_{0,\mathcal{P}_e} \cup \top_{1,\mathcal{P}_e} \neq 1_{\mathcal{P}_e}$ or $\top_{0,\mathcal{P}_e} \cap \top_{1,\mathcal{P}_e} \neq 0_{\mathcal{P}_e}$ or $f_{\mathcal{P}_e}(\top_{0,\mathcal{P}_e}) \neq 1_{\mathcal{P}_e}$ or $f_{\mathcal{P}_e}(\top_{1,\mathcal{P}_e}) \neq 1_{\mathcal{P}_e}$.

(c) There exist $k \in \{0,1\}$ and $x, y \in S_{e,s} \setminus \{0_{\mathcal{P}_e}, \top_{k,\mathcal{P}_e}\}$ such that $x \cup y = \top_{k,\mathcal{P}_e}$, $x \cap y = 0_{\mathcal{P}_e}$, and

- either $f_{\mathcal{P}_e}(x) \neq 1_{\mathcal{P}_e}$ and $f_{\mathcal{P}_e}(y) \neq 1_{\mathcal{P}_e}$, or
- $f_{\mathcal{P}_e}(x) = 1_{\mathcal{P}_e}$ and $f_{\mathcal{P}_e}(y) = y$, or
- $f_{\mathcal{P}_e}(x) = 1_{\mathcal{P}_e}$ and $1_{\mathcal{P}_e} \in \{f^{(m)}_{\mathcal{P}_e}(y) : m \leq s\}$.

(d) There exist $k \in \{0,1\}$ and $x, y \in S_{e,s}$ such that $x \cap y = 0_{\mathcal{P}_e}$, $x \leq_{\mathcal{P}_e} \top_{k,\mathcal{P}_e}$, $y \leq_{\mathcal{P}_e} \top_{k,\mathcal{P}_e}$, and $f_{\mathcal{P}_e}(x) = f_{\mathcal{P}_e}(y) = 1_{\mathcal{P}_e}$.

Then we declare the requirement R_e deactivated. Indeed, in this case the structure $\mathcal{P}_e$ cannot be isomorphic to $(\mathcal{A}^*, \top_0, \top_1)$. To observe this non-isomorphism for Condition (c) above, we recall the following fact: if $x \cup y = \top_0$, $x \cap y = 0_{\mathcal{B}}$ and $x, y \notin \{0_{\mathcal{B}}, \top_0\}$, then precisely one of the elements $b \in \{x, y\}$ satisfies $b \notin Fr(\mathcal{B})$ and $F_{[g]}(b) = 1_{\mathcal{B}}$. By Lemma 22, the remaining element $a \in \{x, y\}$ satisfies $a \in Fr(\mathcal{B})$ and $F_{[g]}(a) \neq a$. In addition, we have $1_{\mathcal{B}} \notin FOrb(a)$.

To observe non-isomorphism for Condition (d), we recall the following: if $x \leq_{\mathcal{B}} \top_0$ and $F_{[g]}(x) = 1_{\mathcal{B}}$, then $x \notin Fr(\mathcal{B})$ and every $y \leq_{\mathcal{B}} \top_0 \cap \mathrm{C}(x)$ satisfies $y \in Fr(\mathcal{B})$ and $F_{[g]}(y) \neq 1_{\mathcal{B}}$.

Due to the described monitoring procedure, in the main construction given below, we may assume (without loss of generality) that every considered structure $\mathcal{P}_e$ has the following properties, for $k \in \{0,1\}$:

(P.0) The reduct of $\mathcal{P}_e$ to the signature $\{\cup, \cap, \mathrm{C}, 0, 1, f\}$ is a punctual modal algebra. In addition, $\top_{0,\mathcal{P}_e} \cup \top_{1,\mathcal{P}_e} = 1_{\mathcal{P}_e}$, $\top_{0,\mathcal{P}_e} \cap \top_{1,\mathcal{P}_e} = 0_{\mathcal{P}_e}$, and $f_{\mathcal{P}_e}(\top_{0,\mathcal{P}_e}) = f_{\mathcal{P}_e}(\top_{1,\mathcal{P}_e}) = 1_{\mathcal{P}_e}$.

(P.1) If $y \leq_{\mathcal{P}_e} \top_{k,\mathcal{P}_e}$, $y \neq 0_{\mathcal{P}_e}$ and $f_{\mathcal{P}_e}(y) \neq 1_{\mathcal{P}_e}$, then $f_{\mathcal{P}_e}(y) \neq y$ and $1_{\mathcal{P}_e} \notin FOrb_{\mathcal{P}_e}(y)$.

(P.2) If $x_1, x_2, \ldots, x_n \leq_{\mathcal{P}_e} \top_{k,\mathcal{P}_e}$ and $x_i \cap x_j = 0_{\mathcal{P}_e}$ for $i \neq j$, then *at most one* element $y \in \{x_1, x_2, \ldots, x_n\}$ satisfies $f_{\mathcal{P}_e}(y) = 1_{\mathcal{P}_e}$.

Now we are ready to describe the main construction. Let $(p_s)_{s \geq 1}$ be the increasing list of all odd prime numbers.

Construction. At stage 0, there are no active requirements. We declare that the requirement R_0 is on the alert.

Stage $s > 0$. Roughly speaking, the main goal of the stage s is to decide whether to add a p_s-cycle to the map g. In addition, our actions will ensure the following property:

(†) Suppose that by the end of the stage s, an active requirement R_e has a witness $c_e \in \mathcal{P}_e$. Then c_e satisfies one of the following two conditions:

- either the forward orbit $FOrb_{\mathcal{P}_e}(c_e)$ is infinite, or
- the set $FOrb_{\mathcal{P}_e}(c_e)$ is finite and the following implication holds: if $r = \mathrm{card}(FOrb_{\mathcal{P}_e}(c_e))$ has a form $r = q_1 \cdot q_2 \cdot \ldots \cdot q_n$, where $n \geq 1$ and $2 < q_1 < q_2 < \cdots < q_n$ are prime numbers, then some prime $q > p_s$ must divide r.

Intuitively speaking, the choice of such form $r = q_1 \cdot q_2 \cdot \ldots \cdot q_n$ is dictated by Lemma 22. If the decomposition of r has any other form, then Lemma 22 ensures that $\mathcal{A}^* \ncong \mathcal{P}_e$.

If the structure $\mathcal{A}^*$ does not contain a p_s-cycle by the end of the stage s, then we say that the prime p_s is *forbidden* from entering the structure $\mathcal{A}^*$.

Our actions at the stage s go as follows. If there is a requirement R_{e_0} which is currently on the alert, then firstly we execute the following strategy.

4.2.1 Strategy for R_e Which Is on the Alert

Let $\mathcal{D}_s$ be the Boolean subalgebra of $\mathcal{B}$ generated by the following elements: $\top_0$, $\top_1$, and the elements of all q-cycles added to g at stages $t < s$. By Lemma 19, the algebra $\mathcal{D}_s$ is finite, and one can computably recover this structure $\mathcal{D}_s$. In addition, Lemma 20 ensures that the values $F_{[g]}(x)$ are defined for all $x \in \mathcal{D}_s$.

Define $M = \mathrm{card}(\mathcal{D}_s)$. Consider the $\leq_{\mathbb{N}}$-least elements $b_0 <_{\mathbb{N}} b_1 <_{\mathbb{N}} \cdots <_{\mathbb{N}} b_{M-1}$ from the structure $\mathcal{P}_e$ such that $b_i \notin \{0_{\mathcal{P}_e}, 1_{\mathcal{P}_e}, \top_{0,\mathcal{P}_e}, \top_{1,\mathcal{P}_e}\}$. We define the following finite Boolean subalgebra of $\mathcal{P}_e$: $\mathcal{Q}_s = gr_{\mathcal{P}_e}(\{\top_{0,\mathcal{P}_e}, \top_{1,\mathcal{P}_e}, b_0, b_1, \ldots, b_{M-1}\})$.

By Lemma 19, the algebra $\mathcal{Q}_s$ has at least $M+3$ atoms. Applying Property (P.2) of the construction, we deduce that at most two of these atoms x satisfy $f_{\mathcal{P}_e}(x) = 1_{\mathcal{P}_e}$ (indeed, at most one atom below $\top_{0,\mathcal{P}_e}$, and at most one atom below $\top_{1,\mathcal{P}_e}$). Therefore, among the atoms of $\mathcal{Q}_s$ we can find $(M+1)$-many pairwise distinct elements $a_0, a_1, \ldots, a_M$ such that $f_{\mathcal{P}_e}(a_i) \neq 1_{\mathcal{P}_e}$, $a_i \neq 0_{\mathcal{P}_e}$, and $a_i \cap a_j = 0_{\mathcal{P}_e}$ for $i \neq j$.

By Property (P.1), we obtain that $f_{\mathcal{P}_e}(a_i) \neq a_i$ and $1_{\mathcal{P}_e} \notin FOrb_{\mathcal{P}_e}(a_i)$. We define $L = \prod_{j=1}^{s} p_j$. We compute the values $f^{(j)}_{\mathcal{P}_e}(a_i)$, for $i \leq M$ and $j \leq L$. Then one of the following five cases is satisfied:

Case (i.a). There exists a_i such that the function $f_{\mathcal{P}_e}$ is not injective on the set $\{f^{(j)}_{\mathcal{P}_e}(a_i) : j \leq L\}$ (i.e., there exist $j_1 < j_2 \leq L$ such that $f^{(j_1)}_{\mathcal{P}_e}(a_i) \neq f^{(j_2)}_{\mathcal{P}_e}(a_i)$ and $f^{(j_1+1)}_{\mathcal{P}_e}(a_i) = f^{(j_2+1)}_{\mathcal{P}_e}(a_i)$).

Then we (safely) declare the requirement R_e deactivated. Indeed, by item (3) of Lemma 22, the structure $\mathcal{P}_e$ cannot be isomorphic to our structure $\mathcal{A}^*$. In all cases (i.X) below, we will assume that the function $f_{\mathcal{P}_e}$ is injective on the set $\{f^{(j)}_{\mathcal{P}_e}(a_i) : j \leq L\}$.

Case (i.b). There exists a_i with the following properties:

- $FOrb(a_i) \subseteq \{f^{(j)}_{\mathcal{P}_e}(a_i) : j \leq L\}$ and $N = \mathrm{card}(FOrb(a_i))$.
- Consider the prime decomposition $N = q_1^{\alpha_1} \cdot q_2^{\alpha_2} \cdot \ldots \cdot q_\ell^{\alpha_\ell}$, where $q_j \neq q_k$ for $j \neq k$ and $\alpha_j \geq 1$. For some $j \leq \ell$, either q_j has already been forbidden from entering $\mathcal{A}^*$, or $q_j = 2$, or $\alpha_j \geq 2$.

Then by Lemma 22, our structure $\mathcal{A}^*$ cannot be isomorphic to $\mathcal{P}_e$, since $\mathcal{A}^*$ does not contain elements $b \in Fr(\mathcal{B})$ with $\mathrm{card}(FOrb(b)) = N$. We declare the requirement R_e deactivated.

Case (i.c). Neither of Cases (i.a) and (i.b) is satisfied, and there exists a_i with the following properties:

- $FOrb(a_i) \subseteq \{f^{(j)}_{\mathcal{P}_e}(a_i) : j \leq L\}$ and $N = \mathrm{card}(FOrb(a_i))$.
- For some $t \geq s$, the prime p_t divides N.

We declare the requirement R_e active, and we set $c_e = a_i$. Note the following: if $t > s$, then R_e will definitely satisfy Property (†) at the end of stage s.

Case (i.d). Neither of Cases (i.a)–(i.c) is satisfied, and there exists a_i such that for all $j < k \leq L$, we have $f^{(j)}_{\mathcal{P}_e}(a_i) \neq f^{(k)}_{\mathcal{P}_e}(a_i)$. Notice that this condition is equivalent to the condition $\mathrm{card}(FOrb(a_i)) \geq L+1$. We declare the requirement R_e active, and we define $c_e = a_i$.

Case (i.e). Suppose that neither of Cases (i.a)–(i.d) is satisfied. Then *every* a_i, $i \leq M$, has the following properties:

(e.1) $FOrb(a_i) \subseteq \{f^{(j)}_{\mathcal{P}_e}(a_i) : j \leq L\}$ and $N_i = \mathrm{card}(FOrb(a_i)) \leq L$.

(e.2) The number N_i has prime decomposition $N_i = q_{i,1} \cdot \ldots \cdot q_{i,\ell_i}$, where $q_{i,j} \neq q_{i,k}$ for $j \neq k$, and $3 \leq q_{i,j} < p_s$ and $q_{i,j}$ has not been forbidden from entering $\mathcal{A}^*$.

We show that in this case the structure $\mathcal{P}_e$ is not isomorphic to $\mathcal{A}^*$. In order to prove this, it is enough to establish the following fact:

▷ Claim 23. $\mathcal{A}^*$ does not contain $(M+1)$-many pairwise disjoint elements a satisfying:

$$\mathrm{card}(FOrb(a)) = N_i \text{ for some } N_i \text{ with Property (e.2).} \tag{7}$$

Proof. Suppose that an element $a \in \mathcal{A}^*$ satisfies Eq. (7). Consider the prime number $q_{i,1}$. By the proof of Lemma 22, there exists an atom $a' \in Atom(\mathcal{B})$ such that $a' \leq_{\mathcal{B}} a$ and a' belongs to a $q_{i,1}$-cycle. By the definition of the finite algebra $\mathcal{D}_s$, we have $a' \in \mathcal{D}_s$. We define $\theta(a) = a'$. We notice the following fact: if $a \cap b = 0_{\mathcal{B}}$, then $\theta(a) \neq \theta(b)$. Indeed, if $\theta(a) = \theta(b)$, then $0_{\mathcal{B}} \neq \theta(a) \leq_{\mathcal{B}} a \cap b$.

Now, towards a contradiction, assume that $\mathcal{A}^*$ contains $(M+1)$-many pairwise disjoint elements a satisfying Eq. (7). Then the algebra $\mathcal{D}_s$ contains at least $(M+1)$-many pairwise distinct elements $\theta(a)$. This contradicts the choice of $M = \mathrm{card}(\mathcal{D}_s)$. ◁

Indeed, recall that $a_i \cap a_j = 0_{\mathcal{P}_e}$ for $i \neq j$. Hence, the structure $\mathcal{P}_e$ *does contain* $(M+1)$-many disjoint elements $a_0, \ldots, a_M$ satisfying Eq. (7). Thus, Claim 23 implies that $\mathcal{P}_e \ncong \mathcal{A}^*$. Therefore, in Case (i.e), we safely declare the requirement R_e deactivated.

This concludes the description of the strategy for $R_e = R_{e_0}$ which is on the alert. After executing this strategy, one-by-one, we execute strategies for the currently active requirements $R_{e'}$ (see below). Notice that in Cases (i.c) and (i.d) above, this execution also includes our requirement R_{e_0} (R_{e_0} had been on the alert, but then it moved to becoming active).

4.2.2 Strategy for an Active Requirement R_e

As usual, here we assume the following: if the requirement R_e was active at the end of the previous stage $s-1$, then it satisfied Property (†) at that moment. Consider the witness c_e for R_e. Recall that

$$L = \prod_{j=1}^{s} p_j. \tag{8}$$

We compute the values $f^{(i)}_{\mathcal{P}_e}(c_e)$, for all $i \leq L$. One of the following two cases is satisfied:

Case (ii.a). Suppose that $f^{(j)}_{\mathcal{P}_e}(c_e) = f^{(k)}_{\mathcal{P}_e}(c_e)$ for some $j < k \leq L$. Then we have $FOrb_{\mathcal{P}_e}(c_e) \subseteq \{f^{(i)}_{\mathcal{P}_e}(c_e) : i \leq L\}$. In particular, the set $FOrb_{\mathcal{P}_e}(c_e)$ is finite. For $N = \mathrm{card}(FOrb_{\mathcal{P}_e}(c_e))$, we compute its prime decomposition $N = q_1^{\alpha_1} \cdot q_2^{\alpha_2} \cdot \ldots \cdot q_\ell^{\alpha_\ell}$.

Similarly to Case (i.b), if for some $i \leq \ell$ either q_i was already forbidden, or $q_i = 2$, or $\alpha_i \geq 2$, then $\mathcal{P}_e \ncong \mathcal{A}^*$. Thus, we declare such R_e deactivated. Hence, in what follows we may assume that $N = q_1 \cdot q_2 \cdot \ldots \cdot q_\ell$, where $2 < q_1 < q_2 < \cdots < q_\ell$.

If some q_i equals p_s, then we *forbid* adding a p_s-cycle to the map g. Since $\mathcal{A}^*$ will not have p_s-cycles, by Lemma 22, we deduce that $\mathcal{P}_e$ is not isomorphic to $\mathcal{A}^*$. We safely declare the requirement R_e deactivated.

If $p_s \notin \{q_i : i \leq \ell\}$, then R_e stays active. Here we claim that some q_i *must be equal* to p_t for some $t > s$. Indeed, if R_e was already active at the stage $s-1$, then this fact is guaranteed by the "$(s-1)$-version" of Property (†). Otherwise, R_e moved from being on the alert to being active at the stage s. But then this move was triggered by Case (i.c), and some p_t with $t > s$ must divide N. We observe that in Case (ii.a), the requirement R_e will satisfy Property (†) at the end of the stage s.

Case (ii.b). Otherwise, $f^{(j)}_{\mathcal{P}_e}(c_e) \neq f^{(k)}_{\mathcal{P}_e}(c_e)$ for all $j < k \leq L$. Then the requirement R_e stays active. Observe that here $N = \mathrm{card}(FOrb_{\mathcal{P}_e}(c_e)) \geq L + 1$.

In Case (ii.b) we need to show that the requirement R_e will still satisfy Property (†) by the end of the stage s. Towards a contradiction, assume that Property (†) fails. Then the set $FOrb_{\mathcal{P}_e}(c_e)$ is finite, and we have $N = q_1 \cdot q_2 \cdot \ldots \cdot q_\ell$, where the primes q_i satisfy $2 < q_1 < q_2 < \cdots < q_\ell \leq p_s$. By Eq. (8), we obtain that $N = q_\ell \cdot q_{\ell-1} \cdot \ldots \cdot q_2 \cdot q_1 \leq p_s \cdot p_{s-1} \cdot \ldots \cdot p_{s-(\ell-2)} \cdot p_{s-(\ell-1)} \leq L$. This contradicts the fact that $N > L$.

We deduce that in each of the two cases, an active requirement R_e will satisfy Property (†) by the end of the stage s. This concludes the description of the strategy for an active R_e.

If by the end of the stage s, no strategy has forbidden to add a p_s-cycle to g, then we proceed as follows:

- Add a p_s-cycle to the map g.
- Find the least i such that the requirement R_i has never been on the alert before. Declare this R_i being on the alert.

4.3 Verification

▶ **Lemma 24.** *For every $a \in Atom(\mathcal{B})$, the value $g(a)$ is eventually defined. Consequently, the structure $\mathcal{A}^*$ from Eq. (6) is a well-defined computable modal algebra.*

Proof. It is sufficient to show that our construction adds a p_s-cycle for infinitely many $s \geq 1$. Towards a contradiction, assume that for every $s > s_0$, the structure $\mathcal{A}^*$ never gets a p_s-cycle. Then for every $s > s_0$, the requirement R_s is never declared on the alert. Consequently, such R_s never becomes active.

Choose a large enough stage $s^* > s_0$ with the following property: if a requirement R_e, where $e \leq s_0$, is eventually declared deactivated, then R_e was deactivated *before* the stage s^*. Since the p_{s^*}-cycle is forever forbidden, this forbiddance was triggered by the following action: at the stage s^*, some currently active requirement R_i, where $i \leq s_0$, forbade p_{s^*}, and after that this R_i was deactivated. But this contradicts the choice of the stage s^*. We deduce that $g(a)$ is defined for all atoms a from $\mathcal{B}$. Lemma 24 is proved. ◀

▶ **Lemma 25.** *Every requirement R_e is satisfied.*

Proof. Since our construction adds a p_s-cycle for infinitely many s (as discussed in Lemma 24), every requirement R_e is eventually declared being on the alert. If a requirement R_e is eventually deactivated, then (as discussed in detail in the construction description) we have $\mathcal{P}_e \not\cong \mathcal{A}^*$ and R_e is satisfied.

Suppose that a requirement R_e is never deactivated. Then there exists a stage s_0 such that R_e is active at every stage $s \geq s_0$. Consider the corresponding witness $c_e \in \mathcal{P}_e$. If the forward orbit $FOrb_{\mathcal{P}_e}(c_e)$ is infinite, then by Lemma 22, $\mathcal{P}_e$ is not isomorphic to $\mathcal{A}^*$. Therefore, we may assume that the set $FOrb_{\mathcal{P}_e}(c_e)$ is finite.

Since $N = \mathrm{card}(FOrb_{\mathcal{P}_e}(c_e)) < \omega$, there exists a large enough stage $s^* \geq s_0$ such that for every $s \geq s^*$, the active requirement R_e satisfies Case (ii.a) at the stage s. The requirement R_e is never deactivated, hence, we have $N = q_1 \cdot q_2 \cdot \ldots \cdot q_\ell$ for some $\ell \geq 1$ and some primes $2 < q_1 < q_2 < \cdots < q_\ell$. But then Property (†) of the construction implies that for every $s \geq s^*$, there exists a prime $q > p_s$ such that q divides N. Hence, N has infinitely many divisors, which gives a contradiction. We conclude that for every e, the structure $\mathcal{A}^*$ is not isomorphic to $\mathcal{P}_e$. ◀

References

1 Pavel E. Alaev and Victor L. Selivanov. Polynomial computability of fields of algebraic numbers. *Doklady Mathematics*, 98(1):341–343, 2018. doi:10.1134/S1064562418050137.

2 Chris J. Ash and Julia Knight. *Computable structures and the hyperarithmetical hierarchy*. Elsevier, 2000.

3 Matthew Askes and Rod Downey. Online, computable and punctual structure theory. *Logic Journal of the IGPL*, 31(6):1251–1293, August 2022. doi:10.1093/jigpal/jzac065.

4 Nikolay Bazhenov. Categoricity spectra for polymodal algebras. *Studia Logica*, 104(6):1083–1097, 2016. doi:10.1007/S11225-016-9667-Y.

5 Nikolay Bazhenov, Rod Downey, Iskander Kalimullin, and Alexander Melnikov. Foundations of online structure theory. *Bulletin of Symbolic Logic*, 25(2):141–181, 2019. doi:10.1017/bsl.2019.20.

6 Nikolay Bazhenov, Marta Fiori-Carones, Lu Liu, and Alexander G. Melnikov. Primitive recursive reverse mathematics. *Annals of Pure and Applied Logic*, 175(1):103354, 2024. doi:10.1016/J.APAL.2023.103354.

7 Nikolay Bazhenov, Iskander Kalimullin, Alexander Melnikov, and Keng Meng Ng. Online presentations of finitely generated structures. *Theoretical Computer Science*, 844:195–216, 2020. doi:10.1016/j.tcs.2020.08.021.

8 Garrett Birkhoff. *Lattice Theory*, volume 25 of *Colloquium Publications*. American Mathematical Society, Providence, Rhode Island, 1940.

9 Achim Blumensath and Erich Grädel. Automatic structures. In *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*, pages 51–62. IEEE Computer Society, 2000. doi:10.1109/LICS.2000.855755.

10 A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 2005.

11 D. Cenzer and J. B. Remmel. Complexity theoretic model theory and algebra. In *Handbook of recursive mathematics, Vol. 1*, volume 138 of *Studies in Logic and the Foundations of Mathematics*, pages 381–513. North-Holland, Amsterdam, 1998. doi:10.1016/S0049-237X(98)80011-6.

12 Douglas Cenzer and Jeffrey Remmel. Polynomial-time abelian groups. *Annals of Pure and Applied Logic*, 56(1):313–363, 1992. doi:10.1016/0168-0072(92)90076-C.

13 Douglas Cenzer and Jeffrey B. Remmel. Polynomial-time versus recursive models. *Annals of Pure and Applied Logic*, 54(1):17–58, 1991. doi:10.1016/0168-0072(91)90008-A.

14 Douglas Cenzer and Jeffrey B. Remmel. Feasible graphs with standard universe. *Annals of Pure and Applied Logic*, 94(1):21–35, 1998. doi:10.1016/S0168-0072(97)00064-X.

15 Alexander Chagrov and Michael Zakharyaschev. *Modal Logic*. Oxford University Press, Oxford, 1997.

16 Marina Dorzhieva and Alexander G. Melnikov. Punctually presented structures I: closure theorems. *Computability*, 12(4):323–337, 2023. doi:10.3233/COM-230448.

17 Rod Downey, Noam Greenberg, Alexander Melnikov, Keng Meng Ng, and Daniel Turetsky. Punctual categoricity and universality. *The Journal of Symbolic Logic*, 85(4):1427–1466, 2020. doi:10.1017/jsl.2020.51.

18 Rod Downey, Alexander Melnikov, and Keng Meng Ng. Foundations of Online Structure Theory II: The Operator Approach. *Logical Methods in Computer Science*, 17(3):6:1–6:35, July 2021. doi:10.46298/lmcs-17(3:6)2021.

19 Rodney Downey and Alexander Melnikov. *Computable structure theory: A unified approach*. Springer, to appear. URL: https://homepages.ecs.vuw.ac.nz/~melnikal/maindoc.pdf.

20 Rodney G. Downey, Matthew Harrison-Trainor, Iskander Sh. Kalimullin, Alexander G. Melnikov, and Daniel Turetsky. Graphs are not universal for online computability. *Journal of Computer and System Sciences*, 112:1–12, 2020. doi:10.1016/J.JCSS.2020.02.004.

21 Sergei S. Goncharov. *Countable Boolean Algebras and Decidability*. Consultants Bureau, New York, 1997.

22 Erich Grädel. Automatic structures: Twenty years later. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 21–34. ACM, 2020. doi:10.1145/3373718.3394734.

23 Noam Greenberg, Matthew Harrison-Trainor, Alexander Melnikov, and Dan Turetsky. Non-density in punctual computability. *Annals of Pure and Applied Logic*, 172(9):102985, 2021. doi:10.1016/j.apal.2021.102985.

24 Serge Grigorieff. Every recursive linear ordering has a copy in DTIME-SPACE($n, log(n)$). *Journal of Symbolic Logic*, 55(1):260–276, 1990. doi:10.2307/2274966.

25 Thomas Jech. *Set Theory: The Third Millennium Edition, revised and expanded*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2007.

26 Iskander Sh. Kalimullin, Alexander G. Melnikov, and Antonio Montalbán. Punctual definability on structures. *Annals of Pure and Applied Logic*, 172(8):102987, 2021. doi:10.1016/J.APAL.2021.102987.

27 Iskander Sh. Kalimullin, Alexander G. Melnikov, and Keng Meng Ng. Algebraic structures computable without delay. *Theoretical Computer Science*, 674:73–98, 2017. doi:10.1016/J.TCS.2017.01.029.

28 Dariusz Kalociński, Luca San Mauro, and Michał Wrocławski. Punctual Presentability in Certain Classes of Algebraic Structures. In Rastislav Královič and Antonín Kučera, editors, *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, volume 306 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 65:1–65:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISSN: 1868-8969. doi:10.4230/LIPIcs.MFCS.2024.65.

29 Bakhadyr Khoussainov and Tomasz Kowalski. Computable isomorphisms of Boolean algebras with operators. *Studia Logica*, 100(3):481–496, 2012. doi:10.1007/S11225-012-9411-1.

30 Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In Daniel Leivant, editor, *Logic and Computational Complexity*, volume 960 of *Lecture Notes in Computer Science*, pages 367–392. Springer Berlin Heidelberg, 1995. doi:10.1007/3-540-60178-3_93.

31 H. A. Kierstead. Recursive and on-line graph coloring. In Yu L. Ershov, S. S. Goncharov, A. Nerode, J. B. Remmel, and V. W. Marek, editors, *Handbook of Recursive Mathematics, Vol. 2*, volume 139 of *Studies in Logic and the Foundations of Mathematics*, pages 1233–1269. Elsevier, 1998. ISSN: 0049-237X. doi:10.1016/S0049-237X(98)80051-7.

32 Donald E. Knuth. *The Art of Computer Programming. Volume I. Fundamental Algorithms*. The Art of Computer Programming. Addison-Wesley, Reading, Massachusetts, 1968.

33 Sabine Koppelberg. *Handbook of Boolean Algebras, Volume 1*. North-Holland, Amsterdam, 1989.

34 Alexander Melnikov and Keng Meng Ng. A structure of punctual dimension two. *Proceedings of the American Mathematical Society*, 148(7):3113–3128, 2020. doi:10.1090/proc/15020.

35 Alexander G. Melnikov and Keng Meng Ng. The back-and-forth method and computability without delay. *Israel Journal of Mathematics*, 234(2):959–1000, 2019. doi:10.1007/s11856-019-1948-5.

36 Anil Nerode and Jeffrey B. Remmel. Complexity-theoretic algebra II: Boolean algebras. *Annals of Pure and Applied Logic*, 44(1–2):71–99, 1989. doi:10.1016/0168-0072(89)90047-X.

37 Helena Rasiowa and Roman Sikorski. *The Mathematics of Metamathematics*. Państwowe Wydawnictwo Naukowe, Warsaw, 1963.