REVIEW OF INTRODUCTION TO LATTICES AND ORDER (B.A. DAVEY AND H.A. PRIESTLEY)

REVIEWED BY: JON COHEN

1. Introduction

The idea that a set may come equipped with a natural ordering on its elements is so basic as to pass by unnoticed by most. However, this belies a wonderful opportunity for unifying the study of many disparate examples of these creatures. Historically, the study of order has led to a great unification of results in algebra and logic. More recently, it has infused into theoretical computer science, particularly into programming language semantics.

Given two elements, x and y, in some partially ordered set, or poset, P, it is often the case that they have both a least upper bound, $\text{lub}\{x, y\}$, and a greatest lower bound, $\text{glb}\{x, y\}$. An example of such a poset is provided by the natural numbers under their usual order. However, this property certainly does not hold for every poset. For instance, suppose that the underlying set of P consists of those finite subsets of $\mathbb{N}$ that have even cardinality. This becomes a poset by saying that x is less than or equal to y just in case x is a subset of y. Now, suppose that $x = \{1, 2\}$ and $y = \{2, 3\}$. Any upper bound of both x and y has to contain $x \cup y = \{1, 2, 3\}$. Necessarily, a least upper bound would have cardinality 4. However, there is no *least* such set containing $\{1, 2, 3\}$.

If we are in the happy situation that any two elements of our poset P have both a least upper bound and a greatest lower bound, then we say that P is a *lattice*. If P is a lattice and $x, y \in P$, then let us save on effort by writing $\text{glb}\{x, y\}$ as $x \vee y$ and $\text{lub}\{x, y\}$ as $x \wedge y$. The operator $\vee$ is called *join* and the operator $\wedge$ is called *meet*. Playing around with these operators a bit, one may notice some identities starting to crop up. Amongst those identities that hold for elements $a, b, c \in P$, we find the following:

$$\begin{array}{ll} a \wedge a = a & a \vee a = a \\ a \wedge b = b \wedge a & a \vee b = b \vee a \\ a \wedge (b \wedge c) = (a \wedge b) \wedge c & a \vee (b \vee c) = (a \vee b) \vee c \\ a \wedge (a \vee b) = a & a \vee (a \wedge b) = a \end{array}$$

In fact, the above equations are sufficient to define P by making the observation that $a \leq b$ if and only if $a = a \wedge b$. So, we have two ways of looking at a lattice. That is, we can see it as either a special sort of poset or as a set with two binary operators satisfying the above equalities.

This brings us to the book under review, which sets out to provide a thorough introduction to both points of view. Examples of lattices abound in theoretical computer science and, indeed, the book makes no bones about dipping into computational applications on a regular basis. Let's see what's between the covers, shall we?

2. What's Inside?

Chapter 1: Ordered Sets. At a very abstract level, one may define a deterministic program to be a function from a set of input states to a set of output states. Of course, one only really wants to consider those functions that are computable, but let us sweep this concern under the rug. Now, it may very well happen that a given program does not terminate when presented with certain inputs. We can build this possibility into our model by only requiring a program to correspond to a *partial* map.

For sets I and O, let us use $(I \Rightarrow O)$ to denote the collection of all partial functions from I to O. This set carries a partial ordering in a natural way. To wit, take two partial functions $f, g \in (I \Rightarrow O)$. The partial ordering arises by declaring that $f \sqsubseteq g$ if and only if $\mathrm{dom}(f) \subseteq \mathrm{dom}(g)$ and $f(x) = g(x)$ for every $x \in \mathrm{dom}(f)$. Viewing f and g as deterministic programs, we have $f \sqsubseteq g$ if and only if whenever f terminates from some input state, so does g and in the same output state as f. Additionally, g may terminate from some input states that f does not.

The above is one of many examples of ordered sets that kick off this chapter and crops up repeatedly throughout the book when developing aspects of the semantics of programming languages.

Following on from the examples, several important constructions on posets are introduced. Of these, we only mention two that are of particular importance for lattices.

Let P be a poset. We say that P has a *bottom* if there is some $\bot \in P$ such that $\bot \leq x$ for all $x \in P$. Dually, P has a top if there is some $\top \in P$ such that $x \leq \top$ for all $x \in P$. A poset with both a top and a bottom is said to be *bounded*. In the deterministic program setting, $\bot$ corresponds to the program that fails on every input and $\top$ corresponds to an ill defined program. At this point, I cannot resist quoting the authors' view on this situation:

> Accordingly, computer scientists often choose models which have bottoms, but prefer them topless.

Suppose now that we are handed a subset $Q \subseteq P$. The *down set* of Q is the set $\downarrow Q$ consisting of all those elements "below" Q. Specifically, $\downarrow Q := \{x \in P \mid (\exists y \in Q)\ x \leq y\}$.The collection of all down sets of P is denoted by $\mathcal{O}(P)$ and itself forms a poset under the set inclusion ordering. This poset is very important for the representation theory of lattices developed later in the book.

Chapter 2: Lattices and Complete Lattices. In the definition of lattices given in the introduction to this review, it was only required that the join and meet of any *two* elements exists. By induction, we can form the join and meet of any finite number of elements of a lattice. But what if the lattice contains infinitely many elements? Can we always form the join and meet of an arbitrary subset? Sadly, the answer is no. A counterexample is provided by the rational numbers under their usual ordering.

A lattice, L, is called *complete* if the join and meet of any subset exists. Every complete lattice is necessarily bounded by taking $\bot = \bigwedge L$ and $\top = \bigvee L$. By the above remarks, every finite lattice is complete. An example of an infinite complete lattice is provided by the powerset of any infinite set, ordered by inclusion. An important observation is that, for any lattice L, the poset $\mathcal{O}(L)$ forms a complete lattice by taking $\wedge := \cap$ and $\vee := \cup$.

After introducing lattices and complete lattices, the chapter goes on to develop some of the basic constructions on lattices. These include homomorphisms, sublattices and products of lattices. Following this, some basic results concerning complete lattices are derived. The most important result along the way for computational purposes is the Knaster-Tarski Fixpoint Theorem. Briefly, if L and K are posets, a map $f : L \to K$ is called *order preserving* if whenever $x \leq_L y$ in L, we have that $f(x) \leq_K f(y)$ in K. The theorem states that if L is a complete lattice, then any order preserving map, f, from L to itself has both a greatest and a least fixpoint, where a fixpoint is defined to be an $x \in L$ such that $F(x) = x$.

Chapter 3: Formal Concept Analysis. One way to describe an abstract object is by the collection of properties it satisfies. If we agree on a collection of properties of interest, then this provides the basis for a rough comparison of equivalence of objects. This is roughly the starting observation for the young field of Formal Concept Analysis. Here, "object" is usually taken to be an arbitrary physical object, but one could equally well consider objects in some object oriented programming language. Getting slightly more technical, we say that a *context* is a triple (O, A, I), where O is a set of *objects*, A is a set of *attributes* and $I \subseteq O \times A$. Given some $o \in O$, we can look at the set, o', of all attributes that it satisfies. Similarly, we can look at the set, a', of objects that possess a given attribute $a \in A$. There is nothing really stopping us from generalising this to

arbitrary subsets $X \subseteq O$ and $Y \subseteq A$ in the following way:

$$X' := \{a \in A \mid (\forall x \in X)\ xIa\}$$
$$Y' := \{o \in O \mid (\forall y \in Y)\ oIy\}$$

Given some context (O, A, I), a *concept* is a pair (X, Y), where $X \subseteq O$, $Y \subseteq A$, $X' = Y$ and $Y' = X$. The collection of all concepts, denoted $\mathcal{B}(O, A, I)$, can be made into a poset by defining

$$(X_1, Y_1) \leq (X_2, Y_2) \iff X_1 \subseteq Y_1 \iff Y_2 \subseteq X_2.$$

In fact, $\mathcal{B}(O, A, I)$ becomes a complete lattice under this ordering.

The main results of this chapter obtain the completeness of $\mathcal{B}(O, A, I)$ and show, essentially, that any complete lattice L is isomorphic to a concept lattice, namely to $\mathcal{B}(L, L, \leq)$. In the special case where we take L to be $\mathbb{R}$ with its usual ordering, this result says that $\mathbb{R}$ is isomorphic, as a complete lattice, to the set of Dedekind Cuts. This is no accident, a proper explanation comes later in the book when completions are discussed.

Chapter 4: Modular, distributive and Boolean lattices. Let L be a lattice. We say that L is *modular* if for any $a, b, c \in L$ such that $c \leq a$, we have $a \wedge (b \vee c) = (a \wedge b) \vee c$. The seemingly arbitrary name for this property comes from the fact that it is satisfied by modules over a ring, where $\wedge$ is taken to be set intersection and $\vee$ is taken to be direct sum. Many other algebraic examples of lattices, such as the lattice of normal subgroups of a group or the lattice of subspaces of a vector space, form modular lattices.

A stronger property to impose on L is to require that it be *distributive*. That is, for any $a, b, c \in L$, we have $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$. Necessarily, any distributive lattice is modular, but the converse need not hold.

If L is a bounded lattice, then one may wonder whether there is any significant relation between the lattice operations and $\top$ and $\bot$. Given $a \in L$, we say that $b \in L$ is a *complement* of a if $a \wedge b = \bot$ and $a \vee b = \top$. If L happens to be distributive then complements, if they exist, are necessarily unique. Armed with these definitions, we may now call a lattice *Boolean* if it is bounded, distributive and every element has a complement. Of course, this name arises from the fact that there is effectively no difference between them and Boolean algebras.

After defining modular and distributive lattices, the chapter goes on to give several examples of each before discussing some methods for establishing that a given lattice is *not* modular or distributive. Following that, Boolean lattices and Boolean algebras are introduced and several basic results on truth tables, normal forms and digital circuits are derived.

Chapter 5: Representation: the Finite Case. Lattices were defined in such a way that we can see them equally as posets or sets with special binary operators. Given this correspondence, can we find classes of ordered sets corresponding to the lattices introduced in Chapter 4, which were defined only in terms of meet and join? In many cases, it turns out that we can. In the finite case, things work out particularly nicely.

Let L be a lattice and let $x \in L$. We say that x is *join irreducible* if whenever $x = a \vee b$ for some $a, b \in L$, either $x = a$ or $x = b$. If L has a bottom, then we further require that $x \neq \bot$. The set of all join irreducible elements of L is denoted by $\mathcal{J}(L)$.

After a brief introduction to the general strategy behind obtaining representation theorems, the representation theorem for finite Boolean lattices is proved. The most important result stemming from this representation is that every Boolean lattice is isomorphic to $\mathbf{2}^n$ for some $n \geq 0$, where $\mathbf{2}$ is the unique Boolean lattice on a two element set.

Of much greater interest for later chapters is the representation theorem given for finite distributive lattices. Here, it turns out that any finite distributive lattice L is isomorphic to the poset $\mathcal{O}(\mathcal{J}(L))$. As a corollary, it follows that every finite distributive lattice is isomorphic to a sublattice of $\mathbf{2}^n$ for some $n \geq 0$. The representation theorem for finite distributive lattices is deceptively straightforward. In essence, it goes through by the fact that any finite lattice is complete, since we

saw previously that $\mathcal{O}(L)$ is complete for *any* lattice L. Since infinite lattices need not be complete, significantly more machinery needs to be developed before considering *infinite* distributive lattices in a later chapter.

Chapter 6: Congruences. This is the shortest chapter of the book and provides a glimpse of one of the ways that lattices arise in modern algebra. It can be safely skipped without losing the continuity of the book.

Roughly speaking, a congruence is an equivalence relation on an algebra that is "compatible with the operations". Congruences correspond, for instance, to normal subgroups of a group and ideals of a ring. Given a lattice L, an equivalence relation $\equiv$ on L is called a *congruence* if whenever $a \equiv b$ and $c \equiv d$ then $a \vee c \equiv b \vee d$ and $a \wedge c \equiv b \wedge d$, where $a, b, c, d \in L$.

After defining congruences on lattices and characterising the special properties of the blocks of the underlying equivalence relation, the chapter goes on to show that the set of all congruences on a lattice itself forms a distributive lattice.

Chapter 7: Complete Lattices and Galois Connections. The collection of all subspaces of a vector space, V, can be given a lattice structure by first defining meet to be set intersection. But how are we to define join? Clearly we cannot just define it to be set union, since the union of two subspaces of V need not be a subspace. The trick is to define, for subspaces S_1 and S_2 of V, their join to be $\mathrm{Span}(S_1 \cup S_2)$.

The above example works because Span() is a *closure operator*. For a poset P, a map $c : P \to P$ is called a *closure operator* if, for any $x, y \in P$, we have:

$$x \leq C(x), \quad C \leq y \Rightarrow C(x) \leq C(y), \quad C(C(x)) = C(x).$$

The first half of the chapter is concerned with a certain class of closure operators, which are called "algebraic". As the name suggests, these are of great importance in modern algebra. In a nutshell, algebraic closure operators are intimately related to "algebraic lattices" and it can be shown that every algebraic lattice is isomorphic to the lattice of subalgebras of some abstract algebra. This last result is certainly beyond the scope of the book under review. Nevertheless, some of the most important features of algebraic lattices and algebraic closure operators are developed, hinting at the development of domain theory later in the book.

The second half of the chapter develops some aspects of the theory of Galois connections, which were hinted at in Chapter 3. Let P and Q be posets. Suppose that $f : P \to Q$ and $g : Q \to P$ are order preserving. Then, we say that f and g form a *Galois connection* if for any $p \in P$ and $q \in Q$, we have $f(p) \leq q \iff p \leq g(q)$. Given that f and g form a Galois connected pair, it is not hard to show that both $f \circ g$ and $g \circ f$ form closure operators. Conversely, one can show that any closure operator arises in this way.

The chapter concludes with a section on completions. Showcased is the Dedekind-Macneille completion, which is a generalisation of the Dedekind Cut method of completing the rationals to an arbitrary poset.

Chapter 8: CPOs and Fixpoint Theorems. After an extended theoretical development, the book now begins to edge back towards computational applications.

Suppose that P is a poset and that $D \subseteq P$. We say that D is *directed* if for any $x, y \in D$, the pair $\{x, y\}$ has an upper bound $z \in D$. Now, consider a map $f : \mathbb{N} \to \mathbb{N}$. If $g \in (N \Rightarrow N)$ is a partial map having finite domain such that $g \sqsubseteq f$, then we may regard g as providing a finite approximation to f. The collection of all such partial maps forms a directed subset of $(N \Rightarrow N)$ whose join is f.

Motivated by the above example, we say that a poset P is a *Complete Partial Order*, or CPO, if the join of any directed subset of P exists. Of course, any complete lattice is also a CPO. An example of a CPO that is not a complete lattice is provided by the set of all binary strings under the prefix order.

After defining CPOs, the chapter goes on to define some basic constructions on CPOs. The heart of the chapter follows this and develops three fundamental least fixpoint theorems for CPOs in the spirit of the Knaster-Tarski fixpoint theorem for complete lattices, which are put to good use in the following chapter. The chapter concludes with some useful results on computing least fixpoints for compositions of Galois connected order preserving maps on complete lattices.

Chapter 9: Domains and Information Systems. This is the computational climax of the book. Suppose that P is a CPO and let $p \in P$. we say that p is *finite* if for every directed subset $D \subseteq P$, if $k \leq \bigvee D$, then $k \leq d$ for some $d \in D$. The collection of all finite elements of a CPO P is denoted $F(P)$. For the CPO consisting of all binary strings under the prefix order, the finite elements are precisely the finite strings.

If the meet of any nonempty subset of a CPO P exists, then we say that P is a *complete semilattice*. A *domain* is a complete semilattice P, such that for every $p \in P$, we have $p = \bigvee(\downarrow \{p\} \cap F(P))$. It can be shown that $\downarrow\{p\} \cap F(P)$ is necessarily a directed set. Examples of domains include $(S \Rightarrow S)$ for any $S \subset \mathbb{R}$, as well as the collection of all binary strings.

After introducing domains, the chapter goes on to define the notion of an information system. Roughly, an information system is a triple $\langle A, Con, \vdash \rangle$ such that A is a set of "tokens", Con is a nonempty set of finite subsets of A and $\vdash$ is an "entailment relation" between members of Con and members of A. The set Con and the relation $\vdash$ are subject to some constraints.

One of the main results of the chapter is to show how to set up a bijective correspondence between domains and informations systems, so that results about one may be transferred to the other. Following this, a very brief introduction to denotational semantics and the role that domains play in them is provided.

Chapter 10: Maximality Principles. This chapter mainly collects together a number of tools that will be needed in the final chapter. Several equivalents to the Axiom of Choice are introduced and their implications for distributive lattices and Boolean algebras are discussed. The most important consequences for the following chapter are that every distributive lattice is isomorphic to a lattice of sets and that this embedding carries over to Boolean algebras.

Chapter 11: Representation: the General Case. This is the mathematical climax of the book. Indeed, the mathematical sophistication of the material jumps up a notch, relying on some point set topology, which is briefly introduced in an appendix. The main goal is to provide a concrete description of the images of the embeddings of distributive lattices and Boolean algebras introduced in the last chapter. The first stop is Stone's Representation Theorem, which shows that every Boolean algebra is isomorphic to the Boolean algebra of clopen subsets of some compact totally disconnected topological space. After a very brief foray back into logic, the chapter returns to the question of what the image of a bounded distributive lattice looks like under the embedding. This is the content of Priestley's Representation Theorem. The main problem is that, since we no longer have complements as in the Boolean case, we need to find a way to exploit the underlying order more. A *Priestley Space* is a set, X, which caries both a partial order and a topology such that the topology is *totally order-disconnected*. What this means is that for any $x, y \in X$ such that $y \not\leq x$, there is a clopen down set U such that $x \in U$ and $y \notin U$. An example of a Priestley Space is provided by the Cantor Set, with the order inherited from the interval $[0, 1]$. Priestley's Representation Theorem says that every bounded distributive lattice is isomorphic to the lattice of clopen down sets of some Priestley Space. Following on from the proof of this very important theorem, the chapter concludes with a discussion of duality.

3. Opinion

The book is written in a very engaging and fluid style. The understanding of the content is aided tremendously by the very large number of beautiful lattice diagrams. TEXnicians may be interested to note that there are descriptions of the algorithms used for drawing various classes of lattices dotted throughout the book.

An interesting pedagogical development is the inclusion of "Exercises from the text" at the end of many chapters. These ask that the reader fill in the details of proofs, substantiate claims made in the chapter etc. In short, it points out things that the reader should have been checking as she read the chapter. This is an excellent idea for an introductory text such as this, as it helps students make the transition to reading monographs and research papers. In addition, there are copious exercises, most of which extend the theory and give a glimpse of more advanced topics. As such, the book is particularly well suited to self study.

While there are technically no prerequisites for the book, as everything is developed from scratch, some prior exposure to abstract algebra would be very useful. In particular, a reader without such a background is likely to find the final few chapters very heavy going.

There are at least two qualitatively different courses that could be taught from this book. For a very computationally oriented course, one could use Chapters 1,2,4,7,8 and 9 as a basis for an introduction to programming language semantics, possibly supplemented with some extra references on domain theory and denotational semantics. For a mathematically oriented course, one could use Chapters 1,2,4,5,7,10 and 11 as an introduction to lattice theory, perhaps supplemented with some additional references on universal algebra.

One minor quibble is that the decision to split the discussion of complete lattices into two chapters, which appear far apart from each other in the book, is slightly strange. While it is clear that the authors' intention is to present the minimal amount of material necessary to introduce a particular application, this does place a strain on some of the discussion. In particular, the constructions in Chapter 3 would be more perspicuous had some material on abstract Galois connections already been presented.

While the authors note in the preface that coverage of universal algebra and category theory is beyond the scope of the book, the inclusion of an entire chapter on congruences seems rather arbitrary. The space used to describe parochial properties of congruences may well have been better used providing an introduction to basic categorical language, which would have helped to unify much of the discussion of duality.

These concerns aside, the book provides a wonderful and accessible introduction to lattice theory, of equal interest to both computer scientists and mathematicians.