Computability of Heyting algebras and Distributive Lattices

Amy Turlington, Ph.D.

University of Connecticut, 2010

Distributive lattices are studied from the viewpoint of effective algebra. In particular, we also consider special classes of distributive lattices, namely pseudocomplemented lattices and Heyting algebras. We examine the complexity of prime ideals in a computable distributive lattice, and we show that it is always possible to find a computable prime ideal in a computable distributive lattice. Furthermore, for any Π^0_1 class, we prove that there is a computable (non-distributive) lattice such that the Π^0_1 class can be coded into the (nontrivial) prime ideals of the lattice. We then consider the degree spectra and computable dimension of computable distributive lattices, pseudocomplemented lattices, and Heyting algebras. A characterization is given for the computable dimension of the free Heyting algebras on finitely or infinitely many generators.

Computability of Heyting algebras and Distributive Lattices

Amy Turlington

B.S. Computer Science, James Madison University, Harrisonburg, VA, 2004

M.S. Mathematics, University of Connecticut, Storrs, CT, 2007

A Dissertation

Submitted in Partial Fullfilment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2010

APPROVAL PAGE

Doctor of Philosophy Dissertation

Computability of Heyting algebras and Distributive Lattices

Presented by

Amy Turlington, B.S., M.S.

Major Advisor

David Reed Solomon

Associate Advisor

Manuel Lerman

Associate Advisor

Alexander Russell

University of Connecticut

2010

ACKNOWLEDGEMENTS

First of all I would like to thank my advisor, Reed Solomon, for all of his help and guidance throughout the years. Even while on sabbatical in Indiana this past year, he continued to meet with me weekly over Skype. I am very grateful to have him as an advisor. Also thanks to Asher Kach for helpful conversations and for proofreading a draft of my thesis. I want to thank my officemates and fellow math grad students as well for making grad school enjoyable. I have many fond memories from our years in MSB 201. Finally, thanks to my family and my fiancé, Samuel Walsh, for all of their love and support.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Computability theory and effective algebra

We will assume that the reader is familiar with the basic concepts from computability theory (cf., e.g., [18]).

Fix an enumeration of the partial computable functions $\varphi_0, \varphi_1, \varphi_2, \ldots$, and write φ_e converges to y on input x as $\varphi_e(x)\downarrow = y$ and φ_e diverges on input x as $\varphi_e(x)\uparrow$. If $\varphi_e(x)$ converges, it does so after a finite number of steps. Denote $\varphi_e(x)$ converges to y in s steps by $\varphi_{e,s}(x)\downarrow = y$. We say that a set of natural numbers X is computable provided that its characteristic function is computable. A common example of a set which is not computable is the halting set, defined as follows.

Definition 1.1.1. The *halting set* $K = \{e \mid \varphi_e(e)\downarrow\}$.

Another important concept from computability theory is that of a Π^0_1 class. Many facts are known about the computational properties of these classes (cf., e.g., [3,9,18]).

Definition 1.1.2. A set $\mathcal{C} \subseteq \mathcal{P}(\mathbb{N})$ is a Π^0_1 class if there is a computable relation

$R(\bar{x}, X)$ on $\mathbb{N}^k \times \mathcal{P}(\mathbb{N})$ such that $Y \in \mathcal{C} \Leftrightarrow \forall \bar{x} R(\bar{x}, Y)$.

An equivalent definition is as follows, where a computable tree T is a computable subset of $2^{<\mathbb{N}}$.

Definition 1.1.3. A Π^0_1 class is the set of infinite paths through a computable subtree of $2^{<\mathbb{N}}$.

Effective algebra is an area of mathematical logic in which computability theory is used to study classical theorems on algebraic structures, such as groups, rings, or linear orders, from a computational perspective. A general overview can be found in [4] and [5]. Often in effective algebra, we consider a computable algebraic structure (such as a ring or partial order) and ask if certain theorems about the structure hold effectively. For instance, if it is known classically that an associated algebraic object exists (e.g., an ideal of a ring or a basis for a vector space), we may ask if there is an algorithm which finds this object.

Intuitively, a countable algebraic structure is computable if its domain can be identified with a computable set of natural numbers and the (finitely many) operations and relations on the structure are computable. If the structure is infinite, we usually identify its domain with ω (the natural numbers). The following definition makes the notion of a computable algebraic structure more precise and works if the language is infinite.

Definition 1.1.4. A *countable algebraic structure* $\mathcal{A}$ *over the language* $\mathcal{L}$ is *computable* if $\mathcal{L}$ is computable, the domain of the structure is ω, and its atomic diagram $\{\varphi(\bar{a}) \mid \varphi(\bar{x})$ is an atomic or negated atomic $\mathcal{L}$-formula, $\mathcal{A} \models \varphi(\bar{a})\}$ is computable.

Remark 1.1.5. *Definition 1.1.4 can be generalized to say that a countable algebraic structure has Turing degree* $\mathbf{d}$ *if the language is computable, the domain is* ω*, and the atomic diagram of the structure has degree* $\mathbf{d}$*. Alternatively, a countable algebraic structure over a finite language has degree* $\mathbf{d}$ *if the join of the degrees of its domain and all of its operations and relations is equal to* $\mathbf{d}$*.*

For instance, a computable linear order is given by a computable set $L \subseteq \mathbb{N}$ together with a computable binary operation $\prec_L$ such that $(L, \prec_L)$ satisfies the axioms for a linear order. An example is the linear order $(\mathbb{N}, \leq)$, where its elements are the natural numbers under the usual ordering $\leq$.

Given a fixed isomorphism type, there may be many different computable copies, and these copies may have different computational properties. For example, there are computable copies of the linear order $(\mathbb{N}, \leq)$ in which the successor relation is computable and computable copies in which it is not. (The successor relation is easily seen to be computable in the standard ordering $0 < 1 < 2 < \ldots$, since y is a successor of x when $y = x + 1$. However, one can construct a more complicated ordering such as $18 < 100 < 34 < 2 < \ldots$, where it is more difficult to determine if one element is a successor of another.) The atoms of a Boolean algebra are another example; there are computable Boolean algebras in which determining whether a given element is an atom is computable, and there are computable copies in which this is not computable. Therefore, the computational properties can depend on the particular computable copy.

Similarly, it is possible for two structures to be isomorphic but have different Turing degrees. For instance, a computable lattice can have an isomorphic copy with the degree of the halting set. This motivates the following two definitions.

Definition 1.1.6. The *degree spectrum* of a structure is the set of Turing degrees of all its isomorphic structures.

Definition 1.1.7. The (Turing) *degree of the isomorphism type* of a structure is the least Turing degree in its degree spectrum (if it exists).

Remark 1.1.8. *Except in trivial cases, degree spectra are always closed upwards in the Turing degrees (cf. [10]).*

It is known that for any Turing degree $\mathbf{d}$, there is a general (non-distributive) lattice whose isomorphism type has degree $\mathbf{d}$. This is also true for graphs, abelian groups, and partial orders, to name a few other examples. However, this is not the case for Boolean algebras, for if the isomorphism type of a Boolean algebra has a least degree, then it must be $\mathbf{0}$ (cf. [16]).

Another important idea in effective algebra is the concept of computable dimension. Often, it is possible to have classically isomorphic computable structures which are not computably isomorphic. In other words, although the copies themselves are computable and structurally identical, there is no algorithm to find the isomorphism between them.

Definition 1.1.9. The *computable dimension* of a computable structure is the number of (classically isomorphic) computable copies of the structure up to computable isomorphism.

If the computable dimension of a structure is 1, then every computable coding of the isomorphism type has the same computational properties. Structures with computable dimension 1 are called computably categorical.

One question of interest is to determine what the possible values for the computable dimension of a structure can be. For instance, it is known that the computable dimension of a Boolean algebra or linear order is always 1 or ω (cf. [14,15]), while the computable dimension of a graph or general (non-distributive) lattice can be n for any $n > 0$ or ω (cf. [8]).

It is also useful to have a nice characterization for the structures which are computably categorical. As an example, it is known that a computable Boolean algebra is computably categorical if and only if it has finitely many atoms (cf. [14]).

With these types of questions in mind, we will now turn our attention to distributive lattices. Coding methods can be used to show that computable general (non-distributive) lattices exhibit every possible computable model theoretic behavior (cf. [8]). However, the more rigid structure of Boolean algebras restricts their computational behavior in many ways; some examples of this were given above. Little is known in effective algebra about classes of lattices between general (non-distributive) lattices and Boolean algebras. How does slightly weakening the structure of a Boolean algebra or slightly strengthening the structure of a non-distributive lattice change the computability results? To address this, we will consider distributive lattices, which lie between general lattices and Boolean algebras.

1.2 Distributive lattices

First, we will give a brief overview of the classical theory of distributive lattices (without computability theory). A more thorough background can be found in [1] and [7].

Definition 1.2.1. A *lattice* L is a partial order in which every pair of elements has a unique join (least upper bound), denoted $\vee$, and a unique meet (greatest lower bound), denoted $\wedge$. The greatest element (if one exists) is denoted 1, and the least element (if one exists) is denoted 0.

Definition 1.2.2. A *distributive lattice* is a lattice in which the meet and join operations are distributive, that is, for every x, y, z,

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z).$$

One example of a non-distributive lattice is the diamond lattice, M_3, shown in Figure 1.1. Notice that M_3 is not distributive because

$$x \wedge (y \vee z) = x \wedge 1 = x \neq 0 = 0 \vee 0 = (x \wedge y) \vee (x \wedge z).$$

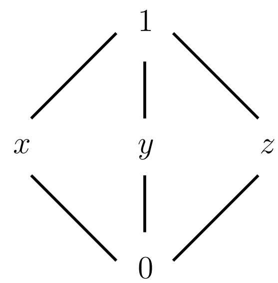


Fig. 1.1: M_3 lattice

Another example is the pentagonal lattice, N_5 (see Figure 1.2). This lattice is also not distributive since

$$y \vee (x \wedge z) = y \vee 0 = y \neq z = 1 \wedge z = (y \vee x) \wedge (y \vee z).$$

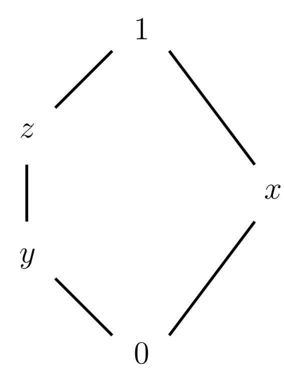


Fig. 1.2: N_5 lattice

In fact, the following theorem gives a way to check if a lattice is distributive using the above examples.

Theorem 1.2.3 (Birkhoff [2]). *A lattice is distributive if and only if neither M_3 nor N_5 embeds into it.*

One method of building a distributive lattice (with least and greatest elements) is by "stacking" smaller distributive lattices on top of one another. For two distributive lattices, we define their sum in the following way.

Definition 1.2.4. If L_1 and L_2 are distributive lattices, the *sum of* L_1 *and* L_2, denoted $L_1 \oplus L_2$, is the distributive lattice with domain $L = L_1 \cup L_2$, $0_L = 0 \in L_1$ (if L_1 has a least element), $1_L = 1 \in L_2$ (if L_2 has a greatest element), and for every other $x, y \in L$,

$$x \leq_L y \Leftrightarrow x, y \in L_i \text{ and } x \leq_{L_i} y \text{ or } x \in L_1 \text{ and } y \in L_2.$$

Under this definition of $\leq_L$ on $L_1 \oplus L_2$, we see that the meet and join operations are as follows for any $x, y \in L$.

$$x \wedge y = \begin{cases} x \wedge_{L_i} y & \text{if } x, y \in L_i \text{ for } i \in \{1, 2\} \\ x & \text{if } x \in L_1 \text{ and } y \in L_2 \\ y & \text{if } x \in L_2 \text{ and } y \in L_1 \end{cases}$$

$$x \vee y = \begin{cases} x \vee_{L_i} y & \text{if } x, y \in L_i \text{ for } i \in \{1, 2\} \\ y & \text{if } x \in L_1 \text{ and } y \in L_2 \\ x & \text{if } x \in L_2 \text{ and } y \in L_1 \end{cases}$$

Also, it is easy to verify that $\leq_L$ is distributive. Notice that the sum operation is associative; i.e., $(L_1 \oplus L_2) \oplus L_3$ and $L_1 \oplus (L_2 \oplus L_3)$ are isomorphic.

A simple example is shown in Figure 1.3 for a four element distributive lattice $D = \{x, y, 0, 1\}$ with $0 < x, y < 1$.

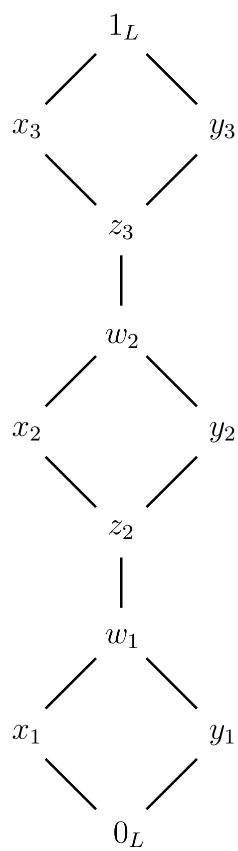


Fig. 1.3: Example of $L = D \oplus D \oplus D$

Note that Definition 1.2.4 can be expanded to $L = L_0 \oplus L_1 \oplus L_2 \oplus \ldots$ for any sequence $\{L_i\}$ of distributive lattices. If L_1 has no least element, we may add a new element 0_L and define $0_L < x$ for every $x \in L$ (if a least element is desired). Similarly, we may add a new element 1_L so that $1_L > x$ for every $x \in L$ (if a greatest element is desired).

Another example of a distributive lattice (with least and greatest elements) is the lattice L generated by open intervals in $\mathbb{Q}$. Denote this lattice by D_η. The basic elements of L have the form (a, b), $(-\infty, a)$, and (b, ∞). The ordering on L is set containment; i.e., $x \leq y$ if $x \subseteq y$. The join of two elements is the usual set union, and the meet is usual set intersection. Then all other elements of L look like finite unions and intersections of these open intervals. The greatest element is $(-\infty, \infty)$, and the least element is $\emptyset$. The fact that this lattice is distributive follows from De Morgan's laws.

Next we will consider several different types of distributive lattices. The most common is a Boolean algebra, a distributive lattice in which every element has a complement. Other types of distributive lattices can be formed by weakening the notion of a complement in some sense. The first example is a pseudocomplement, yielding a pseudocomplemented lattice.

Definition 1.2.5. The *pseudocomplement* of a lattice element x, denoted x^*, is the greatest element y such that $x \wedge y = 0$.

Definition 1.2.6. A *pseudocomplemented lattice* is a distributive lattice with least element such that every element has a (unique) pseudocomplement.

Note that a pseudocomplemented lattice must also have a greatest element, 0^* (by definition of the pseudocomplement).

A Boolean algebra B is one example of a pseudocomplemented lattice, where x^* is the complement of x for every $x \in B$.

A more interesting example is the distributive lattice D_η defined above with the addition of the pseudocomplement. For each $x \in L$, define $x^* = \mathrm{Int}(\bar{x})$, where $\bar{x}$ is the set complement of x and $\mathrm{Int}(y)$ denotes the interior of y. Note that $x \wedge x^* = 0$ since $x \cap \mathrm{Int}(\bar{x}) \subseteq x \cap \bar{x} = \emptyset$, and in fact $\mathrm{Int}(\bar{x})$ is the largest open set that is disjoint from x. To distinguish this lattice from D_η, call this pseudocomplemented lattice P_η. Furthermore, this lattice does not form a Boolean algebra. Consider $(a, b) \in L$. The set $(-\infty, a) \cup (b, \infty)$ is the largest set which is disjoint from (a, b), but $(a, b) \cup (-\infty, a) \cup (b, \infty) \neq (-\infty, \infty)$ because the endpoints a and b are missing. In order to include these endpoints, $(-\infty, a) \cup (b, \infty)$ must be extended to a larger set, but then it would no longer be disjoint from (a, b). Hence (a, b) has no complement in L.

In the case that P_2 and P_2 are pseudocomplemented lattices, we have that $P_1 \oplus P_2$ is also a pseudocomplemented lattice as in Definition 1.2.4. Denote the least element of $P_1 \oplus P_2$ by 0_P and the greatest element by 1_P. (Since P_1 and P_2 are pseudocomplemented lattices, they both have least and greatest elements.)

For any $x \in P_1 \oplus P_2$, we have that x^* is as follows.

$$x^* = \begin{cases} x^{*_{P_1}} & \text{if } x \in P_1 \text{ and } x \neq 0_P \\ 1_P & \text{if } x = 0_P \\ 0_P & \text{if } x \in P_2 \end{cases}$$

In fact, only P_1 must be a pseudocomplemented lattice, and P_2 may be any distributive lattice. To extend this definition to $P = P_1 \oplus P_2 \oplus \ldots$ for any infinite sequence $\{P_i\}$, add a new element 1_P such that $1_P > x$ for every $x \in P$, and let $0_P, \leq_P, \wedge_P, \vee_P$, and $*_P$ be as above.

Another notion of a weakened complement is the relative pseudocomplement of two lattice elements. This gives rise to a special type of distributive lattice called a Heyting algebra.

Definition 1.2.7. The *relative pseudocomplement* of two lattice elements, denoted $x \to y$, is the greatest element z such that $x \wedge z \leq y$.

Definition 1.2.8. A *Heyting algebra* is a distributive lattice with least element such that a (unique) relative pseudocomplement exists for every pair of elements.

Remark 1.2.9. *Every Heyting algebra H is also a pseudocomplemented lattice where $x^* = x \to 0$ for every $x \in H$. Also, a Heyting algebra always has a greatest element given by $0 \to 0$.*

Heyting algebras arise as models of intuitionistic logic, a modification of classical logic in which the law of the excluded middle does not always hold. We will exploit this connection between intuitionistic logic and Heyting algebras in section 4.2.

Notice that any Boolean algebra is automatically a Heyting algebra with $x \to y = \bar{x} \vee y$.

Below is a simple example of a Heyting algebra which is not a Boolean algebra (see Figure 1.4). In this lattice, we have $x \to y = y$, $y \to x = x$, $u \to v = 1$ for every $u \leq v$, and $u \to v = u$ for every $u \geq v$. It is not a Boolean algebra because x and y do not have complements.

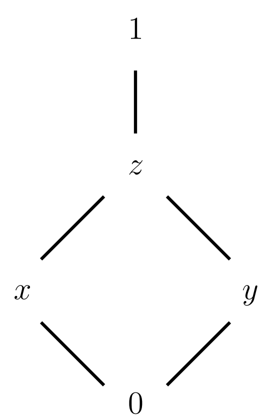


Fig. 1.4: An example of a Heyting algebra

In fact, any finite distributive lattice is a Heyting algebra. To see that $x \to y$ exists for any pair of elements, consider the set $\{z \mid x \wedge z \leq y\}$. This set is nonempty because it always contains 0. Also, if there are two incomparable elements z_1 and z_2 such that $x \wedge z_1 \leq y$ and $x \wedge z_2 \leq y$, then since $x \wedge (z_1 \vee z_2) =$

$(x \wedge z_1) \vee (x \wedge z_2) \leq y$, $z_1 \vee z_2$ is also in the set, and $z_1 \vee z_2$ is strictly above both z_1 and z_2. Therefore, since the lattice is finite, this set has a maximum element, which is the definition of $x \to y$.

The interval algebra D_η can be extended to a Heyting algebra where $\leq, \wedge$, and $\vee$ are as before, and $x \to y = \mathrm{Int}(\bar{x} \cup y)$. Denote this Heyting algebra by H_η.

For any pair of Heyting algebras H_1 and H_2, the sum of H_1 and H_2 is also a Heyting algebra as in Definition 1.2.4, where $H = H_1 \cup H_2$, $0_H = 0_{H_1}$, $1_H = 1_{H_2}$. For any $x, y \in H$, $x \to y$ is as follows.

$$x \to y = \begin{cases} x \to_{H_i} y & \text{if } x, y \in H_i \text{ for } i \in \{1,2\} \text{ and } x \to_{H_i} y \neq 1_{H_1} \\ 1_H & \text{if } x, y \in H_1 \text{ and } x \to_{H_1} y = 1_{H_1} \\ 1_H & \text{if } x \in H_1 \text{ and } y \in H_2 \\ y & \text{if } x \in H_2 \text{ and } y \in H_1 \end{cases}$$

As above, this definition can be expanded to $H_1 \oplus H_2 \oplus \ldots$ for any sequence $\{H_i\}$ of Heyting algebras by adding a new element 1_H and defining it to be above every other element in H.

Finally, we describe a way to build a distributive lattice or Heyting algebra out of (infinitely many) copies of a fixed finite distributive lattice or Heyting algebra. Let L be a linear order and F be any finite distributive lattice or Heyting algebra. We will denote the order on L by $\leq_L$ and the order on F by $\leq_F$. Also denote the greatest element of F by 1_F, the least element of F by 0_F, and meet and join in F by $\wedge_F$ and $\vee_F$, respectively.

Define a new distributive lattice or Heyting algebra $L(F)$ as follows. Replace each point $x \in L$ by a copy of F. Denote this copy of F by F_x. Add two additional elements $1_{L(F)}$ and $0_{L(F)}$. Define the order $\leq_{L(F)}$ on $L(F)$ by making $0_{L(F)}$ the least element, $1_{L(F)}$ the greatest element, and for all other elements of $L(F)$, define $a \leq_{L(F)} b$ if and only if $a \in F_x, b \in F_y$ and $x <_L y$ or $a, b \in F_x$ and $a \leq_F b$.

Lemma 1.2.10. *If F is a distributive lattice, then so is $L(F)$.*

Proof. $L(F)$ is a lattice with the order described above and meet and join defined in the following way: for $a, b \in L(F)$,

$$a \wedge_{L(F)} b = \begin{cases} a \wedge_F b & \text{if } a, b \in F_x \\ a & \text{if } a \in F_x, b \in F_y, \text{ and } x <_L y \end{cases}$$

$$a \vee_{L(F)} b = \begin{cases} a \vee_F b & \text{if } a, b \in F_x \\ b & \text{if } a \in F_x, b \in F_y, \text{ and } x <_L y \end{cases}$$

Also, we have

$$a \wedge_{L(F)} 0_{L(F)} = 0,$$

$$a \vee_{L(F)} 0_{L(F)} = a,$$

$$a \wedge_{L(F)} 1_{L(F)} = a,$$

$$a \vee_{L(F)} 1_{L(F)} = 1_{L(F)}.$$

To see that $L(F)$ is distributive, let $a, b, c \in L(F)$. (To simplify the notation, we will write $\wedge$ for $\wedge_{L(F)}$ and $\vee$ for $\vee_{L(F)}$.) If $a, b, c \in F_x$, then the identities follow from F being distributive.

If $a \in F_x$, $b, c \in F_y$, and $x <_L y$, then

$$a \wedge (b \vee c) = a = a \vee a = (a \wedge b) \vee (a \wedge c), \text{ and}$$

$$a \vee (b \wedge c) = b \wedge c = (a \vee b) \wedge (a \vee c).$$

If $a, b \in F_x, c \in F_y$, and $x <_L y$, then

$$a \wedge (b \vee c) = a \wedge c = a = (a \wedge b) \vee a = (a \wedge b) \vee (a \wedge c), \text{ and}$$

$$a \vee (b \wedge c) = a \vee b = (a \vee b) \wedge c = (a \vee b) \wedge (a \vee c).$$

If $a, c \in F_x, b \in F_y$, and $x <_L y$, then

$$a \wedge (b \vee c) = a \wedge b = a = a \vee (a \wedge c) = (a \wedge b) \vee (a \wedge c), \text{ and}$$

$$a \vee (b \wedge c) = a \vee c = b \wedge (a \vee c) = (a \vee b) \wedge (a \vee c).$$

The other cases follow by symmetry. □

Lemma 1.2.11. *If F is a Heyting algebra, then so is $L(F)$.*

Proof. From above we have that $L(F)$ is a distributive lattice with least and greatest elements. The relative pseudocomplement on $L(F)$ is defined by:

$$a \to_{L(F)} b = \begin{cases} a \to_F b & \text{if } a, b \in F_x \text{ and } a \to_F b \neq 1_F \\ 1_{L(F)} & \text{if } a, b \in F_x \text{ and } a \to_F b = 1_F \\ 1_{L(F)} & \text{if } a \in F_x, b \in F_y, \text{ and } x <_L y \\ b & \text{if } a \in F_x, b \in F_y, \text{ and } y <_L x \end{cases}$$

Also, for each $a \neq 0_L$ we have

$$a \to_{L(F)} 0_{L(F)} = 0_{L(F)},$$

$$a \to_{L(F)} 1_{L(F)} = 1_{L(F)},$$

$$0_{L(F)} \to_{L(F)} a = 1_{L(F)},$$

$$1_{L(F)} \to_{L(F)} a = a.$$

□

Remark 1.2.12. *If $H_i = H$ for each $i \in \omega$ and some fixed finite Heyting algebra H, then $H_1 \oplus H_2 \oplus \ldots$ is the same as $L(H)$ for $L = \omega$.*

1.3 Computable distributive lattices

The definitions for computable distributive lattices, pseudocomplemented lattices, and Heyting algebras follow from Definition 1.1.4 and are given more precisely below.

Definition 1.3.1. A *distributive lattice* $(L, \leq, \wedge, \vee)$ *is computable* if its domain L is computable and the ordering $\leq$ and operations $\wedge$ and $\vee$ are all computable.

Definition 1.3.2. A *pseudocomplemented lattice* $(L, \leq, \wedge, \vee, *)$ *is computable* if its domain L is computable, the ordering $\leq$ is computable, and the operations $\wedge$, $\vee$, and $*$ are all computable.

Definition 1.3.3. A *Heyting algebra* $(L, \leq, \wedge, \vee, \rightarrow)$ *is computable* if its domain L is computable, the ordering $\leq$ is computable, and the operations $\wedge$, $\vee$, and $\rightarrow$ are all computable.

In the above definitions, we are careful to include all of the lattice operations in our language. For instance, the relative pseudocomplement of any two elements in a computable Heyting algebra is computable by definition because $\rightarrow$ is included in the language. One may ask if this is actually necessary. For example, the complement of an element is computable in a computable lattice $(B, \leq, \wedge, \vee)$ which is classically a Boolean algebra (where complementation is not included in the language). This is because, given $x \in B$, we can search for the unique $y \in B$ such that $x \wedge y = 0$ and $x \vee y = 1$. (We may assume that we know 0 and 1, since this is finitely much information.) The element y exists, so we will eventually find it and say that $\bar{x} = y$. Would this also work for Heyting algebras? In other words, given a computable distributive lattice H which is classically a Heyting algebra and a pair $x, y \in H$, is finding $x \rightarrow y$ computable?

Theorem 1.3.4 shows that in general the answer to this question is "no."

Theorem 1.3.4. *There is a computable distributive lattice* $(L, \leq, \wedge, \vee)$ *which is classically a Heyting algebra, but for which the relative pseudocomplementation* *function* $\rightarrow$ *is not computable. In fact, in any computable presentation of* L *as a distributive lattice, the relative pseudocomplementation function has Turing degree* $\mathbf{0}'$.

Proof. We will build the lattice in stages.

Stage 0: Let $L_0 = \{0, 1, a_0, b_0, c_0, x_0, z_0\}$. Define $\leq$, $\wedge$, and $\vee$ for every pair of elements as in Figure 1.5. Notice that L_0 is classically a (finite) Heyting algebra with $c_0 \rightarrow x_0 = x_0$.

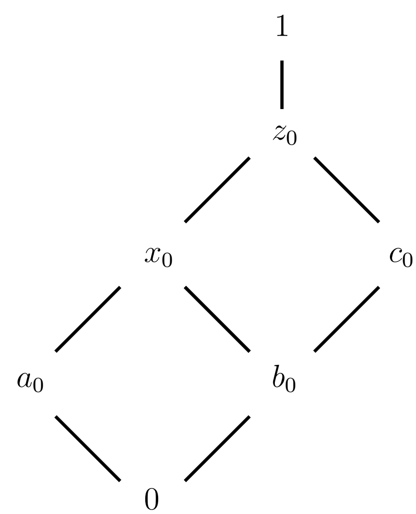


Fig. 1.5: L_0

Stage $s+1$: Given L_s, extend it to L_{s+1} by adding elements a_{s+1}, b_{s+1}, c_{s+1}, x_{s+1}, z_{x+1} and defining $\leq$, $\wedge$, and $\vee$ for every pair of elements as in Figure 1.6. The lattice L_{s+1} is also classically a (finite) Heyting algebra.

For each $e \leq s+1$, check if $e \in K_s \backslash K_{s-1}$. (Assume without loss of generality that at most one number enters K_t at each stage $t \in \omega$.) If so, add a new element y_e between x_e and a_{e+1} (see Figure 1.7). Now $c_e \rightarrow x_e = y_e$ instead of x_e.

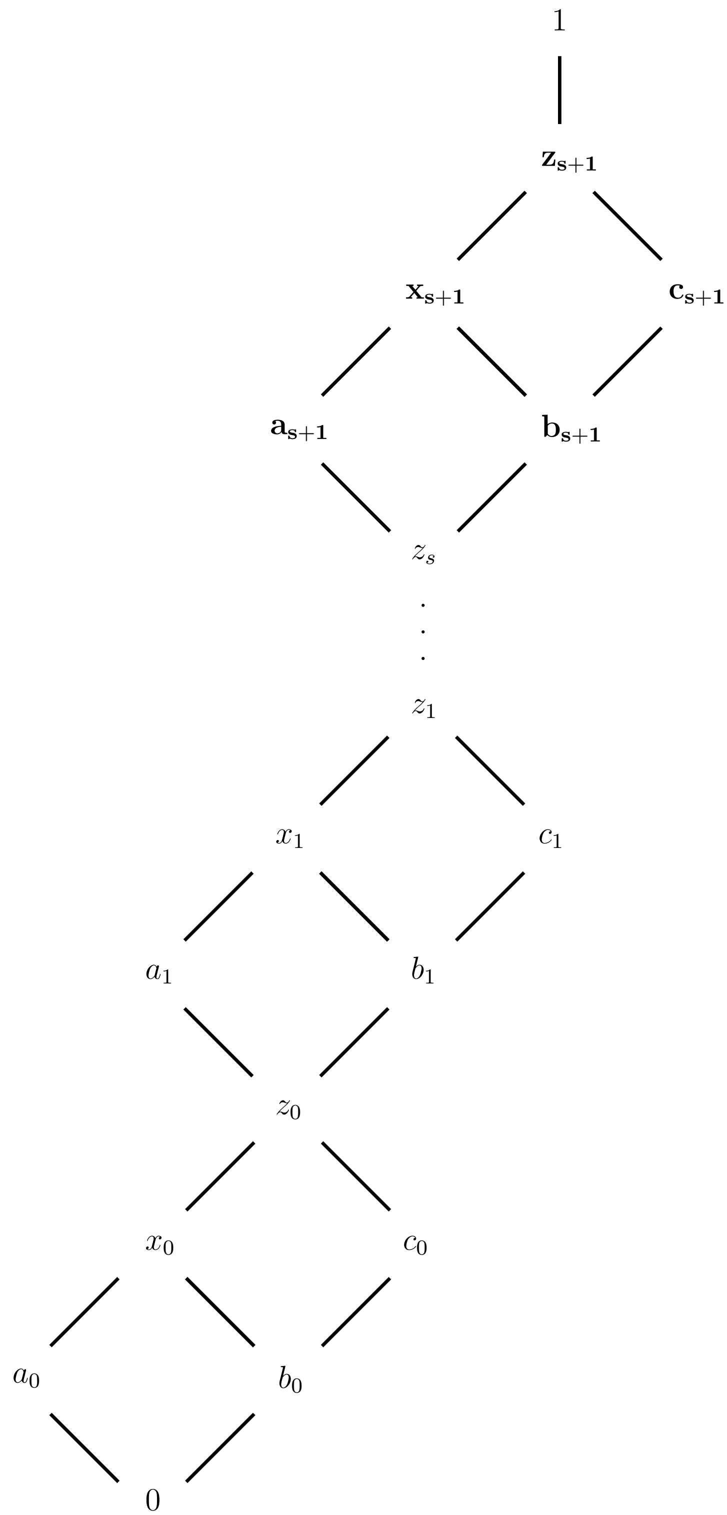

1
$\mathbf{z_{s+1}}$
$\mathbf{x_{s+1}}$
$\mathbf{c_{s+1}}$
$\mathbf{a_{s+1}}$
$\mathbf{b_{s+1}}$
z_s
z_1
x_1
c_1
a_1
b_1
z_0
x_0
c_0
a_0
b_0
0

Fig. 1.6: L_{s+1}

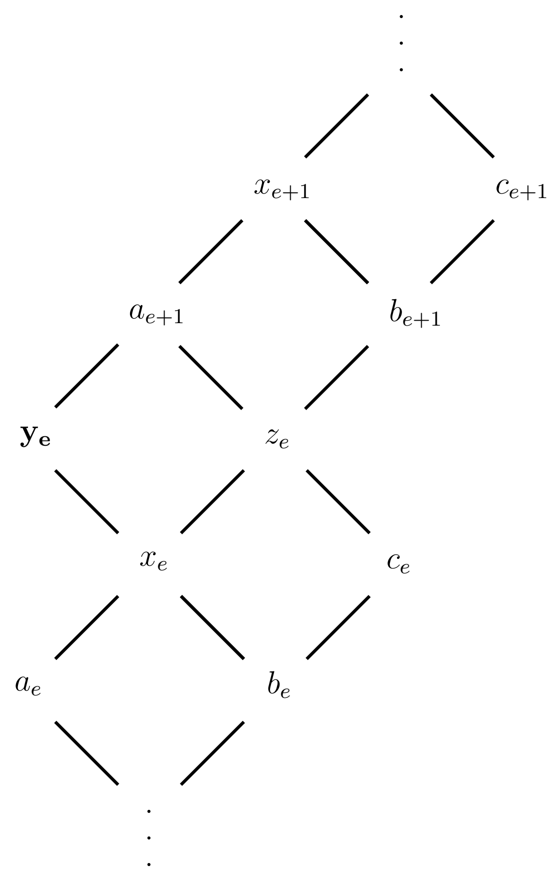


Fig. 1.7: Adding y_e to L_{s+1}

In the end, let $L = \cup_s L_s$. L is an infinite Heyting algebra, as it is a distributive lattice in which $x \to y$ exists for each pair of elements. It is computable as a distributive lattice with the language $(L, \leq, \wedge, \vee)$ since, for any $x, y \in L$, we can run through the construction for finitely many stages until x and y both appear to determine $x \wedge y$, $x \vee y$, and how x relates to y with respect to the ordering $\leq$. The important point is that once $\leq$, $\wedge$, and $\vee$ are defined for any two elements at stage s in the construction, these definitions never change at any future stage $t > s$.

However, the $\to$ operation does change; $c_e \to x_e = x_e$ if and only if $e \notin K$.

Hence, if $x \to y$ were computable for any $x, y \in L$, then we could compute K. Thus, it is not computable in this lattice.

Furthermore, $x \to y$ is not computable in any computable copy of L. This is because L was constructed in such a way that the special elements a_i, b_i, c_i, x_i and z_i for all $i \in \omega$ can be found computably using the following algorithm. Assume that we have a computable copy of L in which we know 0 computably (as this is finitely much information). Search for $u, v, w \in L$ such that all of these relations hold:

$$u \wedge v = 0, u \wedge w = 0, \text{ and } 0 < v < w.$$

Such elements exist, and once found, it must be that $u = a_0$, $v = b_0$, and $w = c_0$. Then $x_0 = a_0 \vee b_0$, and $z_0 = c_0 \vee x_0$. Suppose we know a_k, b_k, c_k, x_k, and z_k. Then we can find $a_{k+1}, b_{k+1}, c_{k+1}, x_{k+1}$, and z_{k+1} computably by again searching for $u, v, w \in L$ such that all of the following hold:

$$u \wedge v = z_k, u \wedge w = z_k, \text{ and } z_k < v < w.$$

It must be that $u = a_{k+1}$, $v = b_{k+1}$, and $w = c_{k+1}$.

Now, if $x \to y$ were computable for every $x, y \in L$, then we would arrive at the same contradiction because $c_e \to x_e = x_e$ if and only if $e \notin K$, and for any e, the elements c_e and x_e can be found in finitely many steps (iterating the method above e times). □

This means that any computable Heyting algebra must include $\to$ in its language; it is not the case that $\rightarrow$ could be derived computably from $\leq$, $\wedge$, and $\vee$ being computable.

One may also wonder if the relative pseudocomplementation operator could be computed from the pseudocomplementation operator. In other words, if we take the language of a computable Heyting algebra to be $(L, \leq, \wedge, \vee, *)$, is $x \rightarrow y$ computable for every $x, y \in L$? Again, the answer is "no." This is immediate from Theorem 1.3.4.

Corollary 1.3.5. *There is a computable pseudocomplemented lattice which is classically a Heyting algebra but for which the relative pseudocomplementation function is not computable. In fact, in any computable presentation of L as a pseudocomplemented lattice, the relative pseudocomplementation function has Turing degree* $\mathbf{0}'$.

Proof. Build the computable lattice L as in Theorem 1.3.4 with one minor change. At each stage $s > 0$, if $e \in K_s \setminus K_{s-1}$, add a new element y_{e+1} between x_{e+1} and a_{e+2}. Then the sublattice containing $0, a_0, b_0, c_0, x_0$, and z_0 remains unchanged, so we can compute the pseudocomplement for each element in L as follows.

$$a_0^* = c_0, b_0^* = a_0, c_0^* = a_0$$

$$y^* = 0 \text{ for every other } y \in L$$

It is easy to see that this holds for the lattice L_0. The key is that z_0 is comparable to every other element in L. Therefore, for every $y, w \notin L_0$, $y, w \geq z_0$, so $y \wedge w \geq z_0$. If $y \notin L_0$ and $w \neq 0 \in L_0$, then $y > z_0 > w$, so $y \wedge w = w$. Hence, for each $y \neq a_0, b_0$ or c_0, the only element w such that $y \wedge w = 0$ is $w = 0$.

Then we have $c_{e+1} \rightarrow x_{e+1} = x_{e+1}$ if and only if $e \notin K$ as before, so the relative pseudocomplementation operator is not computable in (any computable copy of) L. □

It will be useful to have a collection of examples of computable distributive lattices. The following lemmas will show that most of the distributive lattices, pseudocomplemented lattices, and Heyting algebras from section 1.2 can be constructed computably. Both results follow from the explicit definitions of $\wedge, \vee, \leq$ $(*, \rightarrow)$ given in the definitions of $L_0 \oplus L_1 \oplus L_2 \oplus \ldots$ and $L(F)$.

Lemma 1.3.6. *If $\{L_i\}_{i \in \omega}$ is a uniform sequence of computable distributive lattice (pseudocomplemented lattice, Heyting algebra), then $L = L_0 \oplus L_1 \oplus L_2 \oplus \ldots$ is a computable distributive lattice (pseudocomplemented lattice, Heyting algebra).*

Lemma 1.3.7. *If L is a computable linear order and F is a finite distributive lattice or Heyting algebra, then $L(F)$ is a computable distributive lattice or Heyting algebra, respectively.*

1.4 Summary of results

Chapter 2 will deal with prime ideals (or filters) in a computable distributive lattice. It follows from the definition that the class of prime ideals in a lattice forms a Π^0_1 class. We will show that the minimal prime ideals of a pseudocomplemented lattice (or Heyting algebra) also form a Π^0_1 class. Furthermore, we will show that the converse does not hold; it is not true that any Π^0_1 class can be represented by the minimal prime ideals of computable pseudocomplemented lattice or the prime ideals of a computable distributive lattice. This will follow after proving that it is always possible to find a (nontrivial) computable minimal prime ideal in a computable pseudocomplemented lattice and a (nontrivial) computable prime ideal in a computable distributive lattice. (In fact, if a computable distributive lattice has a least or greatest element, we will see that is always possible to find a computable minimal or maximal prime ideal, respectively.) It is known that the set of maximal filters in a computable Boolean algebra forms a Π^0_1 class, and every computable Boolean algebra has a computable maximal ideal (cf. [3]). These facts rely on the existence of the complement for each element in a Boolean algebra. Therefore, we will have shown that the existence of the slightly weaker pseudocomplement suffices to prove the above analogous theorems for pseudocomplemented lattices and Heyting algebras. Moreover, finding a (nontrivial) computable prime ideal in a computable distributive lattice will not require any notion of a complement.

At the end of chapter 2, we will see that, by contrast, it is not always possible to find a computable prime ideal in a computable general (non-distributive) lattice. We will show this by constructing a computable (non-distributive) lattice such that its (nontrivial) prime ideals code an arbitrary Π^0_1 class.

In chapter 3, we will consider the degree spectra of lattices. For general (non-distributive) lattices, the degree spectra is known to be $\{\mathbf{c} \mid \mathbf{c} \geq \mathbf{d}\}$ for any Turing degree $\mathbf{d}$ (cf. [8,16]). We will extend a result of Selivanov [17] to show that this is also true for distributive lattices, pseudocomplemented lattices, and Heyting algebras.

Finally, in chapter 4, we will investigate the possible values of the computable dimension for Heyting algebras and work toward finding a characterization for the computably categorical Heyting algebras. We will show that the computable dimension of the free Heyting algebras is 1 or ω, depending on whether there are finitely many or infinitely many generators, respectively. To show that the computable dimension of the free Heyting algebra on infinitely many generators is ω, we will use the fact that it is a model of intuitionistic propositional logic over infinitely many propositional variables. In general, we can say that two Heyting algebras which are computably categorical as distributive lattices must also be computably categorical as Heyting algebras. However, we will show that the converse does not hold; there are two Heyting algebras which are computably categorical as Heyting algebras but not as distributive lattices. The idea is that having the relative pseudocomplement in the language of computable Heyting algebras will give us more information that can be used to construct a computable isomorphism. Lastly, we will make some final remarks on why certain algebraic properties of Heyting algebras (the set of atoms, the set of join-irreducible elements, the subalgebra which forms a Boolean algebra) are not good candidates for finding a characterization of the computably categorical Heyting algebras in general.

Chapter 2

Computable prime ideals

2.1 The class of prime ideals

Prime ideals and prime filters are natural substructures of lattices to consider from the viewpoint of effective algebra because there are many classical theorems on ideals and filters in the literature to take advantage of. Also, since prime filters have been studied in Boolean algebras (where every prime filter is a maximal filter and vice versa), the analogous results on distributive lattices can be compared to those known for Boolean algebras. Prime ideals and filters of lattices are defined as follows.

Definition 2.1.1. An *ideal* in a lattice L is a set $I \subseteq L$ such that for all $x, y \in L$, I satisfies all of the following:

$$(x \leq y \text{ and } y \in I) \Rightarrow x \in I,$$

$$x, y \in I \Rightarrow x \vee y \in I.$$

Definition 2.1.2. A *prime ideal* in a lattice L is an ideal $P \subseteq L$ such that for all $x, y \in L$,

$$x \wedge y \in P \Rightarrow (x \in P \text{ or } y \in P).$$

Definition 2.1.3. A *filter* in a lattice L is a set $F \subseteq L$ such that for all $x, y \in L$, F satisfies all of the following:

$$(x \geq y \text{ and } y \in F) \Rightarrow x \in F,$$

$$x, y \in F \Rightarrow x \wedge y \in F.$$

Definition 2.1.4. A *prime filter* in a lattice L is a filter $F \subseteq L$ such that for all $x, y \in L$,

$$x \vee y \in F \Rightarrow (x \in F \text{ or } y \in F).$$

Remark 2.1.5. *The empty set and the entire lattice are always trivially prime ideals. If a lattice has a least element* 0 *and greatest element* 1*, then a prime ideal* P *is nontrivial if and only if* $0 \in P$ *and* $1 \notin P$*. Similarly, a prime filter* P *is nontrivial if and only if* $0 \notin P$ *and* $1 \in P$*.*

We will mostly be concerned with prime ideals rather than prime filters, but the results on ideals will transfer to filters in the following way.

Lemma 2.1.6. *If P is a nontrivial prime ideal of L, then $L \setminus P$ is a nontrivial prime filter.*

Proof. Suppose $x \geq y$ and $y \in L \setminus P$. If $x \in P$, then since P is an ideal, $y \in P$, contradicting that $y \notin P$. Therefore, $x \in L \setminus P$.

Now suppose that $x, y \in L \setminus P$. If $x \wedge y \in P$, then because P is prime, either $x \in P$ or $y \in P$, contradicting that $x, y \notin P$. Hence $x \wedge y \in L \setminus P$.

Finally, if $x \vee y \in L \setminus P$ but $x, y \in P$, then $x \vee y \in P$ since P is closed under join. Again we arrive at a contradiction, so $x \in L \setminus P$ or $y \in L \setminus P$. □

From this point on, "prime ideal" will always mean "nontrivial prime ideal," unless otherwise specified.

We are interested in studying the complexity of prime ideals in a computable lattice. Recall the discussion of Π^0_1 classes in section 1.1. By Definition 1.1.2, we have the following.

Theorem 2.1.7. *The collection of prime ideals in a computable lattice with 0 and 1 is Π^0_1 class.*

Proof. This follows from Definition 2.1.2 since it uses only universally quantified statements. It is computable to tell if a given prime ideal is nontrivial by Remark 2.1.5 (using the fact that 0 and 1 are in the lattice). □

As a lattice may contain multiple prime ideals, the concept of a maximal or minimal prime ideal is defined as follows.

Definition 2.1.8. A minimal prime ideal P in a lattice L is a prime ideal such that, for every other prime ideal Q of L, $Q \not\subset P$.

Definition 2.1.9. A maximal prime ideal P in a lattice L is a prime ideal such that, for every other prime ideal Q of L, $P \not\subset Q$.

At first, the class of minimal prime ideals of a lattice L seems more complex than the class of prime ideals because Definition 2.1.8 quantifies over subsets of L. However, we will see that this is not the case for pseudocomplemented lattices and Heyting algebras. The following lemma gives a useful characterization of the minimal prime ideals in a pseudocomplemented lattice.

Lemma 2.1.10 (Grätzer [7])**.** *Let L be a pseudocomplemented lattice and P be a prime ideal of L. The following are equivalent:*

(i) *P is a minimal prime ideal.*

(ii) *If $x \in P$ then $x^* \notin P$.*

(iii) *If $x \in P$ then $x^{**} \in P$.*

Theorem 2.1.11. *The class of minimal prime ideals of a pseudocomplemented lattice is a Π^0_1 class.*

Proof. From Theorem 2.1.7, the class of prime ideals of a pseudocomplemented lattice forms Π^0_1 class, and part (ii) (or part (iii)) of Lemma 2.1.10 is a universally quantified statement for deciding whether a prime ideal is minimal. □

Theorem 2.1.11 can also be proven by appealing to the characterization of Π^0_1 classes as infinite paths through computable subtrees by Definition 1.1.3. The set of infinite paths through a tree T is denoted by $[T]$. This first requires one technical lemma about the pseudocomplementation function.

Lemma 2.1.12 (Grätzer [7]). *Let L be a pseudocomplemented lattice. For $x, y \in L$, the following identities hold.*

(i) $(x \vee y)^* = x^* \wedge y^*$

(ii) $x^{**} \wedge y^{**} = (x \wedge y)^{**}$

Theorem 2.1.13. *If L is a computable pseudocomplemented lattice, there is a computable tree T_L and map F such that*

$$F \ : \ [T_L] \to \ \textit{minimal prime ideals of } L$$

is a bijection and preserves Turing degree.

Proof. Let $\{a_0, a_1, a_2, \dots\}$ be an effective enumeration of the domain of L. Define T_L by induction on the length of σ. For $|\sigma| = k > 0$, σ represents the guess that $\{a_0^{\sigma(0)}, a_1^{\sigma(1)}, \dots, a_{k-1}^{\sigma(k-1)}\}$ can be extended to a minimal prime ideal, with

$$a_i^{\sigma(i)} = \begin{cases} a_i & \text{if } \sigma(i) = 1 \\ a_i^* & \text{if } \sigma(i) = 0 \end{cases}$$

Begin with $T_L = \emptyset$. Suppose by induction that $\sigma \in T_L$ and that we have defined $I_\sigma \supseteq \{a_0^{\sigma(0)}, a_1^{\sigma(1)}, \dots, a_{k-1}^{\sigma(k-1)}\}$. Define

$$I_{\sigma * 1} := I_\sigma \cup \{a_k\} \cup \{a_j \mid j \leq k \text{ and } \exists x, y \in I_\sigma \cup \{a_k\} \ (a_j \leq x \vee y)\}$$

$$I_{\sigma*0} := I_\sigma \cup \{a_k^*\} \cup \{a_j \mid j \leq k \text{ and } \exists x, y \in I_\sigma \cup \{a_k^*\}\ (a_j \leq x \vee y)\}.$$

Note that because of the bounded quantifiers, these sets are computable.

Now check if there is an a_i such that $i \leq k$ and both $a_i, a_i^* \in I_{\sigma*1}$. If so, do not put $\sigma * 1 \in T_L$. Otherwise, put $\sigma * 1 \in T_L$. Perform a similar check for $I_{\sigma*0}$.

Define F on $[T_L]$ by

$$F(f) = P_f = \{a_k \mid f(k) = 0\} \cup \{a_k^* \mid f(k) = 1\}.$$

We first verify that P_f is an ideal of L. Let $a_j, a_k \in P_f$ and suppose that $a_j \vee a_k \notin P_f$. Then, by construction, $(a_j \vee a_k)^* \in P_f$. Also, $a_j \vee a_k = a_\ell$ and $(a_j \vee a_k)^* = a_n$ for some $\ell, n \in \omega$. Take $m = \max(j, k, \ell, n) + 1$. Then $a_j, a_k, (a_j \vee a_k)^* \in I_{f\restriction m}$. Also, $a_j \vee a_k \in I_{f\restriction m}$ since $\ell \leq m-1$ and $a_j, a_k \in I_{f\restriction m-1} \cup \{a_{m-1}^{f(m-1)}\}$ with $a_\ell \leq (a_j \vee a_k)$. Thus, both $(a_j \vee a_k)$ and $(a_j \vee a_k)^*$ are in $I_{f\restriction m}$, which is a contradiction because f would not be a path, as $f \restriction m$ would not have been extended. Similarly, let $a_k \in P_f$ with $a_j \leq a_k$. Suppose that $a_j \notin P_f$. As above, this implies that $a_j^* \in P_f$. Take $m = \max(j, k) + 1$. Then $a_k, a_j^* \in I_{f\restriction m}$, $j \leq m-1$, and $a_j \leq (a_k \vee a_k)$, so $a_j \in I_{f\restriction m}$ as well. Again, this is a contradiction because $f \restriction m$ would not have been extended.

Next we show that P_f is prime. This relies on the fact that either a_i or a_i^* is in P_f for each $i \in \omega$, but not both $a_i, a_i^* \in P_f$ (or the path would not have been extended). Suppose that $x \wedge y \in P_f$ but $x, y \notin P_f$. Then $x^*, y^* \in P_f$ by construction. Since P_f is an ideal, $x^* \vee y^* \in P_f$. This means that $(x^* \vee y^*)^* \notin P_f$. By Lemma 2.1.12, $(x^* \vee y^*)^* = x^{**} \wedge y^{**} = (x \wedge y)^{**}$. Also, since $x \wedge y \in P_f$, we have $(x \wedge y)^* \notin P_f$, so $(x \wedge y)^{**} \in P_f$. This contradicts that $(x \wedge y)^{**} \notin P_f$, so at least one of x or, y is in P_f.

Furthermore, as noted above, the construction ensures that for each $i \in \omega$, if $a_i \in P_f$, then $a_i^* \notin P_f$. Therefore, by Lemma 2.1.10, P_f is a minimal prime ideal.

Any path f can be computed from P_f since $f(i) = 0$ if $a_i \in P_f$ and $f(i) = 1$ if $a_i \notin P_f$. Similarly, knowing f, we can say that $a_k \in P_f$ if and only if $f(k) = 0$. Therefore, $P_f \equiv_T f$.

To see that F is injective, let $f_1, f_2 \in [T_L]$ such that $f_1 \neq f_2$. Then f_1 and f_2 differ on some $i \in \omega$; let i be the least such that $f_1(i) \neq f_2(i)$. This means that one of a_i, a_i^* is in P_{f_1} and the other is in P_{f_2}. We cannot have $P_{f_1} = P_{f_2}$, or we would have that $a_i, a_i^* \in P_{f_1}$, but this contradicts that f_1 is a path in T_L (by construction, $f_1 \upharpoonright i$ would not have been extended). Therefore, $P_{f_1} \neq P_{f_2}$.

Now let P be a minimal prime ideal of L. Again, by Lemma 2.1.10, either $a_i \in P$ or $a_i^* \in P$ for each $i \in \omega$. Let f be such that $f(i) = 1$ if $a_i \in P$ and $f(i) = 0$ if $a_i^* \in P$. Then $f \in [T_L]$, for if not, then there would be some $\sigma \in T_L$ such that $f \upharpoonright n = \sigma \upharpoonright n$ but σ has no extension in T_L. Then there is some j such that $a_j, a_j^* \in I_{\sigma * f(n)}$. However, $I_{\sigma * f(n)} \subseteq P$, so $a_j, a_j^* \in P$, contradicting that P is a minimal prime ideal of L. Therefore, $f \in [T_L]$ and $F(f) = P$, so F is surjective. □

We obtain several corollaries to Theorem 2.1.13 which follow from known facts about Π^0_1 classes. For instance, in a computable pseudocomplemented lattice, there is always a minimal prime ideal of low degree and one of hyperimmune-free degree (cf. [9]).

On the other hand, it is more interesting to know whether these classes of minimal prime ideals can represent all Π^0_1 classes. That is, given a Π^0_1 class C, is there a computable pseudocomplemented lattice L such that C codes the set of prime ideals in L? In the next section, we will see that this is not always possible.

2.2 Computable minimal prime ideals

We will prove that it is always possible to find a computable minimal prime ideal in a computable pseudocomplemented lattice. Since there are Π^0_1 classes with no computable members (cf. [9]), this will show that the converse of Theorem 2.1.13 does not hold; that is, there is no way to code an arbitrary Π^0_1 class into the minimal prime ideals of a computable pseudocomplemented lattice.

First, we need to define the dense set of a pseudocomplemented lattice.

Definition 2.2.1. Let L be a pseudocomplemented lattice. The *dense set* of L, denoted $D(L)$, is given by

$$D(L) = \{x \mid x^* = 0\}.$$

The following lemma uses the dense set to yield another characterization of the minimal prime ideals in a pseudocomplemented lattice.

Lemma 2.2.2. *Let L be a pseudocomplemented lattice. P is a minimal prime ideal of L if and only if P is a prime ideal of L and $P \cap D(L) = \emptyset$.*

Proof. For the forward direction, let P be a minimal prime ideal of L and suppose that $x \in P \cap D(L)$. Then $x \in P$ and $x^* = 0$. By Lemma 2.1.10, $x \in P$ implies $x^* \notin P$. However, 0 is in every prime ideal. Therefore, $P \cap D(L) = \emptyset$.

For the reverse direction, suppose that P is a prime ideal of L and that $P \cap D(L) = \emptyset$. If $x^* \notin P$ for all $x \in P$, then P is a minimal prime ideal by Lemma 2.1.10. Assume for a contradiction that there is an $x \in P$ such that $x^* \in P$. Then $x \vee x^* \in P$, and by Lemma 2.1.12, $(x \vee x^*)^* = x^* \wedge x^{**} = 0$, so $x \vee x^* \in P \cap D(L)$. However, $P \cap D(L) = \emptyset$, meaning $x^* \notin P$ as desired. □

An important property of the dense set of a pseudocomplemented lattice L is that it is a filter in L. To show this, we first need the following lemma.

Lemma 2.2.3. *Let L be a pseudocomplemented lattice. If $x, y \in L$ and $x \leq y$, then $y^* \leq x^*$.*

Proof. Let $x, y \in L$ and suppose that $x \leq y$. By definition, $y^* \wedge y = 0$. Then we have

$$0 = y^* \wedge y = y^* \wedge (x \vee y) = (y^* \wedge x) \vee (y^* \wedge y) = (y^* \wedge x) \vee 0 = y^* \wedge x.$$

Since $x^* = \max\{z \mid z \wedge x = 0\}$ and $y^* \wedge x = 0$, $y^* \leq x^*$. □

Lemma 2.2.4. *Let L be a pseudocomplemented lattice. The dense set $D(L)$ is a filter in L.*

Proof. Suppose that $x \in D(L)$ and that $x \leq y$. Then $y^* \leq x^* = 0$ by Lemma 2.2.3, so $y \in D(L)$. Hence $D(L)$ is closed upward.

Now suppose that $x, y \in D(L)$. Let $z = (x \wedge y)^*$. Assume for a contradiction that $z > 0$. Since $y^* = 0, z \wedge y > 0$ (otherwise, if $z \wedge y = 0$, then $z \leq y^* = 0$). Similarly, $x \wedge (z \wedge y) > 0$. Therefore, we have

$$0 < x \wedge (z \wedge y) = (x \wedge y) \wedge z = 0.$$

Hence $z = 0$, so $(x \wedge y) \in D(L)$, and $D(L)$ is closed under meet. Thus, $D(L)$ is a filter of L. □

The main tool in showing that it is always possible to construct a computable minimal prime ideal in a pseudocomplemented lattice is the prime ideal theorem (also called the Birkhoff-Stone prime separation theorem).

Theorem 2.2.5 (Prime ideal theorem [1])**.** *Let L be a distributive lattice. If I is an ideal in L and F a filter in L such that $I \cap F = \emptyset$, then there is a prime ideal P in L such that $I \subseteq P$ and $P \cap F = \emptyset$.*

Corollary 2.2.6. *In a distributive lattice with 1, every proper ideal is contained in a maximal ideal.*

We are now ready to prove main theorem of this section. Given a computable pseudocomplemented lattice L, the idea is to construct a minimal prime ideal P in stages by including one of x or x^* in P for each $x \in L$ so that $P \cap D(L) = \emptyset$. Then, after each stage, Theorem 2.2.5 ensures that the (finite) ideal we have built so far will extend to a prime ideal. The fact that P does not intersect $D(L)$ guarantees that it will be minimal by Lemma 2.2.2.

Theorem 2.2.7. *Let L be a computable pseudocomplemented lattice. There exists a computable minimal prime ideal in L.*

Proof. Let $(L, \leq, \wedge, \vee, *)$ be a computable pseudocomplemented lattice with domain $\{a_0 = 0, a_1, a_2, \dots\}$. We will build a computable minimal prime ideal $P \subset L$ in stages; at each stage s we will determine whether $a_s \in P$ or $a_s^* \in P$. Also, if $a_i \in P$, then this will imply that $a_i^* \notin P$ (and vice versa), so P will be computable.

Define

$$a_i^{\varepsilon_i} = \begin{cases} a_i & \text{if } \varepsilon_i = 1 \\ a_i^* & \text{if } \varepsilon_i = 0 \end{cases}$$

Stage 0: Let $P_0 = \{0, 1\}$ with $0 < 1$. Note that $0 \notin D(L)$ since $0^* = 1$, so $P_0 \cap D(L) = \emptyset$.

Stage $s + 1$: Given P_s, we will build P_{s+1}. By induction, assume that $P_s = \{a_0^{\varepsilon_0}, a_1^{\varepsilon_1}, \dots, a_s^{\varepsilon_s}\}$ and that $a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \dots \vee a_s^{\varepsilon_s} \notin D(L)$. If one of $a_{s+1} \in P_s$ or $a_{s+1}^* \in P_s$ already, then set $P_{s+1} = P_s$ and go to the next stage.

Otherwise, let

$$I_s = \{x \mid x \leq (a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \dots \vee a_s^{\varepsilon_s})\}.$$

Suppose that there is an $x \in I_s$ such that $x^* = 0$. Then $x \leq (a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \cdots \vee a_s^{\varepsilon_s})$, and, by Lemma 2.2.3, $(a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \cdots \vee a_s^{\varepsilon_s})^* \leq x^* = 0$. This implies that $(a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \cdots \vee a_s^{\varepsilon_s}) \in D(L)$, contradicting the induction hypothesis. Thus, we have that $I_s \cap D(L) = \emptyset$. By Theorem 2.2.5, there exists a prime ideal $J \supseteq I_s$ with $J \cap D(L) = \emptyset$. Since one of $a_{s+1}, a_{s+1}^* \in J$ and $(a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \cdots \vee a_s^{\varepsilon_s}) \in J$, it must be that one of $(a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \cdots \vee a_s^{\varepsilon_s}) \vee a_{s+1}$ or $(a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \cdots \vee a_s^{\varepsilon_s}) \vee a_{s+1}^*$ is in J and not in $D(L)$. If $(a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \cdots \vee a_s^{\varepsilon_s}) \vee a_{s+1} \notin D(L)$, then let $P_{s+1} = P_s \cup \{a_{s+1}\}$ (so $\varepsilon_{s+1} = 1$). Otherwise let $P_{s+1} = P_s \cup \{a_{s+1}^*\}$ (so $\varepsilon_{s+1} = 0$).

In the end, let $P = \cup_s P_s$.

Observe that this construction ensures that for every $i \in \omega$, one of a_i or a_i^* is in P, and $P \cap D(L) = \emptyset$. To verify that P is an ideal, first let $a_k \in P$ with $a_j \leq a_k$. Suppose that $a_j \notin P$. Then, by construction, $a_j^* \in P$. Let $m = \max(j, k)$. At stage m, $a_j^*, a_k \in P_m$ and $P_m \subseteq I_m = \{x \mid x \leq (a_0^{\varepsilon_0} \vee a_1^{\varepsilon_1} \vee \cdots \vee a_m^{\varepsilon_m})\}$. By Theorem 2.2.5, I_m extends to a prime ideal J such that $J \cap D(L) = \emptyset$. Since $a_k \in J$, $a_j \leq a_k$, we have $a_j \in J$, but $a_j^* \in I_m \subseteq J$, so $a_j, a_j^* \in J$. This contradicts Lemma 2.1.10 (since J is a minimal prime ideal of L by Lemma 2.2.2). Thus $a_j \in P$. Now suppose $a_j, a_k \in P$ but $a_j \vee a_k \notin P$. By construction, $(a_j \vee a_k)^* \in P$. By a similar argument, we reach a stage s where P_s extends to a minimal prime ideal J containing $(a_j \vee a_k)$ and $(a_j \vee a_k)^*$, again contradicting Lemma 2.1.10. Therefore, P is an ideal of L.

Now suppose that $(a_j \wedge a_k) \in P$ but $a_j, a_k \notin P$. Then $a_j^*, a_k^* \in P$. As above, there is a stage s where P_s extends to a prime ideal J containing $(a_j \wedge a_k)$ and a_j^*, a_k^*. Since J is prime, J must contain at least one of a_j, a_k. Then J contains both a_j, a_j^* or a_k, a_k^*, which cannot happen. This shows that P is prime.

Finally, since P is a prime ideal of L and $P \cap D(L) = \emptyset$, P is a minimal prime ideal by Lemma 2.2.2, and it is computable by the above comments. □

Note that Theorems 2.1.13 and 2.2.7 hold for Heyting algebras as well, since Heyting algebras are also pseudocomplemented lattices.

2.3 Computable prime ideals in a distributive lattice

We would like to know if it is always possible to find a (nontrivial) prime ideal of a computable lattice effectively. Since the results for finding computable (minimal) prime ideals in computable pseudocomplemented lattices or (maximal) filters in Boolean algebras rely on the existence of the pseudocomplement and complement, respectively, it seems much harder to effectively find a prime ideal in a distributive lattice where this is not necessarily a notion of a complement (even in a weakened sense). It turns out that no complements or pseudocomplements are needed; it is always possible to find a computable prime ideal in a computable distributive lattice. Theorem 2.2.5 is used in a manner similar to Theorem 2.2.7; at each stage, it ensures that the (finite) ideal we have built so far will extend to a prime ideal.

Theorem 2.3.1. *Let L be a computable distributive lattice. There exists a computable prime ideal in L.*

Proof. Let $(L, \leq, \wedge, \vee)$ be a computable distributive lattice with domain $L = \{a_0, a_1, a_2, \dots\}$. We will construct a computable prime ideal $P \subset L$ in stages, where at each stage s, we determine whether $a_{s+1} \in P$ or $a_{s+1} \notin P$ (with the convention that once an element is left out of P, it is left out forever so that P will be computable in the end).

Stage 0: Consider $a_0 \neq a_1 \in L$. If these elements are comparable, say $a_0 < a_1$, then let $P_0 = \{a_0\}$ and $a_1 \notin P_0$. Otherwise, if a_0 and a_1 are not comparable, it does not matter which element goes in and which stays out, so let $P_0 = \{a_0\}$ and $a_1 \notin P_0$.

Stage s: Suppose that $a_{i_0}, a_{i_1}, \dots, a_{i_k} \in P_{s-1}$ and $a_{j_0}, a_{j_1}, \dots, a_{j_\ell} \notin P_{s-1}$. Also, suppose by induction that

$$(a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell}) \nleq (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k}). \tag{2.3.1}$$

To decide whether to put a_{s+1} into P_s or leave it out, check the following two conditions:

(a) If $(a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell}) \leq (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k} \vee a_{s+1})$, then do not put a_{s+1} into P_s.

(b) If $(a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{s+1}) \leq (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k})$, then put a_{s+1} into P_s.

Otherwise do not put a_{s+1} into P_s.

Note that conditions (a) and (b) cannot both be satisfied in a distributive lattice; if so, then

$$
\begin{aligned}
& a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \\
& = (a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell}) \wedge (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k} \vee a_{s+1}) \qquad \text{by (a)} \\
& = (a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{i_0}) \vee (a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{i_1}) \vee \\
& \qquad \cdots \vee (a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{i_k}) \vee (a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{s+1}) \\
& \leq a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k} \vee (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k}) \qquad \text{by (b)} \\
& = a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k}.
\end{aligned}
$$

This contradicts (2.3.1).

We use Theorem 2.2.5 to show that the construction is valid. Suppose that at stage s, we have $a_{i_0}, a_{i_1}, \ldots, a_{i_k} \in P_{s-1}$ and $a_{j_0}, a_{j_1}, \ldots, a_{j_\ell} \notin P_{s-1}$. If $a_{s+1} \in P_s$, consider

$$I_s = \{y \mid y \leq (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k} \vee a_{s+1})\},$$

and

$$F_s = \{y \mid y \geq (a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell})\}.$$

Notice that I_s is an ideal in L and F_s is a filter in L; I_s is the ideal generated by all of the elements which are in P_s, and F_s is the filter generated by all of the elements which are not in P_s. If $I_s \cap F_s \neq \emptyset$, then there is some $y \in L$ such that

$$(a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell}) \leq y \leq (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k} \vee a_{s+1}).$$

Then condition (a) is satisfied, implying that $a_{s+1} \notin P_s$. However, this contradicts that $a_{s+1} \in P$. Hence $I_s \cap F_s = \emptyset$. By Theorem 2.2.5, there is a prime ideal J in

L such that $I_s \subseteq J$ and $J \cap F_s = \emptyset$; that is, everything in P_s so far extends to a prime ideal which does not contain any of the elements left out of P_s.

Similarly, at stage s, if $a_{s+1} \notin P_s$, let

$$I_s = \{y \mid y \leq (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k})\},$$

and

$$F_s = \{y \mid y \geq (a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{s+1})\}.$$

If $I_s \cap F_s \neq \emptyset$, then $(a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{s+1}) \leq (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k})$. This satisfies condition (b), so $a_{s+1} \in P_s$, which is a contradiction. Therefore $I_s \cap F_s = \emptyset$, and Theorem 2.2.5 applies as above.

In the end, let $P = \cup_s P_s$.

Now we verify that the resulting P is a prime ideal in L. Suppose $a_k \leq a_n$ and $a_n \in P$. If $a_k \notin P$, let s be the least stage where $a_n \in P_s$ but $a_k \notin P_s$. From above, P_s extends to a prime ideal J containing a_n and $a_k \notin J$ (since $a_k \in F_s$ where $F_s \cap J = \emptyset$). This contradicts that J is an ideal. Therefore P is closed downward.

Suppose $a_k, a_m \in P$ but $a_k \vee a_m \notin P$. Let s be the least stage where $a_k, a_m \in P_s$ and $a_k \vee a_m \notin P_s$. As above, we get a contradiction because P_s extends to a prime ideal J where $a_k, a_m \in J$ but $a_k \vee a_m \notin J$. Thus P is closed under join.

Similarly, suppose $a_k \wedge a_m \in P$ but $a_k \notin P$ and $a_m \notin P$. Let s be the least stage where $a_k \wedge a_m \in P_s$ and $a_k, a_m \notin P_s$. There is a prime ideal J extending P_s such that $a_k \wedge a_m \in J$ but $a_k, a_m \notin J$, contradicting that J is prime.

Therefore, P is a prime ideal in L. This ideal is computable because for any $a_k \in L$, there is a finite stage s in which we see that $a_k \in P_s$ or $a_k \notin P_s$ (and this never changes in future stages). □

Computable maximal and minimal prime ideals can also be found in a computable distributive lattice, with a few extra conditions. In both cases we need two safe "starter" elements to put in or leave out. For a computable maximal prime ideal, the only safe element to leave out is the greatest element 1 (otherwise, we may accidentally include the entire lattice in the prime ideal.) Furthermore, if the lattice does not have a greatest element, it may not have a maximal prime ideal at all. Likewise, for a computable minimal prime ideal, the only safe element to initially include is the least element 0. Again, if the lattice does not have a least element, it is possible that it has no minimal prime ideal.

Corollary 2.3.2. *Let L be a computable distributive lattice with 1. There exists a computable maximal prime ideal in L.*

Proof. Construct P as in Theorem 2.3.1, except at stage 0, put $a_1 \neq 1$ into P_0 and leave out $a_0 = 1$ (assuming without loss of generality that $a_0 = 1$). Also, at stage s, if neither conditions (a) or (b) from Theorem 2.3.1 hold, put a_s into P_s. Then P is still a computable prime ideal in L, and P is not the entire lattice because $1 \notin P$.

Suppose that there is another prime ideal J in L such that $P \subsetneq J$. Let $s > 0$ be the first stage where $P \neq J$; i.e., $a_{s+1} \in J$ but $a_{s+1} \notin P_s$. (We will not have $s = 0$ because $J \neq L$.) Let $a_{i_0}, a_{i_1}, \ldots, a_{i_k}$ be the elements in P_{s-1} and $a_{j_0}, a_{j_1}, \ldots, a_{j_\ell}$ the elements not in P_{s-1}. Note that $a_{i_0}, a_{i_1}, \ldots, a_{i_k} \in J$ and $a_{j_0}, a_{j_1}, \ldots, a_{j_\ell} \notin J$. Since $a_{s+1} \notin P_s$, it must be that $(a_{j_0}, a_{j_1}, \ldots, a_{j_\ell}) \leq (a_{i_0}, a_{i_1}, \ldots, a_{i_k} \vee a_{s+1})$, as this is the only condition forcing $a_{s+1} \notin P_s$. However, $a_{s+1} \in J$, and this condition contradicts the fact that J is closed downward. Thus, no such J exists, so P is a computable maximal prime ideal in L. □

Corollary 2.3.3. *Let L be a computable distributive lattice with 0. There exists a computable minimal prime ideal in L.*

Proof. Construct P as in Theorem 2.3.1, except at stage 0, put $a_0 = 0$ into P and leave $a_1 \neq 0$ out of P (assuming without loss of generality that $a_0 = 0$). Then P is still a computable prime ideal in L. Suppose that there is another prime ideal J in L such that $J \subsetneq P$. Let $s > 1$ be the first stage where $P \neq J$, so $a_{s+1} \in P_s$ but $a_{s+1} \notin J$. (We will not have $s = 0$ since every ideal contains 0.) Let $a_{i_0}, a_{i_1}, \ldots, a_{i_k} \in P_{s-1}$ and $a_{j_0}, a_{j_1}, \ldots, a_{j_\ell} \notin P_{s-1}$. Since $a_{s+1} \in P_s$, it must be that $(a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{s+1}) \leq (a_{i_0} \vee a_{i_1} \vee \cdots \vee a_{i_k})$, since this is the only condition forcing $a_{s+1} \in P_s$. Now $a_{i_0}, a_{i_1}, \ldots, a_{i_k} \in J$, so $(a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell} \wedge a_{s+1}) \in J$ since J is closed downward. However, $(a_{j_0} \wedge a_{j_1} \wedge \cdots \wedge a_{j_\ell}) \notin J$ and $a_{s+1} \notin J$, contradicting the fact that J is a prime ideal. Therefore, there is no such J properly contained in P, so P is a computable minimal prime ideal in L. □

2.4 Prime ideals in a non-distributive lattice

If a computable lattice is not distributive, then it is not always possible to find a computable prime ideal. We will show this by proving that any Π^0_1 class can be coded as the set of prime ideals of a computable non-distributive lattice with least and greatest elements. Since there are Π^0_1 classes with no computable members (cf. [9]), there will be computable non-distributive lattices with no computable prime ideals.

To build a computable non-distributive lattice, we will first need the concept of a partial lattice.

Definition 2.4.1. A partial lattice $(L, \leq, \wedge, \vee)$ is a set L with a partial order $\leq$ such that the meet and join operations ($\wedge$ and $\vee$) are defined on a subset of L.

It will also be useful to extend the idea of a partial lattice to a partial substructure, such as a partial filter.

Definition 2.4.2. A partial filter F is a subset of a partial lattice L such that for every $a, b \in L$:

$$\text{If } a \leq b \text{ and } a \in F, \text{ then } b \in F.$$

$$\text{If } a, b \in F \text{ and } a \wedge b \text{ is defined in } L, \text{ then } a \wedge b \in F.$$

We may similarly define its dual, a partial ideal.

Definition 2.4.3. A partial ideal I is a subset of a partial lattice L such that for every $a, b \in L$:

$$\text{If } a \geq b \text{ and } a \in I, \text{ then } b \in I.$$

$$\text{If } a, b \in I \text{ and } a \vee b \text{ is defined in } L, \text{ then } a \vee b \in I.$$

The idea will be to build a computable non-distributive lattice out of special "generator" elements $x_0, \hat{x_0}, x_1, \hat{x}_1, x_2, \hat{x}_2, \ldots$ which form an antichain, along with least element 0 and greatest element 1. We will also set $x_i \wedge \hat{x}_i = 0$ and $x_i \vee \hat{x}_i = 1$ for every $i \in \omega$, so x_i and $\hat{x}_i$ can be thought of as complements.

These generator elements will code a fixed $T \subseteq 2^{<\omega}$ into the prime ideals of L in the following way. Given a path $f \in [T]$, the prime ideal coded by f will include x_i or $\hat{x}_i$ for each $i \in \omega$ under the following convention. Denote

$$x_i^{f(i)} = \begin{cases} x_i & \text{if } f(i) = 1 \\ \hat{x}_i & \text{if } f(i) = 0 \end{cases}$$

Then the (nontrivial) prime ideal coded by f will be the ideal generated by $\{x_0^{f(0)}, x_1^{f(1)}, x_2^{f(2)}, \ldots\}$.

We will build L in stages. At stage 0, $L_0 = \{0, 1, x_0, \hat{x}_0\}$ with $0 < x_0, \hat{x}_0 < 1$, x_0 incomparable to $\hat{x}_0$, $x_0 \wedge \hat{x}_0 = 0$, and $x_0 \vee \hat{x}_0 = 1$.

At the beginning of stage $s + 1$, we will have a finite partial lattice L_s generated by $\{x_0, \hat{x}_0, x_1, \hat{x}_1, \ldots, x_s, \hat{x}_s\}$ under the partial meet and join operations with the following properties.

(P1) For each $i \leq s$, $x_i \wedge \hat{x}_i = 0$ and $x_i \vee \hat{x}_i = 1$.

(P2) All finite joins of x_i and $\hat{x}_i$ elements are defined for all $i \leq s$.

(P3) If $\sigma \in T$, $|\sigma| = n + 1 \leq s$, and σ has no extensions in T, then

$x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_n^{\sigma(n)} = 1.$

(P4) If $\sigma \in T$, $|\sigma| = n + 1 \leq s$, and σ has an extension in T, then

$x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_n^{\sigma(n)} < 1.$

(P5) If $u \wedge v$ is defined, then $u \wedge v = 0$ if and only if there is an $i \leq s$ and an $\varepsilon \in \{0, 1\}$ such that $u \leq x_i^{\varepsilon}$ and $v \leq x_i^{1-\varepsilon}$.

(P6) If $u \wedge v$ is defined, $u \wedge v \neq 0$, and $u \wedge v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ for some $i_0, \ldots, i_n \leq s$ and $\varepsilon_j \in \{0, 1\}$ (that is, $u \wedge v$ is bounded by a finite join of x_i and $\hat{x}_i$ elements), then $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$.

There are two special cases of this property.

(P6a) If $u \wedge v$ is defined, $u \wedge v \neq 0$, and there is a $\sigma \in T$ with $|\sigma| = n+1 \leq s$ for which $u \wedge v \leq x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_n^{\sigma(n)}$, then $u \leq x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_n^{\sigma(n)}$ or $v \leq x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_n^{\sigma(n)}$.

(P6b) If $u \wedge v$ is defined, $u \wedge v \neq 0$, and $u \wedge v \leq x_i^{\varepsilon}$ for some $i \leq s$ and $\varepsilon \in \{0, 1\}$, then $u \leq x_i^{\varepsilon}$ or $v \leq x_i^{\varepsilon}$.

It will be helpful to think of defining a partial filter or ideal in stages. If L_s is a finite partial lattice and $u, v \in L_s$, let $U = \{w : w \geq u\}$ and $V = \{w : w \geq v\}$. Let $F(U, V)$ be the (finite) partial filter generated by U and V. We will think of $F(U, V)$ as being defined inductively in finitely many stages as follows.

- $x \in F_0(U, V)$ if and only if $x \geq u$ or $x \geq v$.
- $x \in F_{t+1}(U, V)$ if and only if $x \geq a \wedge b$ for some $a, b \in F_t(U, V)$. (Note that since $a \geq a \wedge a$, $F_t(U, V) \subseteq F_{t+1}(U, V)$.)

Then $F(U, V) = F_t(U, V)$ where t is the least stage such that $F_{t+1}(U, V) = F_t(U, V)$.

We can define the partial ideal $I(U', V')$ generated by $U' = \{w : w \leq u\}$ and $V' = \{w : w \leq v\}$ similarly. To be precise, we will think of $I(U', V')$ as being defined inductively in finitely many stages in the following way.

- $x \in I_0(U', V')$ if and only if $x \leq u$ or $x \leq v$.
- $x \in I_{t+1}(U', V')$ if and only if $x \leq a \vee b$ for some $a, b \in I_t(U', V')$. (Note that since $a \leq a \vee a$, $I_t(U', V') \subseteq I_{t+1}(U', V')$.)

The next few lemmas are properties of partial filters of the form $F(U, V)$ as defined above that will be useful later.

Lemma 2.4.4. *Let L_s be a finite partial lattice satisfying (P1)-(P6).* $0 \in F(U, V)$ *if and only if there is an x_i with $i \leq s$ and an $\varepsilon \in \{0, 1\}$ such that $u \leq x_i^{\varepsilon}$ and $v \leq x_i^{1-\varepsilon}$.*

Proof. If there is an $i \leq s$ and $\varepsilon \in \{0,1\}$ such that $u \leq x_i^\varepsilon$ and $v \leq x_i^{1-\varepsilon}$, then $u \wedge v \leq x_i^\varepsilon \wedge x_i^{1-\varepsilon} = 0$, and $0 \in F(U,V)$.

Conversely, suppose that there is no such i and ε. We will show by induction on t that $0 \notin F_t(U,V)$ and $\{x_i, \hat{x}_i\} \not\subset F_t(U,V)$ for all $i \leq s$ and all t. By assumption, this holds for $F_0(U,V)$. Assume by induction that $0 \notin F_{t-1}(U,V)$ and $\{x_i, \hat{x}_i\} \not\subset F_{t-1}(U,V)$. Since $\{x_i, \hat{x}_i\} \not\subset F_{t-1}(U,V)$, $0 \notin F_t(U,V)$ by (P5) and by definition of $F_t(U,V)$. Suppose for a contradiction that $\{x_i, \hat{x}_i\} \subset F_t(U,V)$. Then $x_i \geq a \wedge b$ and $\hat{x}_i \geq c \wedge d$ for some $a,b,c,d \in F_{t-1}(U,V)$. By (P6b), $a \leq x_i$ or $b \leq x_i$. In either case, $x_i \in F_{t-1}(U,V)$. Similarly, by (P6b), $c \leq \hat{x}_i$ or $d \leq \hat{x}_i$, and hence $\hat{x}_i \in F_{t-1}(U,V)$. Together, these facts contradict that $\{x_i, \hat{x}_i\} \not\subset F_{t-1}(U,V)$. □

Lemma 2.4.5. *Let L_s be a finite partial lattice satisfying (P1)-(P6) and let $F(U,V)$ be as defined above for some fixed $u,v \in L_s$ such that $0 \notin F(U,V)$. If there is a finite join $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F(U,V)$, then $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$.*

Proof. Suppose $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F(U,V)$, so $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F_t(U,V)$ for some t. If $t = 0$, then $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ by definition. Otherwise, we have that $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \geq a \wedge b$ for some $a,b \in F_{t-1}(U,V)$. Since $0 \notin F(U,V)$, we have $a \wedge b \neq 0$. Thus, by (P6), $a \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $b \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$. In either case, $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F_{t-1}(U,V)$. Therefore, if $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F_t(U,V)$, it must be that $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F_0(U,V)$, and $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$. □

We will also need to know how to extend a finite partial lattice by defining one new meet and one new join in such a way that properties (P1)-(P6) are preserved. Let L_s be a finite partial lattice which satisfies (P1)-(P6), and fix $u, v \in L_s$ such that u and v are incomparable.

Suppose that we would like to define $u \wedge v$. Consider the set

$$M_{uv} = \{z \mid z < u \text{ and } z < v \text{ and } (\forall w \in L)[(w < u \text{ and } w < v) \Rightarrow w \not> z]\}.$$

In other words, M_{uv} represents the set of possible candidates for $u \wedge v$. There are two cases to consider. Either $|M_{uv}| \geq 2$ or $|M_{uv}| = 1$. In either case, we will show that it is possible to define $u \wedge v$ in the following lemmas.

Lemma 2.4.6. *If $|M_{uv}| \geq 2$, then it is possible to extend L_s to L_s' by adding a new element $z = u \wedge v$, and properties (P1)-(P6) are maintained in $L_s' = L_s \cup \{z\}$.*

Proof. Let $L_s' = L_s \cup \{z\}$, where $z \notin L_s$. Define the order on z as follows. For each $x \in L_s$, set

$$x < z \Leftrightarrow \exists y \in M_{uv}(x \leq y),$$

$$x > z \Leftrightarrow x \in F(U,V),$$

and x and z to be incomparable otherwise. Define $z = u \wedge v$.

We will verify that if $x \in F(U,V)$, then $x > y$ for every $y \in M_{uv}$ (which also implies that $0 \notin F(U,V)$). That is, the above definition does not lead to a contradiction where we set $x > z$ and $x < z$ simultaneously. We will show this by induction on t in $F_t(U,V)$. If $x \in F_0(U,V)$, then $x \geq u$ or $x \geq v$, and therefore $x > y$ for every $y \in M_{uv}$. Assume by induction that this property holds for all $x \in F_{t-1}(U,V)$. Let $x \in F_t(U,V)$. Then $x \geq a \wedge b$ for some $a, b \in F_{t-1}(U,V)$. Since $|M_{uv}| \geq 2$, let $y_1 \neq y_2 \in M_{uv}$. By the induction hypothesis, $a > y_1, y_2$ and $b > y_1, y_2$. Also, $y_1, y_2, a, b \in L_s$, and $a \wedge b$ is defined in the partial lattice L_s, so $a \wedge b \geq y_1, y_2$. However, since y_1 and y_2 are incomparable, these inequalities are strict, and hence $y_1, y_2 < a \wedge b \leq x$. Therefore, $y < x$ for every $y \in M_{uv}$.

It is also easy to see that $\leq$ is a partial order on L'_s. That is, the definition of the order on z preserves transitivity of $\leq$.

Notice that z is really the greatest lower bound of u and v, for if there is some $w \in L_s$ such that $w < u$ and $w < v$, then $w \leq y$ for some $y \in M_{uv}$. By definition, $w < z$.

Next, we will show that L'_s is a partial lattice. We must check that the addition of z does not interfere with any previously defined meets or joins in L_s. Suppose that $a \wedge b$ is defined in L_s and that $z < a, b$. We wish to show that $z < a \wedge b$. By definition, $a, b \in F(U,V)$. Then $a \wedge b \in F(U,V)$ because it is a partial filter. Therefore, $z < a \wedge b$. Similarly, suppose that $c \vee d \in L_s$ and $z > c, d$. We wish to show that $z > c \vee d$. Since $z > c, d$, both c and d are below u and

v. Therefore, $c \vee d \leq u, v$. Since u and v are incomparable, the inequality must be strict. Then there is some $y \in M_{uv}$ such that $c \vee d \leq y$. By definition, then, $z > c \vee d$.

Finally, we will check that properties (P1)-(P6) are maintained in L'_s. Properties (P1)-(P4) hold automatically since they are satisfied in L_s. Also, since $z \neq 0$, L'_s satisfies (P5). Suppose that $z \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ for some indices $i_0 < i_1 < \cdots < i_n$ in $\{0, 1, \ldots, s\}$ and each $\varepsilon_j \in \{0, 1\}$. Then, by definition, $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F(U, V)$. By Lemma 2.4.5, since L_s satisfies (P1)-(P6) and we have already seen that $0 \notin F(U, V)$, we have that $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$, and (P6) holds as desired. □

Lemma 2.4.7. *If $|M_{uv}| = 1$, then L_s may be extended to L'_s where $u \wedge v$ is defined (possibly by adding a new element), and properties (P1)-(P6) are maintained in L'_s.*

Proof. Suppose that $M_{uv} = \{a\}$ for some $a \in L_s$. If there is an $i \leq s$ and a $\varepsilon \in \{0, 1\}$ such that $u \leq x_i^{\varepsilon}$ and $v \leq x_i^{1-\varepsilon}$, then, by (P5), we must have that $a = 0$. Define $u \wedge v = 0$. This preserves (P5), and since no new elements were added to L_s, we automatically have that (P1)-(P4) and (P6) hold. In this case, it is easy to see that 0 is the greatest lower bound for u and v and that this definition does not change any previously defined meets or joins in L_s.

Otherwise, suppose that there is no such i and ε. By Lemma 2.4.4, $0 \notin F(U, V)$.

Next, check whether there is a finite join $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ for some $i_0 < i_1 < \cdots < i_n$ and $\varepsilon_j \in \{0, 1\}$ such that $a \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ but neither $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ nor $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$. If no such join exists, then define $u \wedge v = a$ in L'_s. By definition, a is the greatest lower bound of u and v, and L'_s remains a partial lattice since $a \in L_s$. Properties (P1)-(P6) are also preserved under this definition.

If there is such a finite join, then let $L'_s = L_s \cup \{z\}$ where $z \notin L_s$. Define the order on z as follows. For each $x \in L_s$, set

$$x < z \Leftrightarrow x \leq a,$$

$$x > z \Leftrightarrow x \in F(U, V),$$

and x and z to be incomparable otherwise. Define $z = u \wedge v$.

We will first verify that if $x \in F(U, V)$, then $x > a$, so we do not simultaneously attempt to define $x < z$ and $x > z$. We will show this by induction on t in $F_t(U, V)$. If $x \in F_0(U, V)$, then $x \geq u$ or $x \geq v$. In either case, $x \geq a$. Since a is strictly below u and v, we have that $x \neq a$, and therefore $x > a$. Assume by induction that this property holds for $F_{t-1}(U, V)$, and suppose that $x \in F_t(U, V)$. Then $x \geq b \wedge c$ for some $b, c \in F_{t-1}(U, V)$. By the induction hypothesis, $a < b, c$. Hence $a \leq b \wedge c$. We will argue that this inequality must be strict. Suppose for a contradiction that $a = b \wedge c$. By hypothesis, there is a finite join $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ such that $a \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ but neither $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ nor $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$. Since $a, b, c \in L_s$ and (P6) holds in L_s, we have that $b \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $c \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$. In either case, this means that $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F(U,V)$. Since $0 \notin F(U,V)$, we have that $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ by Lemma 2.4.5, yielding the desired contradiction. Therefore, $a < b \wedge c \leq x$ as required.

It is easy to see that the order on z preserves transitivity of $\leq$, so $\leq$ is a partial order on L'_s.

Also, if there is some $w \in L_s$ such that $w < u$ and $w < v$, then $w \leq a$, and by definition, $w < z$. Therefore, z is really the greatest lower bound of u and v.

To see that L'_s is a partial lattice, first suppose that $b \wedge c$ is defined in L_s and $z < b, c$. By definition, $b, c \in F(U,V)$. Since $F(U,V)$ is a partial filter, $b \wedge c \in F(U,V)$ as well, and hence $z < b \wedge c$. Now suppose that $d \vee e$ has been defined in L_s and $z > d, e$. Then $d, e \leq a$. Since L_s is a partial lattice, $d \vee e \leq a$, and therefore $d \vee e < z$. Thus, L'_s is still a partial lattice after the addition of z.

Lastly, we will check that properties (P1)-(P6) hold in L'_s. We automatically have that L'_s satisfies (P1)-(P4). It also satisfies (P5) since $z \neq 0$. To verify (P6), suppose that there is a finite join such that $z \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$. Then $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \in F(U,V)$. Since $0 \notin F(U,V)$, then we have that $u \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ or $v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ by Lemma 2.4.5. Therefore, (P6) holds in L'_s as well. □

Now suppose that we would like to define $u \vee v$ for fixed u, v incomparable in L_s which satisfies (P1)-(P6). If $u = x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ and $v = x_{j_0}^{\delta_0} \vee x_{j_1}^{\delta_1} \vee \cdots \vee x_{j_m}^{\delta_m}$

for some $i_0 < i_1 < \cdots < i_n$, $j_0 < j_1 < \cdots < j_m$, and $\varepsilon_j, \delta_k \in \{0, 1\}$, then $u \vee v$ may be defined in the following way.

If there is an $i_k = j_\ell$ and $\varepsilon_k = 1 - \delta_\ell$, then any element y above both u and v must also be above $x_{i_k}^{\varepsilon_k} \vee x_{j_\ell}^{1-\varepsilon_k} = 1$. In this case, define $u \vee v$ to be 1. If this does not happen, let $K = \{i_0, i_1, \ldots, i_n\} \cup \{j_0, j_1, \ldots, j_m\}$ and write $K = \{k_0, k_1, \ldots, k_\ell\}$. For each k_r, we know that the exponents of x_{k_r} in u and v must agree; let this exponent be β_r. Then define $u \vee v$ to be $x_{k_0}^{\beta_0} \vee x_{k_1}^{\beta_1} \vee \cdots \vee x_{k_\ell}^{\beta_\ell}$. (This gives a correct definition because the finite joins of x_i and $\hat{x}_i$ elements have already been defined correctly by property (P2).)

Now suppose that at least one of u or v is not of the form $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ for some $i_0 < i_1 < \cdots < i_n$ and $\varepsilon_j \in \{0, 1\}$. Consider the set

$$m_{uv} = \{z \mid z > u \text{ and } z > v \text{ and } (\forall w \in L)[(w > u \text{ and } w > v) \Rightarrow w \not< z]\}.$$

Similar to before, m_{uv} represents the set of possible candidates for $u \vee v$. Once again, there are two cases to consider; $|m_{uv}| \geq 2$ or $|m_{uv}| = 1$. In either case, we will show that L_s can be extended to L'_s in which $u \vee v$ is defined.

Lemma 2.4.8. *If $|m_{uv}| \geq 2$, then it is possible to extend L_s to L'_s by adding a new element $z = u \vee v$, and properties (P1)-(P6) are maintained in $L'_s = L_s \cup \{z\}$.*

Proof. Let $L'_s = L_s \cup \{z\}$, where $z \notin L_s$. Define the order on z as follows. For each $x \in L_s$, set

$$x > z \Leftrightarrow \exists y \in m_{uv}(x \geq y),$$

$$x < z \Leftrightarrow x \in I(U', V'),$$

and x and z to be incomparable otherwise. Define $z = u \vee v$.

First, we will show that if $x \in I(U', V')$, then $x < y$ for every $y \in m_{uv}$. This will ensure that the above definition does not force $x < z$ and $x > z$ simultaneously for any $x \in L_s$. Proceed by induction on t in $I_t(U', V')$. If $x \in I_0(U', V')$, then $x \leq u$ or $x \leq v$, and therefore $x < y$ for every $y \in m_{uv}$. Assume by induction that this property holds for $x \in I_{t-1}(U', V')$. Let $x \in I_t(U', V')$. Then $x \leq a \vee b$ for some $a, b \in I_{t-1}(U', V')$. Since $|m_{uv}| \geq 2$, let $y_1 \neq y_2 \in m_{uv}$. By induction, we have that $a < y_1, y_2$ and $b < y_1, y_2$. Therefore, $a \vee b \leq y_1, y_2$. Since y_1 and y_2 are incomparable, it must be that $x \leq a \vee b < y_1, y_2$. Thus, $x < y$ for every $y \in m_{uv}$.

It is easy to see that $\leq$ on L'_s is a partial order, since the order on z preserves transitivity of $\leq$.

Also, if there is some $w \in L_s$ such that $w > u, v$, then $w \geq y$ for some $y \in m_{uv}$. By definition, then, $w > z$. Hence z really is the least upper bound of u and v.

Now we will verify that L'_s is a partial lattice. Suppose that $a \wedge b$ was defined in L_s and that $z < a, b$. Then a and b are both above u and v. Since L_s is a partial lattice, we have that $a \wedge b \geq u, v$. Since u and v are incomparable, this inequality must be strict. Then there is some $y \in m_{uv}$ such that $a \wedge b \geq y$. Therefore,

$z < a \wedge b$ by definition. Next, suppose that $c \vee d$ was defined in L_s and that $z > c, d$. Then $c, d \in I(U', V')$. Since $I(U', V')$ is a partial ideal, $c \vee d \in I(U', V')$. Therefore, $z > c \vee d$.

Lastly, we will show that L_s' preserves properties (P1)-(P6). We automatically have that L_s' satisfies (P1)-(P4) since they are satisfied in L_s. Property (P5) is also satisfied because no new meets were added to L_s'. Property (P6) also holds because no new meets were satisfied, and because z is a new element, $z = u \vee v$ is not of the form $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ for any $i_0 < i_1 < \cdots < i_n$ and $\varepsilon_j \in \{0, 1\}$. □

Lemma 2.4.9. *If $|m_{uv}| = 1$, then L_s may be extended to L_s' where $u \vee v$ is defined in such a way that L_s' remains a partial lattice and properties (P1)-(P6) are maintained.*

Proof. Suppose that $|m_{uv}| = \{a\}$ for some $a \in L_s$. Define $u \vee v = a$ in L_s'. It is clear that a is the least upper bound of u and v. Since no new elements were added to L_s', it is still a partial lattice. Because no new meets were defined and no new elements were added to L_s', properties (P1)-(P6) are still satisfied. (If $a = x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ for any $i_0 < i_1 < \cdots < i_n$ and $\varepsilon_j \in \{0, 1\}$, then any meet less than a in L_s' was also less than a in L_s.) □

The following lemma shows that the ideal coded by f is a prime ideal in any lattice which satisfies properties (P1)-(P6).

Lemma 2.4.10. *Let $T \subseteq 2^{<\omega}$ and L be a lattice such that $L = \cup L_s$ and each L_s*

satisfies (P1)-(P6) with bound s. Then, for any path $f \in [T]$*, the ideal generated by* $\{x_0^{f(0)}, x_1^{f(1)}, \ldots\}$ *is prime in* L*.*

Proof. Let I be the ideal generated by $\{x_0^{f(0)}, x_1^{f(1)}, \ldots\}$, and suppose that $u \wedge v \in I$. If $u \leq x_i^{\varepsilon}$ and $v \leq x_i^{1-\varepsilon}$ for $\varepsilon \in \{0,1\}$, then since $f(i) = \varepsilon$ or $f(i) = 1-\varepsilon$ and $a \in I$ for every $a \leq x_i^{f(i)}$, at least one of $u \in I$ or $v \in I$.

Otherwise, by (P5), $u \wedge v \neq 0$. Since I is closed downward and under joins, $u \wedge v \leq x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_n^{f(n)}$ for some n. Then, by (P6), $u \leq x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_n^{f(n)}$ or $v \leq x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_n^{f(n)}$. Therefore, $u \in I$ or $v \in I$.

Furthermore, I nontrivial. Clearly $I \neq \emptyset$ since it contains one of x_i or $\hat{x}_i$ for every $i \in \omega$. The fact that $I \neq L$ follows from (P3) and (P4), since a join $x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_n^{\sigma(n)} = 1$ only if $\sigma \in T$ does not have an extension. For any path f, there is always an extension of $f \upharpoonright n$ for any $n \in \omega$, so $x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_n^{\sigma(n)}$ is always set to be strictly less than 1. Hence $1 \notin I$, so $I \neq L$. □

Theorem 2.4.11. *Every* Π_1^0*-class can be coded as the set of prime ideals of a computable (non-distributive) lattice with least and greatest elements.*

Proof. Let $T \subseteq 2^{<\omega}$ be given. We will build a computable (non-distributive) lattice $(L, \leq, \wedge, \vee)$ such that

$$F : [T] \to \text{ Prime ideals of L}$$

is a bijection and preserves Turing degree.

As mentioned above, we will build L in stages. At each stage, the partial lattice L_s will be extended to L_{s+1} by adding new generator elements, defining some meets and joins of generator elements, and defining one new meet and one new join.

Stage 0: Let $L_0 = \{0, 1, x_0, \hat{x}_0\}$ with $0 < x_0, \hat{x}_0 < 1$, x_0 incomparable to $\hat{x}_0$, $x_0 \wedge \hat{x}_0 = 0$, and $x_0 \vee \hat{x}_0 = 1$.

Stage s+1: Suppose by induction that we have a partial lattice L_s generated by the antichain of x_i and $\hat{x}_i$ elements for $i \leq s$ which satisfies (P1)-(P6). We will extend L_s to L_{s+1} by first adding new elements x_{s+1} and $\hat{x}_{s+1}$ to maintain these properties with bound $s+1$ (i.e., the properties will hold for $i \leq s+1$ and $\sigma \in T$ with $|\sigma| = n+1 \leq s+1$). Then we will define one new meet and one new join.

Let $L'_{s+1} = L_s \cup \{x_{s+1}, \hat{x}_{s+1}\}$, where $0 < x_{s+1}, \hat{x}_{s+1} < 1$, and x_{s+1} and $\hat{x}_{s+1}$ are incomparable to all other elements of L_s (and each other). Then define $x_{s+1} \wedge \hat{x}_{s+1} = 0$ and $x_{s+1} \vee \hat{x}_{s+1} = 1$. Now L'_{s+1} satisfies (P1).

In order for L'_{s+1} to satisfy (P2)-(P4), add new elements and define new joins as follows. For each $n < s$ and each choice of indices $i_0 < i_1 < \cdots < i_n$ from $\{0, 1, \ldots, s\}$ and each choice of $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_n \in \{0, 1\}$, check whether $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} = 1$ in L_s. If so, set $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \vee x_{s+1}^{\varepsilon} = 1$. If not, add a new element defined as $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \vee x_{s+1}^{\varepsilon}$. Order these elements (with each other and all elements of L_s) as follows. Set this element to be greater than 0 and strictly less than 1. We define $y \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \vee x_{s+1}^{\varepsilon}$ if and only if either

(1) $y \in L_s$ and $y \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$, or

(2) $y = x_{s+1}^{\varepsilon}$, or

(3) $y = x_{j_0}^{\varepsilon_{j_0}} \vee x_{j_1}^{\varepsilon_{j_1}} \vee \cdots \vee x_{i_k}^{\varepsilon_{i_k}} \vee x_{s+1}^{\varepsilon}$ for some indices $j_0 < j_1 < \cdots < j_k$ from $\{i_0, i_1, \ldots, i_n\}$.

L'_{s+1} satisfies (P5) because the only new meet defined in L'_{s+1} that was not defined in L_s is $x_{s+1} \wedge \hat{x}_{s+1} = 0$. Also, notice that all new joins in L'_{s+1} have the form $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \vee x_{s+1}^{\varepsilon}$ for indices $i_0 < i_1 < \cdots < i_n$ in $\{0, 1, \ldots, s\}$ and $\varepsilon, \varepsilon_j \in \{0, 1\}$. Suppose $u \wedge v$ is defined in L'_{s+1} and $u \wedge v \neq 0$ (hence, $u \wedge v$ was defined in L_s). If $u \wedge v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n} \vee x_{s+1}^{\varepsilon}$, then by construction, either (1) $u \wedge v \in L_s$ and $u \wedge v \leq x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$, or (2) $u \wedge v = x_{s+1}^{\varepsilon}$, or (3) $u \wedge v = x_{j_0}^{\varepsilon_{j_0}} \vee x_{j_1}^{\varepsilon_{j_1}} \vee \cdots \vee x_{j_n}^{\varepsilon_{j_n}} \vee x_{s+1}^{\varepsilon}$ for indices $j_0 < j_1 < \cdots < j_k$ from $\{i_0, i_1, \ldots, i_n\}$. Clearly (2) and (3) do not hold because $u \wedge v \in L_s$. Therefore, (1) holds, and it follows that L'_{s+1} satisfies (P6) by induction.

Next, for each σ of length $s+2$, we define $x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_{s+1}^{\sigma(s+1)}$ in the following way. If $\sigma \notin T$ or if $\sigma \in T$ but has σ has no extensions in T, then set $x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_{s+1}^{\sigma(s+1)} = 1$. If σ does have an extension in T, then add a new element to L_{s+1} for $x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_{s+1}^{\sigma(s+1)}$. Set this element to be above 0 and strictly below 1. Also, define $y \leq x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_{s+1}^{\sigma(s+1)}$ if and only if $y \leq x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_s^{\sigma(s)}$ or $y \leq x_{s+1}^{\sigma(s+1)}$.

Clearly, the resulting structure after defining these new joins is a finite partial lattice which satisfies (P1)-(P4). Since no additional meets have been defined which were not defined in L'_{s+1}, by induction we have that (P5) is maintained.

To see that (P6) is still satisfied, as no new meets have been defined, and the only new joins are of the form $x_0^{\sigma_0} \vee x_1^{\sigma_1} \vee \cdots \vee x_{s+1}^{\sigma_{s+1}}$ for some $\sigma \in T$ of length $s+2$, suppose that $u \wedge v \neq 0$ is defined and that $u \wedge v \leq x_0^{\sigma_0} \vee x_1^{\sigma_1} \vee \cdots \vee x_{s+1}^{\sigma_{s+1}}$. If $x_0^{\sigma_0} \vee x_1^{\sigma_1} \vee \cdots \vee x_{s+1}^{\sigma_{s+1}} = 1$, then (P6) holds because we already have $u, v \leq 1$. Otherwise, $x_0^{\sigma_0} \vee x_1^{\sigma_1} \vee \cdots \vee x_{s+1}^{\sigma_{s+1}}$ was added as a new element, and either (1) $u \wedge v \leq x_0^{\sigma_0} \vee x_1^{\sigma_1} \vee \cdots \vee x_s^{\sigma_s}$, or (2) $u \wedge v \leq x_{s+1}^{\sigma_{s+1}}$. However, the only element below $x_{s+1}^{\sigma_{s+1}}$ is 0, so (2) contradicts the fact that $u \wedge v \neq 0$. Therefore, (1) holds, and (P6) is satisfied by induction.

Now we will define one new meet and one new join in L'_{s+1}. Let u and v be the $\mathbb{N}$-least elements such that $u \wedge v$ is not yet defined. If $u \leq v$, then $u \wedge v = u$. Similarly, if $u \geq v$, then $u \wedge v = v$. Otherwise, suppose that u and v are incomparable. Consider the set

$$M_{uv} = \{z \mid z < u \text{ and } z < v \text{ and } (\forall w \in L)[(w < u \text{ and } w < v) \Rightarrow w \not> z]\}.$$

If $|M_{uv}| \geq 2$, then Lemma 2.4.6 shows how to extend L'_{s+1} to a finite partial lattice L''_{s+1} in which $u \wedge v$ is defined and properties (P1)-(P6) are maintained with bound $s+1$. In the other case, if $|M_{uv}| = 1$, then Lemma 2.4.7 gives us the extension to L''_{s+1}.

Next, search for the $\leq_{\mathbb{N}}$-least pair of elements u, v such that $u \vee v$ is not yet defined in L''_{s+1}. If $u \leq v$, set $u \vee v = v$. Similarly, if $u \geq v$, set $u \vee v = u$. Otherwise, suppose that u and v are incomparable. If $u = x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ and $v = x_{j_0}^{\delta_0} \vee x_{j_1}^{\delta_1} \vee \cdots \vee x_{j_m}^{\delta_m}$ for $i_0 < i_1 < \cdots < i_n$, $j_0 < j_1 < \cdots < j_m$, and $\varepsilon_j, \delta_k \in \{0,1\}$, then define $u \vee v$ as in the comments above Lemma 2.4.8. Otherwise, suppose at least one of u or v is not of the form $x_{i_0}^{\varepsilon_0} \vee x_{i_1}^{\varepsilon_1} \vee \cdots \vee x_{i_n}^{\varepsilon_n}$ for $i_0 < i_1 < \cdots < i_n$ and $\varepsilon_j \in \{0,1\}$. Consider the set

$$m_{uv} = \{z \mid z > u \text{ and } z > v \text{ and } (\forall w \in L)[(w > u \text{ and } w > v) \Rightarrow w \not< z]\}.$$

If $|m_{uv}| \geq 2$ or $|m_{uv}| = 1$, then L''_{s+1} may be extended to a finite partial lattice in which $u \vee v$ is defined and properties (P1)-(P6) are maintained with bound $s+1$ by Lemma 2.4.8 and Lemma 2.4.9, respectively. Let L_{s+1} be the resulting structure.

In the end, let $L = \cup_s L_s$. L is a partial order in which every pair of elements has a least upper bound and greatest lower bound defined at some finite stage s, and this definition never changes at future stages $t > s$. Therefore, L is a computable lattice. Also, properties (P1)-(P6) are satisfied with bound s for every in $L_s \subset L$.

By Lemma 2.4.10, the ideal generated by $I_f = \{x_0^{f(0)}, x_1^{f(1)}, \dots\}$ is prime for each $f \in [T]$.

Finally, we show that the map $F : [T] \to$ prime ideals of L is a bijection which preserves Turing degree.

To see that F is injective, let $f_1, f_2 \in [T]$ with $f_1 \neq f_2$. Also, let $F(f_1) = I_{f_1}$ and $F(f_2) = I_{f_2}$. Then there is some i such that $f_1(i) \neq f_2(i)$, so one of $x_i, \hat{x}_i$ is in I_{f_1} and the other is in I_{f_2}. If $I_{f_1} = I_{f_2}$, then both $x_i, \hat{x}_i \in I_{f_1}$, and since I_{f_1} is an ideal, $x_i \vee \hat{x}_i \in I_{f_1}$. However, $x_i \vee \hat{x}_i = 1$, meaning $I_{f_1} = L$, but I_{f_1} is a nontrivial prime ideal in L.

Now let I be a prime ideal in L. Since $0 \in I$ and $x_i \wedge \hat{x}_i = 0$ for every $i \in \omega$, I contains at least one of x_i or $\hat{x}_i$ for every $i \in \omega$. Also, since $I \neq L$ and $x_i \vee \hat{x}_i = 1$ for every $i \in \omega$, I does not contain both x_i and $\hat{x}_i$. Let $f(i) = 1$ if $x_i \in I$ and $f(i) = 0$ if $\hat{x} \in I$. Then $f \in [T]$, since if there is a $\sigma \in T$ such that $|\sigma| = n$, σ does not have an extension of length $n+1$ in T, and f and σ agree on the first n elements, then by construction, $x_0^{\sigma(0)} \vee x_1^{\sigma(1)} \vee \cdots \vee x_n^{\sigma(n)} = x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_n^{f(n)} = 1$ in L. Since I is closed under join, $x_1^{f(1)} \vee \cdots \vee x_n^{f(n)} = 1 \in I$, contradicting that $I \neq L$. Thus, $f \in [T]$ and $F(f) = I$. Therefore, F is surjective.

For a fixed $f \in [T]$, let $F(f) = I_f$. The path f can easily be computed from I_f since $f(i) = 1$ if and only if $x_i \in I_f$ and $f(i) = 0$ if and only if $x_i \notin I_f$. Thus $f \leq_T I_f$.

In the other direction, let $w \in L$. We will show how to compute whether $w \in I_f$ or $w \notin I_f$ from f. If $w = 0$, then $w \in I_f$, and if $w = 1$, then $w \notin I_f$. Otherwise, let s be the first stage such that w enters L_s. Check whether $w \leq x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_s^{f(s)}$. If so, then $w \in I_f$. If not, then we will show that $w \notin I_f$. Suppose for a contradiction that $w \not\leq x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_s^{f(s)}$ and $w \in I_f$.

Then there must be a least stage $t > s$ such that $w \leq x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_t^{f(t)}$ in L_t. Since $t > s$, $w \in L_{t-1}$. By definition, we have that either (1) $w \leq x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_{t-1}^{f(t-1)}$ or (2) $w \leq x_t^{f(t)}$. Since t is the least such that w is below a finite join of the form $x_0^{f(0)} \vee x_1^{f(1)} \vee \cdots \vee x_t^{f(t)}$, (1) cannot hold. Also, (2) is not possible since the only element less than $x_t^{f(t)}$ at stage t is 0. Therefore, $w \notin I_f$, and $I_f \leq_T f$.

Lastly, notice that L is not distributive. If it were, then we would have

$$(x_0 \vee x_1) \wedge (\hat{x}_0 \wedge \hat{x}_1) = (x_0 \wedge \hat{x}_0 \wedge \hat{x}_1) \vee (x_1 \wedge \hat{x}_0 \wedge \hat{x}_1) = 0 \vee 0 = 0.$$

However, since L satisfies (P5), and it is not true that both $u = x_0 \vee x_1 \not\leq x_i^{\varepsilon}$ and $v = \hat{x}_0 \wedge \hat{x}_1 \leq x_i^{1-\varepsilon}$ for any $i \in \omega$ and $\varepsilon \in \{0, 1\}$, a new element $z = (x_0 \vee x_1) \wedge (\hat{x}_0 \wedge \hat{x}_1) > 0$ would be added in the construction of L. Then $z > 0$ and $z = 0$, a contradiction. Thus, L is not distributive. $\square$

Chapter 3

Degree spectra of lattices

3.1 Background

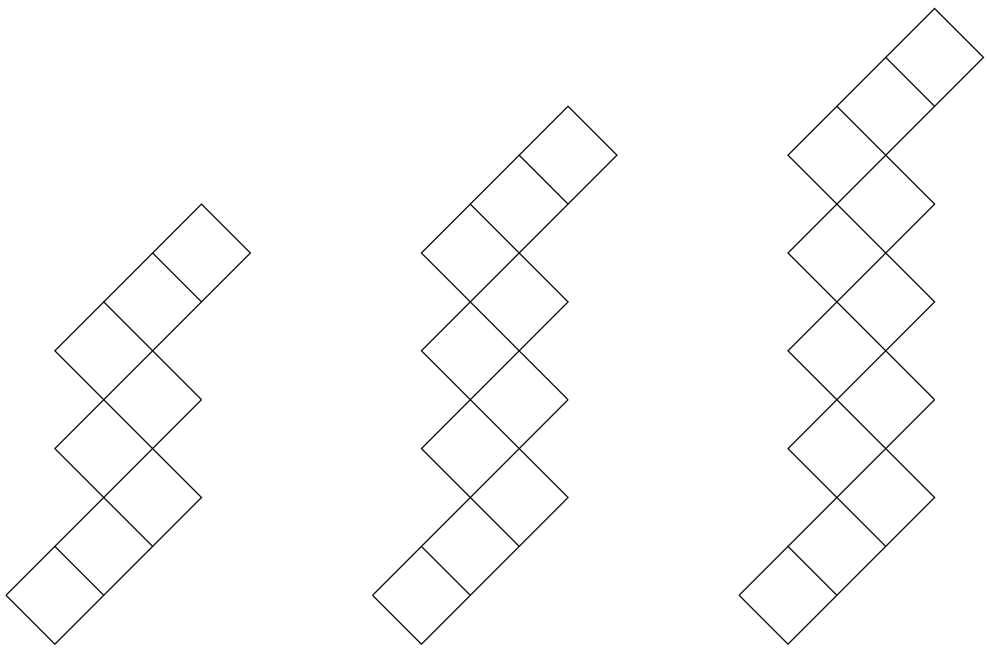
We would like to know the degree spectra of lattices as in Definition 1.1.6. It is known that there is a general (non-distributive) lattice with degree spectra $\{\mathbf{c} \mid \mathbf{c} \geq \mathbf{d}\}$ for any Turing degree $\mathbf{d}$ (cf. [8,16]). The following results will be combined to show that the same is true for distributive lattices, pseudocomplemented lattices, and Heyting algebras.

Theorem 3.1.1 (Selivanov [17])**.** *There exists a computable sequence of finite distributive lattices $\{L_i\}_{i\in\omega}$ such that L_i does not embed into L_j for any $i \neq j$.*

The sequence of distributive lattices in Theorem 3.1.1 is such that each L_i has sixteen elements in common (eight at the bottom and eight at the top) and $i + 1$ "blocks" in the middle formed out of $4i + 2$ additional elements. The first three lattices in the sequence are shown in Figure 3.1.

Theorem 3.1.2 (Richter [16])**.** *Fix an algebraic structure (e.g., group or lattice). If there is a computable sequence of finite algebraic structures $\{\mathcal{A}_i\}_{i\in\omega}$ such that*

$\mathcal{A}_i$ does not embed into $\mathcal{A}_j$ for $i \neq j$, and, for each $S \subseteq \omega$, there is an algebraic structure $\mathcal{A}_S$ such that $\mathcal{A}_S \leq_T S$, and $\mathcal{A}_i$ embeds into $\mathcal{A}_S$ if and only if $i \in S$, then, for each $\mathbf{d}$, there is an algebraic structure whose isomorphism class has degree $\mathbf{d}$.



L_0 L_1 L_2

Fig. 3.1: The sequence L_i for $i = 0, 1, 2$

3.2 Degree spectra for distributive lattices

Theorems 3.1.1 and 3.1.2 together yield the following result.

Theorem 3.2.1. *For any Turing degree $\mathbf{d}$, there is a distributive lattice, a pseudocomplemented lattice and a Heyting algebra with degree spectra $\{\mathbf{c} \mid \mathbf{c} \geq \mathbf{d}\}$.*

Proof. Since each L_i in the sequence from Theorem 3.1.1 is a finite distributive lattice, each L_i is also a Heyting algebra (and hence also a pseudocomplemented lattice). Also, for any $i \neq j$, L_i does not embed into L_j as a lattice, so L_i also does not embed into L_j as a pseudocomplemented lattice or Heyting algebra.

Fix $S = \{i_0, i_1, i_2, \dots\} \subseteq \omega$. We can build an infinite Heyting algebra $L_S = L_{i_0} \oplus L_{i_1} \oplus L_{i_2} \oplus \dots$, where $L_S \leq_T S$ since we add the block L_i to L_S if $i \in S$. If, instead, we are thinking of each L_i as a distributive lattice or pseudocomplemented lattice, then L_S is an infinite distributive lattice or pseudocomplemented lattice, respectively.

The same argument to show that L_i does not embed into L_j (as a distributive lattice) for $i \neq j$ shows that if $i \notin S$, then L_i does not embed into L_S. The key fact is that, due to the stacking of the lattices, for any $x, y \in L_S$ where x is in a copy of L_i and y is in a copy of L_j with L_i below L_j in L_S, $x < y$. In other words, if x and y are incomparable, then they must be in the same copy of L_i in L_S.

For each L_i, there are exactly two elements a and b that are incomparable to four other elements in L_i (see Figure 3.2). Every other element of L_i is incomparable to only three or fewer other elements. Therefore, any embedding of L_i into L_S must send a to $\hat{a}$ or $\hat{b}$ where $\hat{a}, \hat{b}$ sit in a copy of L_j in L_S for some $j \in \omega$ and $\hat{a} < \hat{b}$. If a is mapped to $\hat{b}$, then $z_1 < z_2 < z_3 < z_4$, the four elements incomparable to a in L_i, must map to $\hat{y}_4 < \hat{y}_3 < \hat{y}_2 < \hat{y}_1$, the four elements incomparable to $\hat{b}$ in a copy of L_j in L_S. However, then we have that

$a \vee z_2 = a \vee z_3$, but $\hat{b} \vee \hat{y}_3 \neq \hat{b} \vee \hat{y}_2$. Thus $a \in L_i$ must be mapped to $\hat{a}$ in a copy of L_j in L_S.

Once $a \mapsto \hat{a}$ is fixed, this forces $z_1 \mapsto \hat{z}_1$, $z_2 \mapsto \hat{z}_2$, $z_3 \mapsto \hat{z}_3$, $z_4 \mapsto \hat{z}_4$, $a \wedge z_2 \wedge z_3 \mapsto \hat{a} \wedge \hat{z}_2 \wedge \hat{z}_3$, $a \wedge z_3 \mapsto \hat{a} \wedge \hat{z}_3$, $a \vee z_3 \mapsto \hat{a} \vee \hat{z}_3$, and $a \vee z_3 \vee z_4 \mapsto \hat{a} \vee \hat{z}_3 \vee \hat{z}_4$. Now consider $c_0 \in L_i$. This element is incomparable to z_4. The only elements incomparable to $\hat{z}_4$ are $\hat{a}, \hat{a} \vee \hat{z}_3$, and $\hat{c}_0$. Since $\hat{a}$ and $\hat{a} \vee \hat{z}_3$ are already images of elements in L_i, the only valid choice for the image of c_0 is $\hat{c}_0$. This forces $w_1 = c_0 \vee a \vee z_3 \vee z_4 \mapsto \hat{c}_0 \vee \hat{a} \vee \hat{z}_3 \vee \hat{z}_4 = \hat{w}_1$.

Now consider c_1, which is incomparable to c_0. The only elements incomparable to $\hat{c}_0$ are $\hat{z}_4$, $\hat{a} \vee \hat{z}_3 \vee \hat{z}_4$, and $\hat{c}_1$. The first two are already images under the embedding, so it must be that $c_1 \mapsto \hat{c}_1$. Similarly, we must have $c_2 \mapsto \hat{c}_2, c_3 \mapsto \hat{c}_3, \ldots, c_{2i-1} \mapsto \hat{c}_{2i-1}$.

If $j < i$ for this copy of L_j in L_S, then at some point we map $c_k \mapsto \hat{b}$ for some odd number k. Suppose the three elements incomparable to c_k are $c_{k-1} < d < c_{k+1}$. In this case, we have already sent $c_{k-1} \mapsto \hat{y}_4$. This forces $c_{k-1} \vee c_k \mapsto \hat{y}_4 \vee \hat{b} = \hat{b} \vee \hat{y}_3$. Now d is incomparable to c_k, so its image must be incomparable to $\hat{b}$. The elements incomparable to $\hat{b}$ are y_1, y_2, y_3, and y_4, but y_4 is already the image of c_{k-1}. Also, $d < c_k \vee c_{k-1}$, so its image must be less than $\hat{b} \vee \hat{y}_3$. Of $\hat{y}_1, \hat{y}_2$, and $\hat{y}_3$, only $\hat{y}_3 < \hat{b} \vee \hat{y}_3$, so we must have $d \mapsto \hat{y}_3$. Since c_{k+1} is incomparable to c_k, its image must be incomparable to $\hat{b}$ in L_S, making $\hat{y}_1$ and $\hat{y}_2$ the only choices for the image of c_{k+1}.

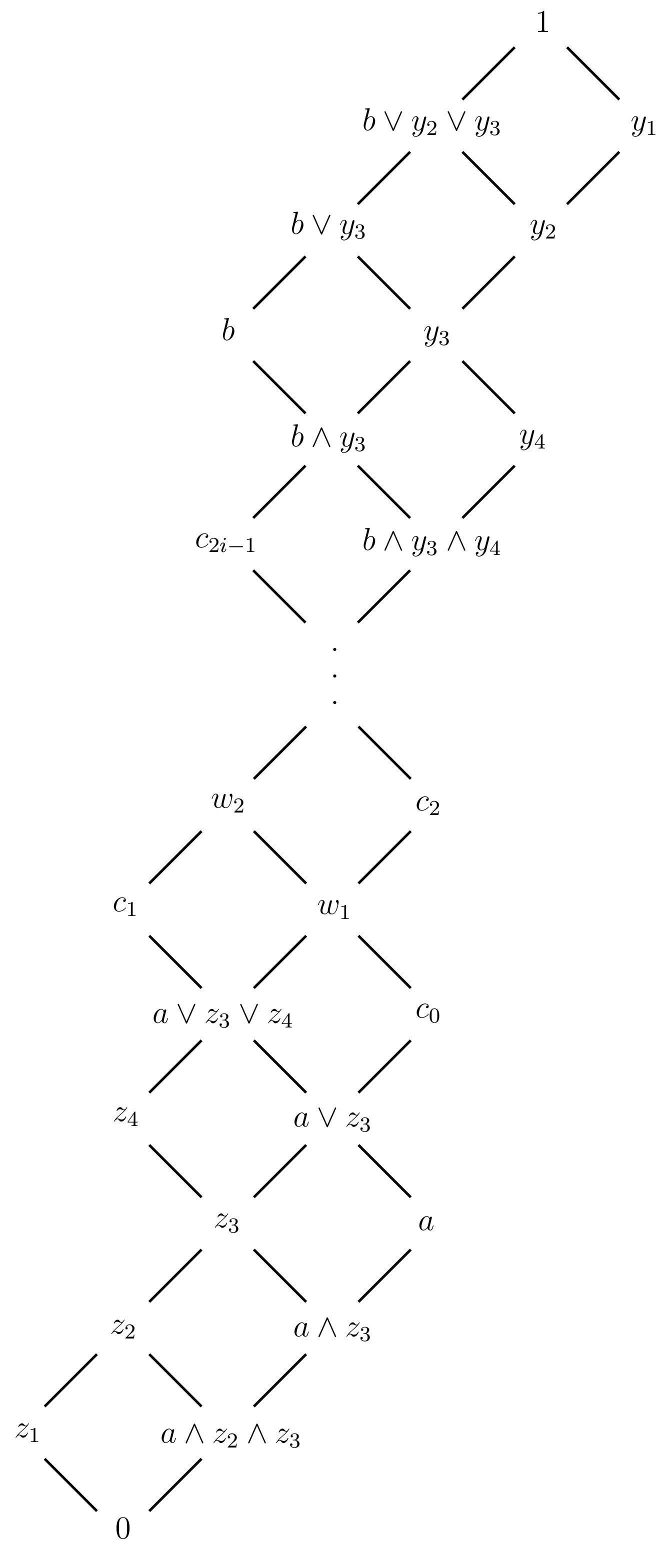

1
$b \vee y_2 \vee y_3$
y_1
$b \vee y_3$
y_2
b
y_3
$b \wedge y_3$
y_4
c_{2i-1}
$b \wedge y_3 \wedge y_4$
w_2
c_2
c_1
w_1
$a \vee z_3 \vee z_4$
c_0
z_4
$a \vee z_3$
z_3
a
z_2
$a \wedge z_3$
z_1
$a \wedge z_2 \wedge z_3$
0

Fig. 3.2: L_i

Next, we seek to map c_{k+2} (possibly b) to an element in L_S. Only four elements remain in the copy of L_j in L_S which are not already the image of something from L_i: $\hat{y}_2, \hat{y}_1, \hat{b} \vee \hat{y}_2 \vee \hat{y}_3$, and 1. Since c_{k+2} is incomparable to c_{k+1} and two other distinct elements, its image must be incomparable to three elements, one of which being $\hat{y}_1$ or $\hat{y}_2$. However, $\hat{y}_2$ is incomparable to only two elements, $\hat{b} \vee \hat{y}_2 \vee \hat{y}_3$ is incomparable to only one element, and 1 is comparable to every other element in L_S. While $\hat{y}_1$ is incomparable to three elements, one of these includes $\hat{b}$, but $c_{k+2} > c_k$, so c_{k+2} cannot be sent to $\hat{y}_1$. There is nowhere in L_S to send c_{k+2}, so it is not possible to have $i > j$.

Finally, if $i < j$, we have the reverse problem; we send b to $\hat{c}_k$ for some odd number k. Since b is incomparable to four elements and $\hat{c}_k$ is incomparable to at most three elements, the embedding will fail for $i < j$.

Therefore, the only embedding of L_i into L_S is the one which sends L_i into the copy of itself in L_S. That is, L_i embeds into L_S if and only if $i \in S$. By Theorem 3.1.2, then, the degree spectra of L_S (as a distributive lattice, pseudocomplemented lattice, or Heyting algebra) is $\{\mathbf{c} \mid \mathbf{c} \geq \mathbf{d}\}$ for any fixed Turing degree $\mathbf{d}$. □

Roughly speaking, this means that one can code information into the isomorphism type of a Heyting algebra instead of just coding information into particular presentations. Recall from section 1.1 that this is not the case for Boolean algebras, for if the isomorphism type of a Boolean algebras has a least degree, then it must be **0** (cf. [16]). In this sense, distributive lattices, pseudocomplemented lattices, and Heyting algebras behave more like general (non-distributive) lattices than Boolean algebras. Contrast this to what we have seen in the previous chapter; when searching for prime ideals effectively, distributive lattices (and pseudocomplemented lattices and Heyting algebras) are more similar to Boolean algebras than general (non-distributive) lattices.

Chapter 4

Computable dimension of Heyting algebras

4.1 Free Heyting algebras and intuitionistic logic

Intuitionistic logic is classical logic without the law of the excluded middle. That is, $\varphi \vee \neg\varphi$ is not intuitionistically valid. We will mostly work with propositional intuitionistic logic over the language $(X, \wedge, \vee, \rightarrow, \bot, \top)$, where $X = \{x_0, x_1, x_2, \dots\}$ is the set of propositional variables, $\wedge, \vee$, and $\rightarrow$ are the logical connectives, $\bot$ is "false," and $\top$ is "truth." Denote provability under intuitionistic logic by $\vdash_i$. Let $\varphi \equiv_i \psi$ denote that the formula φ is equivalent to the formula ψ under intuitionistic logic, meaning that $\vdash_i \varphi \leftrightarrow \psi$. Like propositional classical logic, propositional intuitionistic logic is decidable (cf., e.g., [11]).

A special class of formulas are those which cannot be written as a disjunction of other nonequivalent formulas. Propositional intuitionistic variables have this property.

Definition 4.1.1. An intuitionistic formula $\varphi \not\equiv_i \bot$ is called *join-irreducible* if, whenever $\varphi \equiv_i \psi_1 \vee \psi_2$, then either $\varphi \equiv_i \psi_1$ or $\varphi \equiv_i \psi_2$. A formula which is not join-irreducible is called *join-reducible.*

Join-reducible formulas in intuitionistic logic have the following nice decomposition property. This can be used to analyze the complexity of the set of join-reducible or join-irreducible formulas.

Theorem 4.1.2 (O'Connor [13])**.** *Let φ be a formula in intuitionistic logic. If φ is join-reducible, then $\varphi \equiv_i \varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n$, where each φ_i is a conjunction of subformulas of φ and $\varphi_i \not\equiv_i \varphi$.*

Corollary 4.1.3. *Let φ be a formula in intuitionistic logic. It is computable to determine whether φ is join-reducible or join-irreducible.*

Proof. There are finitely many subformulas of φ and therefore finitely many conjunctions of these subformulas (not counting the conjunction of a subformula with itself). Therefore, we can check all of the finitely many combinations $\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n$, where each φ_i is a conjunction of subformulas of φ and $\varphi_i \not\equiv_i \varphi$. By Theorem 4.1.2, if φ is join-reducible, we will find that $\varphi \equiv_i \varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n$ for some combination. Otherwise, φ is join-irreducible. □

Corollary 4.1.4. *Propositional variables in intuitionistic logic are join-irreducible.*

Proof. This follows from Theorem 4.1.2 since the only subformula of a propositional variable x is x itself. □

The following substitution lemma for formulas in propositional intuitionistic logic will also be useful later.

Lemma 4.1.5. *Let $\varphi(x)$ and $\psi(x)$ be formulas in intuitionistic logic. If x_m and x_n are new variables not appearing in either formula, then the following equivalence holds.*

$$\varphi(x) \equiv_i \psi(x) \Leftrightarrow \varphi\left[\frac{x_m \vee x_n}{x}\right] \equiv_i \psi\left[\frac{x_m \vee x_n}{x}\right].$$

Proof. Let $\mathrm{Var}(\varphi)$ denote the set of propositional variables in φ. Given an intuitionistic propositional formula $\varphi(x)$ with $x \in \mathrm{Var}(\varphi)$ and $x_m, x_n \notin \mathrm{Var}(\varphi)$, let $\hat{\varphi}$ be the formula $\varphi\left[(x_m \vee x_n)/x\right]$. Then $\mathrm{Var}(\hat{\varphi})$ includes x_m and x_n but not x. It follows directly from Theorem 5.2.4 in [19] that

$$\varphi \equiv_i \psi \Rightarrow \hat{\varphi} \equiv_i \hat{\psi}.$$

Now let α be an intuitionistic propositional formula with $x_m, x_n \in \mathrm{Var}(\alpha)$ but $x \notin \mathrm{Var}(\alpha)$, and let α' be the formula $\alpha[x/x_m, x/x_n]$. Again by Theorem 5.2.4 in [19] we have that

$$\alpha \equiv_i \beta \Rightarrow \alpha' \equiv_i \beta'.$$

Combining these, we have

$$\varphi \equiv_i \psi \Rightarrow \hat{\varphi} \equiv_i \hat{\psi} \Rightarrow \hat{\varphi}' \equiv_i \hat{\psi}'.$$

Notice that $\hat{\varphi}' = \varphi[(x \vee x)/x]$, and since $x \vee x \equiv_i x$, $\hat{\varphi}' \equiv_i \varphi$. Therefore,

$$\varphi \equiv_i \psi \Rightarrow \hat{\varphi} \equiv_i \hat{\psi} \Rightarrow \hat{\varphi}' \equiv_i \hat{\psi}' \Rightarrow \varphi \equiv_i \psi$$

as desired. □

Heyting algebras are models of intuitionistic logic. The free Heyting algebra on finitely many generators is defined as follows.

Definition 4.1.6. The *free Heyting algebra on n generators* is the Heyting algebra formed from the equivalence class of formulas of n propositional variables under provable equivalence in intuitionistic logic.

Currently, only the free Heyting algebra on one generator has been completely characterized (cf. [12]). It is an infinite lattice, as there are infinitely many nonequivalent intuitionistic formulas of one propositional variable. The free Heyting algebra on infinitely many generators is similarly defined for infinitely many propositional variables. Let H_n and H_ω denote the free Heyting algebra on n and ω generators, respectively.

4.2 Computable dimension of free Heyting algebras

The free Heyting algebras provide a good starting point for investigating the computable dimension of general Heyting algebras because of their connection to intuitionistic logic. It is easy to see that the free Heyting algebra on n generators is computably categorical because any isomorphism is completely determined by the n generators, which is finitely much information. On the other hand, we will see that H_ω is not computably categorical.

For the following theorem, we think of H_ω as the collection of propositional formulas in infinitely many variables modulo equivalence under intuitionistic logic.

More formally, fix a Gödel numbering of the propositional formulas of intuitionistic logic and assume that the code for a formula $\varphi \wedge \psi$, $\varphi \vee \psi$, or $\varphi \to \psi$ is always greater than the codes for φ and ψ. Let the elements of H_ω be the equivalence classes $[\varphi(\bar{x})]$ under provable equivalence in intuitionistic logic. Since propositional intuitionistic logic is decidable, this gives a computable copy of the free Heyting algebra on ω generators. The operations are computed in the obvious way:

$$[\varphi] \leq [\psi] \text{ if and only if } \vdash_i \varphi \to \psi,$$

$$[\varphi] \wedge [\psi] = [\varphi \wedge \psi],$$

$$[\varphi] \vee [\psi] = [\varphi \vee \psi].$$

Let this copy be fixed.

We will build $\hat{H}_\omega$, a computable copy which is not computably isomorphic to H_ω. Once a number n enters the domain of $\hat{H}_\omega$ in the construction, it is given a label at stage s, denoted by $\alpha_s(n)$. The label is a propositional formula in intuitionistic logic, and it should satisfy the following properties:

$$\alpha(n) = \lim_s \alpha_s(n) \text{ exists;} \tag{4.2.1}$$

$$n \neq m \Rightarrow \alpha(n) \not\equiv_i \alpha(m); \tag{4.2.2}$$

$$\text{For each } \varphi \text{ there is an } n \text{ such that } \alpha(n) \equiv_i \varphi. \tag{4.2.3}$$

The function α will be the Δ^0_2 isomorphism from $\hat{H}_\omega$ to H_ω. Condition (4.2.1) says that α is a total Δ^0_2 function. Conditions (4.2.2) and (4.2.3) say that α is one-to-one and onto, respectively.

To ensure α is an isomorphism, we will have the following requirements.

$$C_e \ : \quad \text{For all } \ \square \in \{\wedge, \vee, \rightarrow\}, \text{ and for all } s \in \omega, \ k, \ell, m \in \hat{H}_\omega,$$

$$\text{If } \alpha_s(k) \ \square \ \alpha_s(\ell) \equiv_i \alpha_s(m), \text{ then for all } t \geq s,$$

$$\alpha_t(k) \ \square \ \alpha_t(\ell) \equiv_i \alpha_t(m).$$

These requirements will also make $\hat{H}_\omega$ computable, since once we define the join, meet, or relative pseudocomplement of elements, these relationships never change in future stages (even though the labels themselves may change).

Finally, we have requirements D_e to diagonalize against all possible computable isomorphisms.

$$D_e \ : \quad \varphi_e : \hat{H}_\omega \rightarrow H_\omega \text{ is not an isomorphism}$$

The main action of D_e is the following.

- At stage s of the construction, D_e picks a large index $s(e)$, takes the first number n not in the domain of $\hat{H}_\omega$ yet, and defines $\alpha_s(n) = x_{s(e)}$. We call n the witness for D_e.

- At stages $t > s$, we check if $\varphi_{e,t}(n)$ converges. If not, there is nothing to do.

- If $\varphi_{e,t}(n)$ does converge, we check whether $\varphi_{e,t}(n)$ is the code of a join irreducible formula in H_ω (using Corollary 4.1.3). If not, then φ_e is not an isomorphism by Corollary 4.1.4, since it maps a join-irreducible element to a join-reducible element. If this happens, we say D_e is satisfied.

- If $\varphi_{e,t}(n)$ converged and is the code for a join irreducible intuitionistic formula, then we say that D_e is waiting to diagonalize. When D_e is next eligible to act, we diagonalize as follows. Let m_1 and m_2 be large numbers (not in the domain of $\hat{H}_\omega$), and let x_i and x_{i+1} be large indexed propositional variables (that have never been used before). Set $\alpha_s(m_1) = x_i$ and $\alpha_s(m_2) = x_{i+1}$. Change the label of n to $\alpha_s(n) = x_i \vee x_{i+1}$. (Now $\alpha_s(n)$ is join-reducible, and D_e will be satisfies as long as $\alpha(n) = \alpha_s(n)$, which will follow in the general construction.) Similarly, for any label that is currently using $x_{s(e)}$, change the label by substituting $x_i \vee x_{i+1}$ for $x_{s(e)}$ (Plus perform some other clean-up duties to add new elements to map to formulas which were labels and are no longer labels.)

We are now ready to go through the construction of $\hat{H}_\omega$.

Theorem 4.2.1. *H_ω is not computably categorical.*

Proof. We will build $\hat{H}_\omega$ in stages.

Stage 0: Let $\hat{H}_{\omega,0} = \{0, 1\}$ and define $\alpha_0(0) = \bot$ and $\alpha_0(1) = \top$. (These values will never change.)

Stage $s+1$: Assume that we have already defined $\hat{H}_{\omega,s}$ and α_s. We first take care of the next D_e requirement that needs attention. Let D_e be the highest priority diagonalization requirement that is either waiting to diagonalize or is waiting to define its witness.

If D_e is waiting to define its witness, then let n be the $\leq_{\mathbb{N}}$-least element not in the domain of $\hat{H}_{\omega,s}$. Let $s(e)$ be a large index. Put n into $\hat{H}_{\omega,s+1}$ and set $\alpha_{s+1}(n) = x_{s(e)}$. For all other m in the domain of $\hat{H}_{\omega,s}$, set $\alpha_{s+1}(m) = \alpha_s(m)$.

If D_e is waiting to diagonalize, we do the following.

- *Splitting*: As described above, split n into the join of m_1 and m_2. Set $\alpha_{s+1}(m_1) = x_i$, $\alpha_{s+1}(m_2) = x_{i+1}$, and $\alpha_{s+1}(n) = x_i \vee x_{i+1}$.

- *Relabeling*: For all other m in the domain of $\hat{H}_{\omega,s}$, if $x_{s(e)}$ occurs in $\alpha_s(m)$, then set

  $$\alpha_{s+1}(m) = \alpha_s(m) \left[\frac{x_i \vee x_{i+1}}{x_{s(e)}} \right].$$

  If $x_{s(e)}$ does not occur in $\alpha_s(m)$, then set $\alpha_{s+1}(m) = \alpha_s(m)$.

  For each formula φ which was in the range of α_s and is not currently in the range of α_{s+1}, add a new large number ℓ to $\hat{H}_{\omega,s+1}$ and map $\alpha_{s+1}(\ell) = \varphi$.

Second, at stage $s+1$, we extend the range of α_{s+1}. Let φ be the formula with the least Gödel number which is not intuitionistically equivalent to any formula currently in the range of α_{s+1}. Let k be the $\leq_{\mathbb{N}}$-least number not currently in $\hat{H}_{\omega,s+1}$. Put k into $\hat{H}_{\omega,s+1}$ and set $\alpha_{s+1}(k) = \varphi$.

This ends the construction.

Note that the requirement D_e picks a new variable $x_{s(e)}$ (not already seen in the construction) to split if φ_e looks like an isomorphism. If s is the first stage at which $\alpha_s(n)$ is defined, then the label of n can change only once for each of the finitely many variables x_i occurring in $\alpha_s(n)$ which is a witness for some D_e. Therefore, an element's label changes at most finitely often, and each α_s reaches a limit.

Next we will show that the requirement C_e is satisfied for $\vee$ ($\wedge$ and $\rightarrow$ follow similarly). Fix s and elements k, ℓ, and m in the domain of $\hat{H}_\omega$, and suppose $\alpha_s(k) \vee \alpha_s(\ell) \equiv_i \alpha_s(m)$. If $\alpha_t = \alpha_s$ on these elements for all $t > s$, there is nothing to show. Otherwise, let t be the least stage such that $\alpha_t \neq \alpha_s$ for at one element. By the construction, α_t differs from α_s by substituting in $x_i \vee x_{i+1}$ for $x_{s(e)}$, where x_i and x_{i+1} do not appear in $\alpha_s(n)$ for any n so far. Notice that if any element n does not mention $x_{s(e)}$, then

$$\alpha_t(n) = \alpha_s(n) \left[\frac{x_i \vee x_{i+1}}{x_{s(e)}} \right] = \alpha_s(n).$$

Then, by Lemma 4.1.5,

$$\begin{aligned} & \alpha_s(k) \vee \alpha_s(\ell) \equiv_i \alpha_s(m) \\ \Leftrightarrow \quad & \big(\alpha_s(k) \vee \alpha_s(\ell)\big) \left[\frac{x_i \vee x_{i+1}}{x_{s(e)}} \right] \equiv_i \alpha_s(m) \left[\frac{x_i \vee x_{i+1}}{x_{s(e)}} \right] \\ \Leftrightarrow \quad & \alpha_s(k) \left[\frac{x_i \vee x_{i+1}}{x_{s(e)}} \right] \vee \alpha_s(\ell) \left[\frac{x_i \vee x_{i+1}}{x_{s(e)}} \right] \equiv_i \alpha_s(m) \left[\frac{x_i \vee x_{i+1}}{x_{s(e)}} \right] \\ \Leftrightarrow \quad & \alpha_t(k) \vee \alpha_t(\ell) \equiv_i \alpha_t(m). \end{aligned}$$

To see that the map α is one-to-one and onto, suppose that $\alpha(m) \equiv_i \alpha(n)$ but $m \neq n$. Then from the construction, it must be that the substitution of $x_i \vee x_{i+1}$ for $x_{s(e)}$ made $\alpha(m)$ and $\alpha(n)$ intuitionistically equivalent. That is, there is some least stage t where $\alpha_s(m) \not\equiv \alpha_s(n)$ for every $s < t$, but $\alpha_t(m) \equiv_i \alpha_t(n)$. This contradicts the fact that the requirements C_e are satisfied (again using Lemma 4.1.5).

The labeling map is onto because every intuitionistic propositional formula φ is eventually the image of some n in the domain of $\hat{H}_\omega$ by the construction, and if n is relabeled, the second part of the relabeling procedure picks a new m so that $\alpha_{s+1}(m) = \varphi$. Relabeling is done only finitely often (if s is the first stage at which $\alpha_s(n) = \varphi$, then we pick a new natural number m such that $\alpha_t(m) = \varphi$ at a later stage $t > s$ at most once for each variable occurring in φ which is a witness for some D_e requirement), so there is some element which is mapped to φ in the end.

Finally, as noted above, meeting the C_e requirements ensures that $\hat{H}_\omega$ is computable and isomorphic to H_ω. The labeling map α is a bijection, and the C_e requirements give us that α preserves $\wedge, \vee$, and $\rightarrow$, thus also preserving $\leq$. □

The following result will give us that the computable dimension of H_ω is ω.

Theorem 4.2.2 (Goncharov [6])**.** *If $\mathcal{A}$ and $\mathcal{B}$ are computable structures which are Δ^0_2 isomorphic but not computably isomorphic, then the computable dimension of $\mathcal{A}$ is ω.*

Corollary 4.2.3. *The computable dimension of H_ω is ω.*

Proof. This follows from Theorem 4.2.2, since the isomorphism $\alpha : \hat{H}_\omega \rightarrow H_\omega$ in Theorem 4.2.1 is Δ^0_2 and not computable. □

This completely characterizes the computably categorical free Heyting algebras; a free Heyting algebra is computably categorical if and only if it has finitely many generators. In contrast, the free Boolean algebra on infinitely many generators (the countable atomless Boolean algebra) is computably categorical (cf. [14]).

The proof of Theorem 4.2.1 uses the generators to diagonalize against all partial computable functions. The next theorem is about the complexity of the set of join-irreducible elements of H_ω (which contains the generators).

Theorem 4.2.4. *Let* **c** *be any c.e. degree. There is a computable copy of* H_ω *in which the set of join-irreducible elements has Turing degree* **c**.

Proof. Let C be an infinite c.e. set which is not computable. We will build a copy of H_ω with requirements C_e as in Theorem 4.2.1, modifying the requirements D_e so that the element initially labeled by x_{2e} is join-irreducible if and only if $e \notin C$. To achieve this, let stage 0 be as before, and at stage $s+1$, D_e picks the first natural number n not yet in the domain of $\hat{H}_\omega$ and sets $\alpha_s(n) = x_{2e}$. Then, if $e \in C_s$, we say that D_e is waiting to diagonalize. When it is allowed to act at a stage $t > s$, it chooses large numbers $2m+1$ and $2m+3$ (larger than any index currently in the domain of $\hat{H}_\omega$) and splits x_{2e} by setting $\alpha_t(n) = x_{2m+1} \vee x_{2m+3}$ and relabels all other elements of $\hat{H}_\omega$ appropriately. Also, extend the range of α_{s+1} at stage $s+1$ as before; find the least formula φ which is not currently in the range of α_{s+1}, the least natural number n which is not currently in $\hat{H}_\omega$, and set $\alpha_{s+1}(n) = \varphi$.

Notice that, if $\alpha_s(n) = \varphi$, then the label for $n \in \hat{H}_\omega$ will change at most once for each even indexed variable in φ which is currently a witness for some D_e at stage s. Once x_{2e} is split into the join of x_{2m+1} and x_{2m+3}, these odd indexed variables are never split at a later stage. Therefore, $\lim_s \alpha_s$ exists, and $\hat{H}_\omega$ is computable.

Let $\hat{J}_\omega$ denote the set of join-irreducible elements of $\hat{H}_\omega$. To compute C from $\hat{J}_\omega$, run through the construction of $\hat{H}_\omega$ above and wait until n enters the domain of $\hat{H}_\omega$ at some stage s with $\alpha_s(n) = x_{2e}$. If $n \in \hat{J}_\omega$, then $\alpha(n) = \alpha_s(n)$ (x_{2e} is never split), so $e \notin C$. Otherwise, $e \in C$. Conversely, knowing C, to determine if $n \in \hat{J}_\omega$, run through the construction of $\hat{H}_\omega$ until n enters $\hat{H}_\omega$ at some stage s (for the sake of satisfying some C_e or D_e). Then $\alpha_s(n)$ will be an intuitionistic formula of finitely many variables $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$. For every j, $1 \leq j \leq k$, if any i_j is even and $i_j/2 \in C$, then x_{i_j} will eventually be split. Continue running through the construction until the stage $t \geq s$ where every x_{i_j} has already been split if it is supposed to be (according to whether $i_j/2 \in C$). Then $\alpha(n) = \alpha_t(n)$, and it is computable to check whether the intuitionistic formula $\alpha(n)$ is join-irreducible. If so, then $n \in \hat{J}_\omega$. Otherwise, $n \notin \hat{J}_\omega$. Therefore, $C \equiv_T \hat{J}_\omega$. □

4.3 Computably categorical Heyting algebras

In addition to the finitely generated free Heyting algebras, we will show another class of Heyting algebras is computably categorical. Namely, we will consider

Heyting algebras of the form $L(F)$ from section 1.2, where F is a finite Heyting algebra and L is a linear order, and the Heyting algebras have the L-good property defined below. These results can also be stated for distributive lattices, so we will obtain a class of distributive lattices which are computably categorical as well.

Moreover, we will show that there is a computably categorical Heyting algebra which is not computably categorical as a distributive lattice. On the other hand, if a Heyting algebra is computably categorical as a distributive lattice, then it is also computably categorical as a Heyting algebra. Thus, we will prove that the converse of this statement does not always hold.

Definition 4.3.1. A finite distributive lattice F is called *L-good* if, whenever F embeds into $L(F)$, the image must be of the form F_x for some $x \in L$.

The embedding in Definition 4.3.1 is a lattice embedding; that is, the map preserves $\leq$, $\wedge$, and $\vee$. A Heyting algebra embedding, abbreviated as "HA-embedding," must also preserve $\rightarrow$. Distributive lattice embeddings will be abbreviated as "DL-embeddings."

The following is a sufficient for determining whether a finite distributive lattice is L-good.

Lemma 4.3.2. *If $|F| > 2$, and if for every $a \in F$ with $a \neq 0, 1$, there is a $b \in F$ such that a and b are not comparable, then $L(F)$ is L-good.*

Proof. For a contradiction, suppose that F is not L-good and fix an image of F in $L(F)$ which lies in more than one F_x component. (We will view F as this copy of itself inside $L(F)$.) Fix F_x such that $0_F \in F_x$ and fix a minimal nonzero element a_0 of F . We claim that $a_0 \in F_x$ as well. Clearly, we cannot have $a_0 \in F_y$ for $y <_L x$ for then we would have $0_F <_F a_0$ and $a_0 <_{L(F)} 0_F$, violating the embedding. Suppose $a_0 \in F_y$ for some $y >_L x$. By hypothesis, there is an incomparable element $b_0 \in F$ and we can assume that $a_0 \wedge_F b_0 = 0_F$. If $b_0 \in F_z$ for $z <_L y$, then in the embedded copy of F, we have $b_0 <_{L(F)} a_0$ and so $a_0 \wedge_{L(F)} b_0 = b_0$, which is incorrect. Similarly, $b_0 \notin F_z$ for $z >_L y$: if $b_0 \in F_z$, then $a_0 <_{L(F)} b_0$, so $a_0 \wedge_{L(F)} b_0 = a_0$. If $b_0 \in F_y$, then $a_0 \wedge_{L(F)} b_0 \in F_y$ which is not equal to (the embedded image of) 0_F. Therefore, $a_0 \in F_x$.

Now consider $a_0 \vee b_0$. Because a_0 and b_0 are incomparable, $a_0 \vee b_0$ is strictly above both a_0 and b_0. Let $a_1 = (a_0 \vee b_0)$. We know a_1 cannot be 1_F because then all of F would lie in F_x (since we'd have the images of 0_F and 1_F in F_x). Also, clearly $a_1 \neq 0$. So again, a_1 has some element $b_1 \in F$ such that a_1 and b_1 are incomparable. (Note b_1 cannot be either a_0 or b_0.) Again, $a_1 \vee b_1$ must be strictly above a_1 and b_1. Again, $a_1 \vee b_1$ cannot be 1_F because all of F would lie in F_x. Repeat using the hypothesis. However, F is a finite lattice, and this process will generate an infinite chain in F:

$$a_0 < a_1 = (a_0 \vee b_0) < a_2 = (a_1 \vee b_1) < a_3 = (a_2 \vee b_2) < \dots$$

Thus F cannot be split by the map such that part of it is sent to F_x and part is sent to F_y for $x <_L y$. □

Lemma 4.3.3. *Let L be a computable dense linear order without endpoints and* *let F be a finite distributive lattice or Heyting algebra which is L-good. The computable structure (distributive lattice or Heyting algebra) $L(F)$ is computably categorical.*

Proof. Fix $L(F)$, a computable "nice" copy in which it is possible to determine $x \in L$ such that $a \in F_x$ for each $a \in L(F)$. Let D be a computable distributive lattice or Heyting algebra which is isomorphic to $L(F)$. Use a back-and-forth argument to build the computable isomorphism $\varphi : D \to L(F)$. Initially let $\varphi(0_D) = 0_L$ and $\varphi(1_D) = 1_L$.

Forward direction: Let n be the $\leq_{\mathbb{N}}$-least element of D that is not yet in the domain of φ. Search in D for a copy of the lattice F that includes n. Call this copy $F' \subset D$. Since φ will be defined for all points of each copy of F at the same time, all points of F' are not currently in the domain of φ. Once F' is found, because D is L-good, F' must be of the form F_x for some $x \in L$. Currently, φ maps only finitely many copies of F from $D \to L(F)$, so there are finitely many copies of F already mapped to the left of F' (copies of the form F_y for $y <_L x$) and finitely many copies of F to the right (copies of the form F_y for $y >_L x$). Find a copy of F in $L(F)$ which has the same number of copies of F to the left and to the right. (This can be done since $L(F)$ is a computable "nice" copy.) Then define $\varphi(F') = F$ by mapping each $a \in F'$ to the corresponding $a \in F$.

Backward direction: Let n be the $\leq_{\mathbb{N}}$-least element of $L(F)$ such that n is not yet in the range of φ. Since $L(F)$ is the "nice" copy, we can find $x \in L$ and

$F_x \subset L(F)$ such that $n \in F_x$. By construction, φ does not yet map into any of the members of this F_x. As above, there are only finitely many copies of F which have already been mapped to the left of F_x and finitely many copies to the right. Search in D for a copy of F which has the same number of copies of F to the left and to the right. Call this copy F' in D. D is L-good, so $F' = F_y$ for some $y \in L$. Therefore, define $\varphi(F') = F$ by matching up elements as in the forward direction. □

Let L be a countable dense linear order without endpoints, and let F_1 and F_2 be finite distributive lattices or Heyting algebras. Also, let D_1 and D_2 be subsets of L which partition L and are dense in L. That is, $D_1 \cup D_2 = L$, $D_1 \cap D_2 = \emptyset$, and for any $x <_L y$, $[x, y] \cap D_1 \neq \emptyset$ and $[x, y] \cap D_2 \neq \emptyset$. Form the structure $L(F_1, F_2)$ by replacing each point $x \in D_1$ by a copy of F_1 (call this copy $F_{1,x}$) and replacing each point $x \in D_2$ by a copy of F_2 (call this copy $F_{2,x}$). As above, $L(F_1, F_2)$ is a computable distributive lattice or Heyting algebra. We will refer to $F_{1,x}$ and $F_{2,x}$ as components of the lattice $L(F_1, F_2)$.

Say that the pair (F_1, F_2) is L-*good* if, whenever F_i (for $i = 1$ or 2) is embedded in $L(F_1, F_2)$, then the image of this embedded copy of F_i has to lie entirely in some $F_{1,x}$ or some $F_{2,x}$. That is, in this definition, we allow F_1 to embed into $L(F_1, F_2)$ by embedding into a component of $L(F_1, F_2)$, but we do not allow F_1 to embed in such a way that part of it lies in one component and part of it lies in another component.

The basic idea will be to build a Heyting algebra H of the form $L(F_1, F_2)$, where L is a countable dense linear order without endpoints and the pair (F_1, F_2) is L-good, such that $L(F_1, F_2)$ is computably categorical as a Heyting algebra but not as a distributive lattice. Also, the finite Heyting algebras F_1 and F_2 will have the following properties.

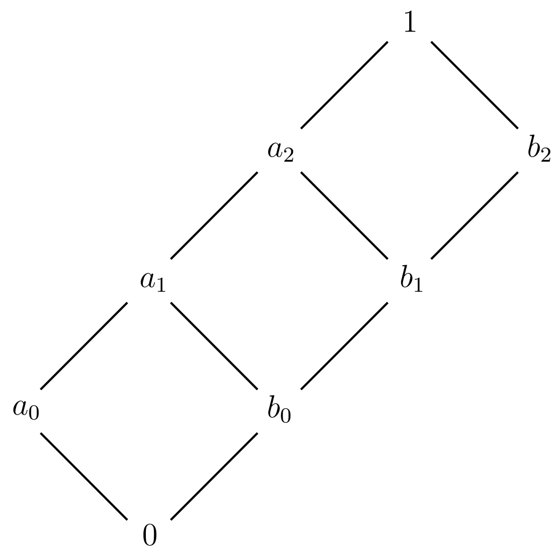
(P1) F_1 DL-embeds into F_2.

(P2) If we have a copy of F_1 DL-embedded inside F_2, then using $\rightarrow$, we can tell that this copy of F_1 sits inside a copy of F_2 and is not a complete copy of F_1.

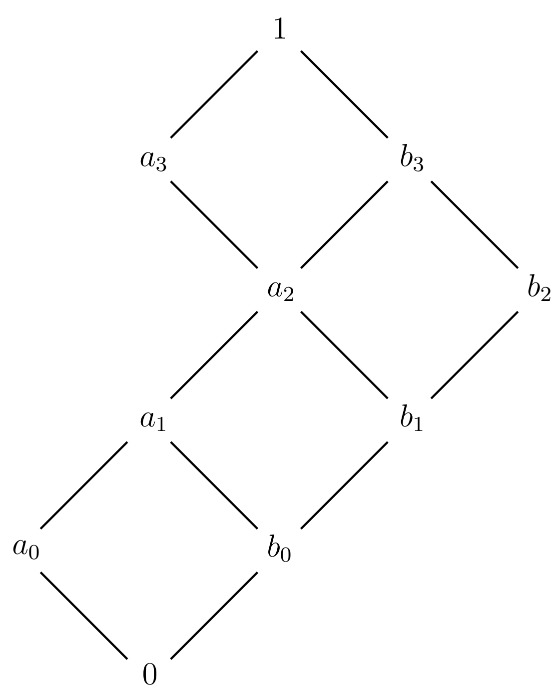
Consider the finite distributive lattices (which are also Heyting algebras) F_1 and F_2 given in Figure 4.1. We will verify that F_1 and F_2 satisfy the above desired properties and that (F_1, F_2) is L-good.

Lemma 4.3.4. *F_1 DL-embeds into F_2.*

Proof. Clearly F_1 embeds into F_2 by matching up points with the same names (with 1 to b_3). In F_1, a_0 and b_2 have three distinct elements which are not comparable to them. All other elements have fewer distinct non-comparable elements (b_0 has one, a_1 has two, b_1 has two, a_2 has one, and 0 and 1 are comparable to every other element). Therefore, when F_1 embeds into F_2, a_0 and b_2 must be sent to elements of F_2 which are not comparable to at least three other distinct elements (possibly others). In F_2, a_0 is not comparable to three elements, b_2 is



F_1



F_2

Fig. 4.1: Heyting algebras F_1 and F_2

not comparable to four elements, and all other elements are not comparable to fewer than three elements. If $a_0 \in F_1$ were mapped to any element other than a_0 or $b_2 \in F_2$, there would not be three incomparable elements to map $b_0, b_1, b_2 \in F_1$ to. Similarly, $b_2 \in F_1$ cannot be mapped to any element other than a_0 or b_2 in F_2. However, if a_0 is mapped to b_2, there are three elements above a_0 in F_1 and only two elements above b_2 in F_2, so there would be no room for all of the elements above a_0 in F_2. Hence the only embedding of F_1 into F_2 is the obvious one. □

Lemma 4.3.5. *The $\rightarrow$ operation can distinguish between F_1 DL-embedded into F_1 itself or F_1 DL-embedded into F_2.*

Proof. Notice that if F_1 DL-embeds into F_2, then

$$b_2 \rightarrow a_2 = a_3,$$

but if F_1 embeds into F_1 itself, then

$$b_2 \rightarrow a_2 = a_2.$$

In other words, if $\varphi : F_1 \rightarrow F_2$ is the embedding,

$$\varphi(b_2 \rightarrow_{F_1} a_2) \neq \varphi(b_2) \rightarrow_{F_2} \varphi(a_2).$$ □

Remark 4.3.6. *Using just the $\vee$ and $\wedge$ operations, though, it is not possible to tell whether F_1 is inside F_2 or a copy of itself, since*

$$a_2 \vee b_2 = b_3 = \varphi(1), \text{ and}$$

$$a_2 \wedge b_2 = b_1.$$

Lemma 4.3.7. *If L is a countable dense linear order without endpoints and F_1, F_2 are as above, then (F_1, F_2) is L-good.*

Proof. Both F_1 and F_2 have the property that every element other than 0 or 1 is not comparable with at least one other element. Then (F_1, F_2) is L-good by an argument similar to the one in Lemma 4.3.2. □

Theorem 4.3.8. *If L is a countable dense linear order without endpoints and F_1, F_2 are as above, then $L(F_1, F_2)$ is computably categorical as a Heyting algebra.*

Proof. Let H_1 and H_2 be computable Heyting algebras which are isomorphic to $L(F_1, F_2)$. We will think of H_2 as being a computable "nice" copy of $L(F_1, F_2)$. This will mean that each element of H_2 is labeled as in Figure 4.1 and that, for each $a \in H_2$, we will know the $x \in L$ and $i \in \{0, 1\}$ such that $a \in F_{i,x}$.

Match up the least and greatest elements of H_1 and H_2. Use a back-and-forth argument to build the rest of the computable isomorphism.

Forward direction: Let n be the $\leq_{\mathbb{N}}$-least element of H_1 that is not in the domain of the isomorphism yet. Search in H_1 for a copy of the lattice F_1 that includes the point n. This copy of F_1 is either a component of the form $F_{1,x}$ or it sits inside a component of the form $F_{2,x}$. Use the $\rightarrow$ operation as in Lemma 4.3.5 to tell which situation is the case. If this copy of F_1 sits in an F_2 component, then search for the remaining elements of the component.

Since we will map components to components all at once, all of the points in this component are currently not in the domain of the isomorphism. Finitely much of the isomorphism has been determined so far, so there are finitely many copies of F_1 and F_2 (possibly none) to the left of this component and finitely many copies to the right which have already been mapped somewhere under the isomorphism.

From the structure of H_2, we know that there is a copy of the component in H_2 with the same number of copies of F_1 and F_2 to the left and to the right. Map the component to this copy (by matching up named elements).

Backward direction: This time we are given an element of H_2 which is not in the range yet. Since H_2 is "nice", we already know what lattice element n is and which component it is in (a copy of F_1 or F_2). Search for this component in H_1 as above: look for a copy of this component which has the same number of copies of F_1 and F_2 to the left and right in the isomorphism so far. The fact that H_1 is L-good means we know that if we find this copy, then the component embeds entirely in it. Use the $\rightarrow$ operation to determine whether this copy is F_1 or F_2. If the components match (they are both F_1 or both F_2), map the component containing n to its copy in H_1 by matching named elements. Otherwise, keep searching and checking for an appropriate copy in H_1 in which the components match up, and map the current component to its copy in H_1. □

Theorem 4.3.9. *There are computable distributive lattices which are classically isomorphic to a Heyting algebra but are not computably isomorphic to each other.*

Proof. Consider the Heyting algebra $L(F_1, F_2)$ from above, where L is a countable dense linear order without endpoints. We will build computable distributive lattices L_1 and L_2 such that $L_1 \cong L_2 \cong L(F_1, F_2)$, but L_1 and L_2 are not computably isomorphic. To achieve this, we will meet the following R_e requirements.

$$R_e \ : \qquad \varphi_e : L_1 \to L_2 \text{ is not an isomorphism}$$

Stage 0: Initially add 0_{L_i} and 1_{L_i} to be the least and greatest elements of L_i, respectively (for $i = 1, 2$). Then add one copy of F_1 and one copy of F_2 to L_1 with every element in the F_1 copy less than every element in the F_2 copy under the ordering $\leq_{L_1}$. Do the same for L_2.

Stage s+1: Suppose by induction that L_i (for $i = 1, 2$) consists of a finite sequence of alternating copies of F_1 and F_2. Between each pair of these components, add a new copy of F_1 and a new copy of F_2 so that they are still alternating. Also, if the current least component is a copy of F_1 (or F_2), add a new copy of F_2 (or F_1) and make that the new least component. Similarly, add a new greatest component.

Next, we will take care of the R_e requirements. Let R_e for $e < s$ be the least diagonalization requirement which is waiting to diagonalize or waiting to define its witness. If R_e is waiting to define its witness, assign R_e a copy of F_1 in L_1 which is not assigned to any other R_i requirement. Call this copy of L_1 the witness for R_e.

If R_e is waiting to diagonalize, check if $\varphi_{e,s}(n)$ converges for every n in the copy of F_1 which is the witness for R_e and that the images $\varphi_{e,s}(n)$ all lie in one copy of F_1 in L_2. (If this never happens, then R_e is trivially satisfied.) If this happens, change the image copy of F_1 to a copy of F_2 by adding the two extra points at the top as in Figure 4.1. Then we say that R_e is satisfied. Now there will be two copies of F_2 in a row in L_2, so add a new copy of F_1 between these two copies to keep the copies of F_1 and F_2 alternating in L_2.

A standard back-and-forth argument (without computability theory) can be used to show that L_1 and L_2 are isomorphic to $L(F_1, F_2)$, given that D_1 and D_2 are dense and partition L. Also, L_1 and L_2 are computable, since once the relationship between any two elements in L_i (for $i = 1, 2$) is defined at stage s, this relationship never changes in future stages $t > s$. In particular, diagonalizing by changing copies of F_1 into copies of F_2 does not change $\leq_{L_2}, \wedge_{L_2}$, or $\vee_{L_2}$, since we have already verified in Lemma 4.3.4 that F_1 embeds into F_2 in such a way that the ordering and meet and join operations are preserved. Therefore, L_1 and L_2 are computable, and $L_1 \cong L_2 \cong L(F_1, F_2)$. However, meeting the R_e requirements ensures that there is no computable isomorphism from L_1 to L_2. Thus, the distributive lattice $L(F_1, F_2)$ is not computably categorical. $\square$

Remark 4.3.10. *Theorem 4.3.9 would not work with* $H_1 \cong H_2 \cong L(F_1, F_2)$ *as Heyting algebras because changing a copy of* F_1 *into a copy of* F_2 *would change* $\rightarrow_{H_2}$ *as in Lemma 4.3.5.*

In searching for a characterization for the computably categorical Heyting algebras, three possible candidates are as follows.

(1) A Heyting algebra is computably categorical if and only if it has finitely many atoms.

(2) A Heyting algebra is computably categorical if and only if it has finitely many join-irreducible elements.

(3) A Heyting algebra is computably categorical if and only if its regular subalgebra as a Boolean algebra is computably categorical.

If (1) were true, then the characterization for Heyting algebras and Boolean algebras would be the same. However, it is possible to construct a computable Heyting algebra with finitely many atoms which is not computably categorical. As a simple example, consider the Heyting algebra H with least elements 0 and 1, two atoms x and y such that $x \wedge y = 0$ and $x \vee y = z$, and an $\mathbb{N}$-chain $n_0 < n_1 < n_2 < \ldots$ above z (see Figure 4.2). Let H' be a computable copy of H in which the successor relation $S(n_i, n_j)$ is not computable. Then there is no computable isomorphism from H' to H. If there were a computable isomorphism $\varphi : H' \to H$, then we could compute $S(n_i', n_j')$ for any $n_i', n_j' \in H'$ by computing $S(\varphi(n_i), \varphi(n_j))$, since the successor relation is computable in H (where $S(n_i, n_j)$ holds if and only if $j = i + 1$).

Furthermore, the free Heyting algebra on infinitely many generators has no

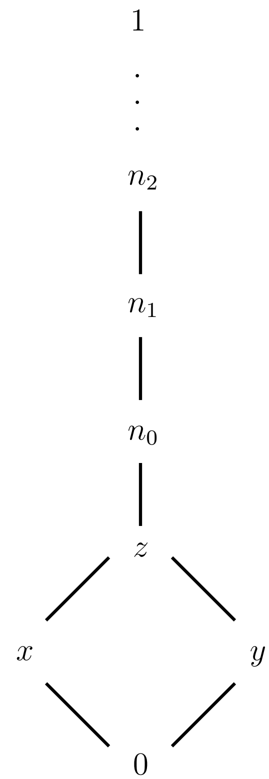


Fig. 4.2: Heyting algebra with an $\mathbb{N}$-chain above two atoms

atoms but is not computably categorical. In contrast, it is known that the free Boolean algebra on infinitely many generators (that is, the countable atomless Boolean algebra) is computably categorical. Therefore, it is not enough to look at the atoms of a Heyting algebra when studying computable dimension.

The possible characterization given by (2) is motivated by the fact that the join-irreducible elements are the same as the atoms in a Boolean algebra but not necessarily so in a Heyting algebra. One direction of (2) fails because the free Heyting algebra on finitely many generators is computably categorical but has infinitely many join-irreducible elements (cf. [13]).

The regular subalgebra of a Heyting algebra H is the set of regular elements $\mathrm{Reg}(H) = \{x : x^{**} = x\}$, and this always forms a Boolean algebra. However, the same example from Figure 4.2 can be used to show that one direction of (3) fails, since its regular subalgebra is the Boolean algebra $B = \{0, x, y, 1\}$.

Bibliography

[1] R. Balbes and P. Dwinger, *Distributive lattices*, University of Missouri Press, Columbia, Mo., 1974.

[2] G. Birkhoff, *Applications of lattice algebras*, Proc. Cambridge Philos. Soc., 30 (1934), pp. 115–122.

[3] D. Cenzer and J. B. Remmel, Π^0_1 *classes in mathematics*, in Handbook of recursive mathematics, Vol. 2, vol. 139 of Stud. Logic Found. Math., North-Holland, Amsterdam, 1998, pp. 623–821.

[4] Y. L. Ershov, S. S. Goncharov, A. Nerode, J. B. Remmel, and V. W. Marek, eds., *Handbook of recursive mathematics. Vol. 1*, vol. 138 of Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam, 1998. Recursive model theory.

[5] ——, eds., *Handbook of recursive mathematics. Vol. 2*, vol. 139 of Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam, 1998. Recursive algebra, analysis and combinatorics.

[6] S. S. Goncharov, *Limit equivalent constructivizations*, in Mathematical logic and the theory of algorithms, vol. 2 of Trudy Inst. Mat., "Nauka" Sibirsk. Otdel., Novosibirsk, 1982, pp. 4–12.

[7] G. Grätzer, *Lattice theory. First concepts and distributive lattices*, W. H. Freeman and Co., San Francisco, Calif., 1971.

[8] D. R. Hirschfeldt, B. Khoussainov, R. A. Shore, and A. M. Slinko, *Degree spectra and computable dimensions in algebraic structures*, Ann. Pure Appl. Logic, 115 (2002), pp. 71–113.

[9] C. G. Jockusch, Jr. and R. I. Soare, Π^0_1 *classes and degrees of theories*, Trans. Amer. Math. Soc., 173 (1972), pp. 33–56.

[10] J. F. Knight, *Degrees coded in jumps of orderings*, J. Symbolic Logic, 51 (1986), pp. 1034–1042.

[11] A. Nerode and R. A. Shore, *Logic for applications*, Graduate Texts in Computer Science, Springer-Verlag, New York, second ed., 1997.

[12] I. Nishimura, *On formulas of one variable in intuitionistic propositional calculus.*, J. Symbolic Logic, 25 (1960), pp. 327–331 (1962).

[13] M. O'Connor, *Embeddings into free Heyting algebras and translations into intuitionistic propositional logic*, in Logical foundations of computer science, vol. 4514 of Lecture Notes in Comput. Sci., Springer, Berlin, 2007, pp. 437–448.

[14] J. B. Remmel, *Recursive isomorphism types of recursive Boolean algebras*, J. Symbolic Logic, 46 (1981), pp. 572–594.

[15] ——, *Recursively categorical linear orderings*, Proc. Amer. Math. Soc., 83 (1981), pp. 387–391.

[16] L. J. Richter, *Degrees of structures*, J. Symbolic Logic, 46 (1981), pp. 723–731.

[17] V. L. Selivanov, *Algorithmic complexity of algebraic systems*, Mat. Zametki, 44 (1988), pp. 823–832, 863.

[18] R. I. Soare, *Recursively enumerable sets and degrees*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.

[19] D. van Dalen, *Logic and structure*, Universitext, Springer-Verlag, Berlin, fourth ed., 2004.