Lattice Representations for Computability Theory

Peter A. Fejer*
University of Massachusetts at Boston

October 3, 1997

Introduction

Lattice embeddings are a common tool employed by computability theorists to study the structures they are interested in. They are used both to show the undecidability of the full theory of a structure and to decide fragments of the structure's theory. Since every countable distributive lattice can be embedded into the countable atomless Boolean algebra, and this Boolean algebra has easily constructed computable representations, there is no problem of representation if one wishes to embed all countable distributive lattices into a structure. However, if one wishes to embed all finite or all computably presentable lattices into a structure, some sort of representation for these lattices is needed.

Not surprisingly, representations for lattices have concerned lattice theorists since the early days of their subject. They have looked at representations of the elements of an arbitrary lattice (up to isomorphism) by various sorts of objects, such as subgroups of a fixed group or geometries on a fixed set, but they have recognized as most fundamental, representations of an arbitrary lattice as a lattice of equivalence relations on a fixed set. It is exactly this sort of representation that computability theorists have used to create the representations they need to carry out their embedding theorems. The first use of a type of lattice representation derived from a representation of a lattice by equivalence relations to prove a computability-theoretic embedding theorem is due to Thomason in [16]. Since this paper, practitioners in the field have recognized the connection between the type of representation they use and the type of representation developed by lattice theorists, but this connection has not received a thorough discussion in print.

*The material in this paper is an expansion of talks the author gave while a Guest Professor at Heidelberg University. The author thanks Klaus Ambos-Spies and Gert Müller for their hospitality. He also thanks Bjarni Jónsson for his kind answers to requests for information and his pointers to the literature.

In many papers in computability theory that use lattice representations, it is desirable to assume that the representations have certain additional properties that do not follow immediately from the underlying lattice-theoretic representations. In some cases, (most notably the homogeneity assumptions used in initial segments results) these desired additional properties take quite some work to obtain. In other cases, the additional properties are more easily obtained, and papers say that these properties can be assumed without loss of generality, but do not give any justification. Although it is folklore that these additional properties can in fact be obtained, no easily accessible justification exists in print.

In this expository paper, we present the basic definitions and results about lattice representations needed by computability theorists. The plan of the paper is as follows. We first define lattice representations both from the lattice- theoretic and computability-theoretic points of view, give examples and show the connection between the two types of representations. Then, we discuss some of the known theorems on the existence of lattice representations that are of interest to computability theorists. Next, we discuss some additional properties of representations that computability theorists have found useful and show how they can be obtained. Finally, we give a simple example of the use of lattice representations in an embedding result.

Definition of Representation

For $n \in \omega$, we write I_n for $\{0, \ldots, n-1\}$ and we write I_ω for ω. If $\mathcal{P}$ is a partial order, then we assume that $\mathcal{P} = (P, \leq_{\mathcal{P}})$. We will call a lattice $\mathcal{L}$ *computably presented* if there is an enumeration without repetitions $\{p_i | i \in I_m\}$ of L with $m \in \omega^+$ such that both the sets $\{(i,j,k) | p_i \vee_{\mathcal{L}} p_j = p_k\}$ and $\{(i,j,k) | p_i \wedge_{\mathcal{L}} p_j = p_k\}$ are computable.

Definition 1 *For any set A, $Eq(A)$ denotes the set of all equivalence relations over A.*

Theorem 2 *For any set A, $(Eq(A), \subseteq)$ is a lattice.*

Proof: Inclusion is obviously a partial order. The intersection of any nonempty family of equivalence relations over A is again an equivalence relation over A and $A \times A$ is the greatest element of $(\mathrm{Eq}(A), \subseteq)$, so this partial order is a complete lower semilattice with greatest element, so is a lattice, with meet given by intersection. ∎

We can also give an explicit description of the join operation in $\mathrm{Eq}(A)$. Let E_1 and E_2 be equivalence relations over A. Then $(E_1 \cup E_2)^+$, the transitive closure of $E_1 \cup E_2$, is an equivalence relation on A ($(E_1 \cup E_2)^+$ is obviously reflexive and transitive; since $E_1 \cup E_2$ is symmetric, $((E_1 \cup E_2)^+)^{-1}$ is a transitive relation containing $E_1 \cup E_2$, so, by definition of transitive closure, $(E_1 \cup E_2)^+ \subseteq ((E_1 \cup E_2)^+)^{-1}$, which implies that $(E_1 \cup E_2)^+$ is symmetric) and is certainly contained in any equivalence relation on A which contains both E_1 and E_2, so it is the join of E_1 and E_2 in $\mathrm{Eq}(A)$. From general principles about the transitive closure, it follows that for each $a, b \in A$, $(a, b) \in (E_1 \cup E_2)^+$ if and only if there is an $n \geq 1$ and a sequence $x_0, \ldots, x_n$ of elements of A such that

- $x_0 = a$,
- $x_n = b$
- for each $i < n$, $x_i(E_1 \cup E_2)x_{i+1}$.

Because E_1 is transitive and reflexive and E_2 is transitive, the last condition can be replaced by

- for each $i < n$, if i is even then $x_i E_1 x_{i+1}$ and if i is odd then $x_i E_2 x_{i+1}$.

To say the same thing in yet another way, if E_1 and E_2 are equivalence relations, then

$$(E_1 \cup E_2)^+ = E_1 \cup E_1E_2 \cup E_1E_2E_1 \cup \cdots$$

where we denote the usual relation product by juxtaposition. We will denote the $n+1$st term in the union given above as $P_n(E_1, E_2)$ (so $P_0(E_1, E_2) = E_1$, $P_1(E_1, E_2) = E_1E_2$, etc.). If E_1 and E_2 are equivalence relations and $n \leq m$, then $P_n(E_1, E_2) \subseteq P_m(E_1, E_2)$.

Definition 3 *Let $\mathcal{L}$ be a lattice and let A be a set. A* lattice-theoretic A-representation for *$\mathcal{L}$ is a lattice monomorphism (i.e., a meet and join preserving injection) $\eta : L \to Eq(A)$. A* lattice-theoretic representation *for $\mathcal{L}$ is a function which is a lattice-theoretic A-representation for $\mathcal{L}$ for some A. A lattice-theoretic A-representation for $\mathcal{L}$ is*

- finite *if A is finite,*
- computable *if there are enumerations without repetitions $L = \{p_i | i \in I_n\}$ and $A = \{a_i | i \in I_m\}$ with $n, m \in \omega^+$ such that $\{\langle i, j, k\rangle | a_j \eta(p_i) a_k\}$ is computable, and*
- type m *(for $m \in \omega$) if for all $p, q \in L$, $(\eta(p) \cup \eta(q))^+ = P_m(\eta(p), \eta(q))$, where $P_m(\eta(p), \eta(q))$ is the $m+1$-factor product of $\eta(p)$ and $\eta(q)$ defined above.*

(As mentioned previously, lattice theorists have studied other types of representations for lattices, but since they have recognized the type given above as being most fundamental, it is not too much of an abuse to call these "lattice-theoretic" representations.) The study of the lattices $(\mathrm{Eq}(A), \subseteq)$ and the idea of representing an arbitrary lattice as a lattice of equivalence relations began with the paper [1] of Birkhoff.

We now turn to the computability-theoretic view of lattice representations.

Definition 4 *Let $\mathcal{L}$ be a lattice and let A be a set. A* computability-theoretic A-representation for $\mathcal{L}$ *is an A- indexed collection $\Theta = \{\theta_a | a \in A\}$ of functions each with domain L such that for all $p, q, r \in L$:*

$$p \leq_{\mathcal{L}} q \leftrightarrow (\forall \alpha, \beta \in \Theta)(\alpha(q) = \beta(q) \rightarrow \alpha(p) = \beta(p)). \tag{1}$$

$$p \vee_L q = r \leftrightarrow (\forall \alpha, \beta \in \Theta)((\alpha(p) = \beta(p) \& \alpha(q) = \beta(q)) \leftrightarrow \alpha(r) = \beta(r)). \tag{2}$$

$$\begin{aligned} p \wedge_L q = r \leftrightarrow & (\forall \alpha, \beta \in \Theta)(\alpha(r) = \beta(r) \leftrightarrow \\ & (\exists n)(\exists \gamma_0, \ldots, \gamma_{n+1} \in \Theta)(\gamma_0 = \alpha \& \gamma_{n+1} = \beta \& \\ & (\forall i \leq n)((i \text{ even } \rightarrow \gamma_i(p) = \gamma_{i+1}(p)) \& (i \text{ odd } \rightarrow \gamma_i(q) = \gamma_{i+1}(q))))). \end{aligned} \tag{3}$$

A sequence $(\gamma_0, \ldots, \gamma_{n+1})$ as in (3) is called a sequence of (p, q)-interpolants *between α and β. A* computability-theoretic representation for $\mathcal{L}$ *is a collection of functions which is a computability-theoretic A-representation for $\mathcal{L}$ for some A. A computability-theoretic A-representation for $\mathcal{L}$ is*

- finite *if A is finite,*
- computable *if there are enumerations without repetitions $L = \{p_i | i \in I_n\}$ and $A = \{a_i | i \in I_m\}$ with $n, m \in \omega^+$ such that $\{\langle i, j, k\rangle | \theta_{a_j}(p_i) = \theta_{a_k}(p_i)\}$ is computable, and*
- type m *(for $m \in \omega$) if the following condition is met for all $p, q \in L$:*

  $(\forall \alpha, \beta \in \Theta)$ (if there is any sequence of (p, q)-interpolants between α and β, then there is a sequence of (p, q)-interpolants between α and β of length $m + 2$) (4_m)

The first published use of lattice representations similar to the ones defined in Definition 4 is in Lerman's paper [9]. (Around the same time, Shoenfield used a similar type of representation in an unpublished manuscript.) Although similar sorts of representations were used by Thomason in [16] and Lerman in [10], the first explicit definition of computability-theoretic lattice representation, essentially the same as ours, was given by Lerman in [11], where representations are called lattice tables.

Note that a finite representation is computable and is type m for some m.

Our definition of computability-theoretic representation is somewhat broader than what a computability theorist means in practice by a representation. Namely, one usually assumes that all of the functions in the representation take their values in ω and that the index set A is an initial segment of ω. We will see later why this can be assumed.

The conditions given in (1), (2) and (3) do not depend on the index set A at all. This suggests that one might define a representation simply to be a collection of functions Θ satisfying the given conditions. However, the index set is important in practice, and making it explicit will help to explain the connection between the lattice-theoretic and computability-theoretic viewpoints.

In our definition of computability-theoretic representation of a lattice, we do not specify where the functions in the representation take their values, as this makes for a smoother theory. To a large extent, what matters about a computability-theoretic A-representation is not where it takes its values but rather, for each a, b in A and p in the lattice, whether the functions corresponding to a and b have the same value on p. This is made precise in the following definition.

Definition 5 *Let A be a set, $\mathcal{L}$ be a lattice and $\Theta = \{\theta_a | a \in A\}$, $\Theta' = \{\theta'_a | a \in A\}$ be two families of functions with domain L. We call Θ and Θ'* equivalent *if for all $a, b \in A$ and $p \in L$ we have*

$$\theta_a(p) = \theta_b(p) \leftrightarrow \theta'_a(p) = \theta'_b(p).$$

It is immediate that if Θ and Θ' as above are equivalent then Θ is a (computable) representation for $\mathcal{L}$ if and only if Θ' is a (computable) representation for $\mathcal{L}$.

Equivalent Definitions of Representation

The conditions given in Definition 4 are in a strong form, which is useful if one has a representation and wants to use it for a lattice-embedding construction. As the next theorem shows, the individual conditions can be weakened or eliminated in various ways with the resulting set of conditions equivalent to the ones of the definition. Such weakened sets of conditions are useful for showing that a given collection of functions is in fact a representation of a given lattice.

Theorem 6 *Let $\mathcal{L}$ be a lattice, A be a set and $\Theta = \{\theta_a | a \in A\}$ be a collection of functions with domain L. For $p, q, r \in L$, let (2′) and (3′) be the following conditions:*

$$p \vee_L q = r \rightarrow (\forall \alpha, \beta \in \Theta)((\alpha(p) = \beta(p) \& \alpha(q) = \beta(q)) \rightarrow \alpha(r) = \beta(r)). \tag{2'}$$

$$\begin{aligned} p \wedge_L q = r \rightarrow {} & (\forall \alpha, \beta \in \Theta)(\alpha(r) = \beta(r) \rightarrow \\ & (\exists n)(\exists \gamma_0, \ldots, \gamma_{n+1} \in \Theta)(\gamma_0 = \alpha \& \gamma_{n+1} = \beta \& \\ & (\forall i \leq n)((i \text{ even } \rightarrow \gamma_i(p) = \gamma_{i+1}(p)) \& (i \text{ odd } \rightarrow \gamma_i(q) = \gamma_{i+1}(q))))). \end{aligned} \tag{3'}$$

Then Θ is a computability-theoretic A-representation for $\mathcal{L}$ if and only if Θ meets one of the following sets of conditions for all $p, q, r \in L$:

1. *(1), $(2')$, and $(3')$.*

2. *(2) and $(3')$.*

3. *$(2')$ and (3).*

Proof: We will show that condition (2) holds for all $p, q, r \in L$ if and only if conditions (1) and $(2')$ hold for all $p, q, r \in L$ and that condition (3) holds for all $p, q, r \in L$ if and only if conditions (1) and $(3')$ hold for all $p, q, r \in L$. This will clearly imply the result.

It is obvious that condition (2) implies condition $(2')$. Setting r to be q in (2) gives a statement that is easily seen to be equivalent to (1). Thus, (2) implies (1) and $(2')$. Conversely, suppose that (1) and $(2')$ hold for all $p, q, r \in L$. Let p, q, r be in L. We show that (2) holds. Suppose first that $p \vee_L q = r$. If α and β are in Θ and they agree on both p and q then, by $(2')$, they agree on r and, if they agree on r, then, by (1), since $p, q \leq_{\mathcal{L}} r$, they agree on p and q. Thus, we get the left to right implication of (2). Now, suppose that any two elements of Θ agree on r if and only if they agree on p and q. Then, by (1), r is an upper bound for $\{p, q\}$. Suppose that t is an upper bound for $\{p, q\}$. Then, if α and β are two elements of Θ which agree on t, then, by (1), they agree on both p and q and hence, by hypothesis, they agree on r. Thus, by (1), $r \leq_{\mathcal{L}} t$, so $r = p \vee_L q$, as desired.

Now suppose that condition (3) holds. Then condition $(3')$ obviously holds. To see that (1) holds, we argue as follows. Suppose first that $p \leq_{\mathcal{L}} q$ and that $\alpha(q) = \beta(q)$ for $\alpha, \beta \in \Theta$. Then, since $q \wedge_L p = p$ and (α, β) is a sequence of (q, p)-interpolants from α to β, (3) implies that $\alpha(p) = \beta(p)$. This establishes the left to right direction of (1). Now suppose that any two elements of Θ that agree on q also agree on p. Let α, β be in Θ. If $\alpha(p) = \beta(p)$, then (α, β) is a sequence of (p, q)-interpolants from α to β. Conversely, if $(\gamma_0, \ldots, \gamma_{n+1})$ is a sequence of (p, q)-interpolants from α to β, then $\gamma_i(p) = \gamma_{i+1}(p)$ for all $i \leq n$ (because any two elements of Θ that agree on q also agree on p), so $\alpha(p) = \beta(p)$. Thus, two elements of Θ agree on p if and only if there is a sequence of (p, q)-interpolants between them. It follows from (3) that $p \wedge_L q = p$ and, hence, $p \leq_{\mathcal{L}} q$. This establishes the right to left implication of (1).

Conversely, suppose that conditions (1) and $(3')$ hold for all $p, q, r \in L$. We show that (3) holds for all $p, q, r \in L$. Suppose first that $p \wedge_L q = r$. If α and β are in Θ and $\alpha(r) = \beta(r)$, then, by $(3')$, a sequence of (p, q)-interpolants between α and β exists. Conversely, if a sequence $(\gamma_0, \ldots, \gamma_{n+1})$ of (p, q)-interpolants exists between α and β, then, by (1), since $r \leq_{\mathcal{L}} p, q$, $\gamma_i(r) = \gamma_{i+1}(r)$ for all $i \leq n$, so $\alpha(r) = \beta(r)$. This gives the left to right implication of (3). Now, suppose that two elements of Θ agree on r if and only if there is a sequence of (p, q)-interpolants between them. If α and β are two elements of Θ that agree on p, then (α, β) is a sequence of (p, q)-interpolants between α and β, so, by hypothesis, α and β agree on r. It follows by (1) that $r \leq_{\mathcal{L}} p$. Similarly, if α and β agree on q, then (α, α, β) is a sequence of (p, q)-interpolants between α

and β, so α and β agree on r. Thus, $r \leq_{\mathcal{L}} q$ and r is a lower bound for $\{p, q\}$. Suppose that t is any lower bound for $\{p, q\}$. If α and β are two elements of Θ that agree on r, then, by hypothesis, there is a sequence $(\gamma_0, \ldots, \gamma_{n+1})$ of (p, q) interpolants between α and β. By (1) and the fact that $t \leq_{\mathcal{L}} p, q$, $\gamma_i(t) = \gamma_{i+1}(t)$ for all $i \leq n$, so $\alpha(t) = \beta(t)$. Thus, any two elements of Θ that agree on r agree on t. It follows from (1) that $t \leq_{\mathcal{L}} r$. Hence, $p \wedge_L q = r$. This establishes the right to left implication of (3) ∎

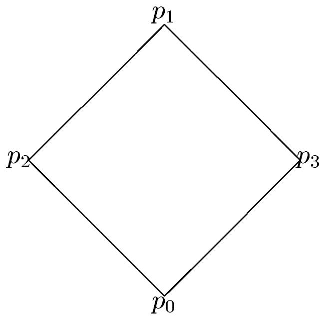
There are still more weakenings of Definition 4 that turn out to be equivalent to it. However, not every variation on the definition is equivalent. In defining representation, Fejer-Shore [3] and Haught [4] replace the second "$\leftrightarrow$" in conditions (2) and (3) with "$\rightarrow$", but leave the first "$\leftrightarrow$" alone. This version is strictly stronger than our definition, in fact, no nontrivial lattice has a "representation" in this stronger sense. (In both papers, the usual definition of representation is what was meant.)

We can prove a result similar to Theorem 6 for computability-theoretic representations of type m.

Theorem 7 *Let $\mathcal{L}$ be a lattice, A be a set and $\Theta = \{\theta_a | a \in A\}$ be a collection of functions with domain L. For $p, q, r \in L$ and $n \in \omega$, let (3_n) and $(3'_n)$ be the conditions obtained from (3) and (3′) by removing "$(\exists n)$". Then, for all $m \in \omega$, Θ is a computability-theoretic A-representation for $\mathcal{L}$ of type m if and only if Θ meets one of the following sets of conditions for all $p, q, r \in L$:*

1. *(1), (2), and (3_m).*
2. *(1), (2′), and $(3'_m)$.*
3. *(2) and $(3'_m)$.*
4. *(2′) and (3_m).*

Proof: First we claim that condition (3_m) holds for all $p, q, r \in L$ if and only if conditions (1) and $(3'_m)$ hold for all $p, q, r \in L$. For $m > 0$, the proof of this is an easy modification of the proof given in Theorem 6 that condition (3) is equivalent to the conjunction of conditions (1) and (3′). (Here it is useful to observe that if $(\gamma_0, \ldots, \gamma_{n+1})$ is a sequence of (p, q)-interpolants between α and β and $m > n$, then there is a sequence $(\gamma'_0, \ldots, \gamma'_{m+1})$ of (p, q)-interpolants between α and β, obtained by adding $m - n$ copies of β to the end of the sequence.) For $m = 0$, most of the proof from Theorem 6 works, except to show that if (1) and $(3'_m)$ hold, then the right to left implication of (3_m) holds. To see this last point, we argue as follows. If $(3'_0)$ holds for all $p, q, r \in L$, then $p \wedge_L q = r$ implies that any members of Θ that agree on r also agree on p. By (1), this implies that if $p \wedge_L q = r$, then $p \leq_{\mathcal{L}} r$, so $p = r$. This in turn implies that L has only one element. Thus, for all $p, q, r \in L$, we have $p \wedge_L q = r$, so the right to left implication of (3_m) holds trivially.



	α_0	α_1	α_2
p_0	0	0	0
p_1	0	1	2
p_2	0	1	0
p_3	0	0	1

Figure 1: The diamond lattice and a representation for it.

Suppose now that Θ is a representation of type m. It is obvious that conditions (3) and (4_m) together imply condition (3_m), so Θ satisfies conditions (1), (2), and (3_m). Now suppose that conditions (1), (2) and (3_m) hold. We have seen that (2) implies ($2'$) and (3_m) clearly implies ($3'_m$), so we have (1), ($2'$) and ($3'_m$). If (1), ($2'$) and ($3'_m$) hold, then, since (1) and ($2'$) imply (2), we have (2) and ($3'_m$). If (2) and ($3'_m$) hold, then from (2) we get (1) and ($2'$). As noted at the beginning of the proof, (1) and ($3'_m$) imply (3_m), so we have ($2'$) and (3_m). Finally, suppose that ($2'$) and (3_m) hold. Then, as shown at the beginning of the proof, (3_m) implies (1). Also, (3_m) clearly implies ($3'$), so we have (1), ($2'$) and ($3'$). By Theorem 6, Θ is a representation. To see that condition (4_m) holds, let p, q be in L and let $r = p \wedge_L q$. If there is a sequence of (p, q)-interpolants between $\alpha, \beta \in \Theta$, then, by (3), (which we have just argued holds) $\alpha(r) = \beta(r)$, so, by (3_m), there is a sequence of (p, q)-interpolants of length $m + 2$ between α and β. Thus, (4_m) holds, and Θ is a representation of type m. ∎

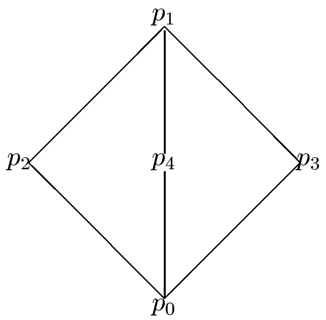
Examples of Representations

Now we give some examples of computability-theoretic representations.

Example 8 The diamond lattice $\mathcal{L}$ (i.e., the Boolean algebra with two generators) and a lattice-theoretic $\{0, 1, 2\}$-representation $\{\alpha_0, \alpha_1, \alpha_2\}$ of type 2 for $\mathcal{L}$ are given in Figure 1. ∎

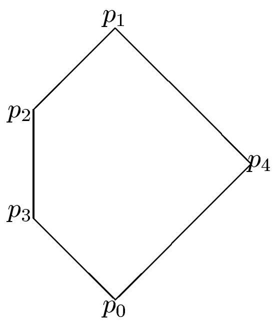
Example 9 The five element modular but nondistributive lattice M_3 and a lattice-theoretic $\{0, 1, 2\}$-representation $\{\alpha_0, \alpha_1, \alpha_2\}$ of type 2 for M_3 are given in Figure 2. ∎

Example 10 The five element nonmodular lattice N_5 and a lattice-theoretic $\{0, 1, 2, 3\}$-representation of type 3 for N_5 are given in Figure 3. ∎



	α_0	α_1	α_2
p_0	0	0	0
p_1	0	1	2
p_2	0	1	0
p_3	0	0	1
p_4	0	1	1

Figure 2: M_3 and a representation for it.



	α_0	α_1	α_2	α_3
p_0	0	0	0	0
p_1	0	1	2	3
p_2	0	1	2	2
p_3	0	0	1	1
p_4	0	1	0	1

Figure 3: N_5 and a representation for it.

Connection Between the Two Types of Representations

As mentioned above, the use of computability-theoretic representations of lattices began with Lerman and Shoenfield, but it was Thomason in [16] who recognized how the type of representation needed in computability theory could be generated from a representation of a lattice as a lattice of equivalence relations.

The connection between the two types of representation is given in the next theorem.

Theorem 11 *Let $\mathcal{L}$ be a lattice and let A be a set. For each computability-theoretic A-representation $\Theta = \{\theta_a | a \in A\}$ of $\mathcal{L}$, define $\Psi(\Theta) : L \to Eq(A)$ by*

$$\Psi(\Theta)(p) = \{(a,b) | \theta_a(p) = \theta_b(p)\}$$

for each $p \in L$.

If η is a lattice-theoretic A-representation of $\mathcal{L}^{\mathrm{dual}}$, define $\Phi(\eta) = \{\delta_a | a \in A\}$ where for each $a \in A$ and $p \in L$

$$\delta_a(p) = [a]_{\eta(p)}.$$

(Here we use $[a]_{\eta(p)}$ to denote the equivalence class of a in the equivalence relation $\eta(p)$.)

Then we have:

1. *For each computability-theoretic A-representation Θ of $\mathcal{L}$, $\Psi(\Theta)(p)$ is in $Eq(A)$ for every $p \in L$ and $\Psi(\Theta)$ is a lattice-theoretic A-representation of $\mathcal{L}^{\mathrm{dual}}$.*

2. *If η is a lattice-theoretic A-representation of $\mathcal{L}^{\mathrm{dual}}$, $\Phi(\eta)$ is a computability-theoretic A-representation for $\mathcal{L}$.*

3. *For every lattice-theoretic A-representation η of $\mathcal{L}^{\mathrm{dual}}$, $\Psi(\Phi(\eta)) = \eta$.*

4. *For any two computability-theoretic A-representations Θ and Θ' of $\mathcal{L}$, we have $\Psi(\Theta) = \Psi(\Theta')$ if and only if Θ and Θ' are equivalent.*

5. *If Θ is a computability-theoretic A-representation, then Θ is computable if and only if $\Psi(\Theta)$ is a computable lattice-theoretic A-representation of $\mathcal{L}^{\mathrm{dual}}$.*

6. *If Θ is a computability-theoretic A-representation and $m \in \omega$, then Θ is type m if and only if $\Psi(\Theta)$ is a type m lattice-theoretic A-representation of $\mathcal{L}^{\mathrm{dual}}$.*

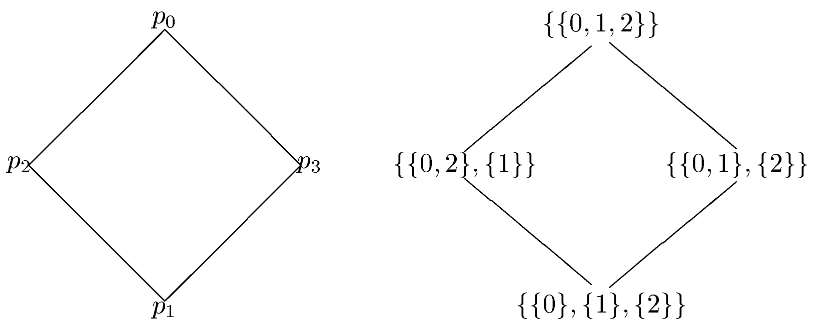


Figure 4: Lattice-theoretic representation corresponding to representation of Example 8.

Proof: The verifications are routine. ∎

If Θ is a computability-theoretic A-representation of a lattice $\mathcal{L}$, then we will refer to $\Psi(\Theta)$ as the lattice-theoretic A-representation of $\mathcal{L}^{\mathrm{dual}}$ corresponding to Θ. According to parts (3) and (4) of the previous theorem, every lattice-theoretic A-representation of $\mathcal{L}^{\mathrm{dual}}$ corresponds to a class of equivalent computability-theoretic A-representations of $\mathcal{L}$.

We will illustrate Theorem 11 by giving the lattice-theoretic representations corresponding to the computability-theoretic representations given in Examples 8 to 10.

Example 12 Figure 4 shows the dual of the diamond lattice $\mathcal{L}$ of Example 8 and then shows the lattice-theoretic $\{0, 1, 2\}$-representation of $\mathcal{L}^{\mathrm{dual}}$ corresponding to the computability-theoretic representation given in Example 8. The lattice-theoretic representation is given by replacing each element of $\mathcal{L}^{\mathrm{dual}}$ by the equivalence relation on $\{0, 1, 2\}$ that it is mapped to. Each equivalence relation is given by its associated partition. ∎

Example 13 Figure 5 shows the lattice-theoretic $\{0, 1, 2\}$- representation for M_3^{dual} corresponding to the representation given in Example 9. ∎

Example 14 Figure 6 shows the lattice-theoretic $\{0, 1, 2, 3\}$- representation for N_5^{dual} corresponding to the representation given in Example 10. ∎

Existence Theorems for Lattice Representations

We now describe the results on the existence of representations proven by mathematicians outside of computability theory that are of most interest to computability theorists. The most basic result is due to Whitman in 1946, answering a question raised by Birkhoff in [1].

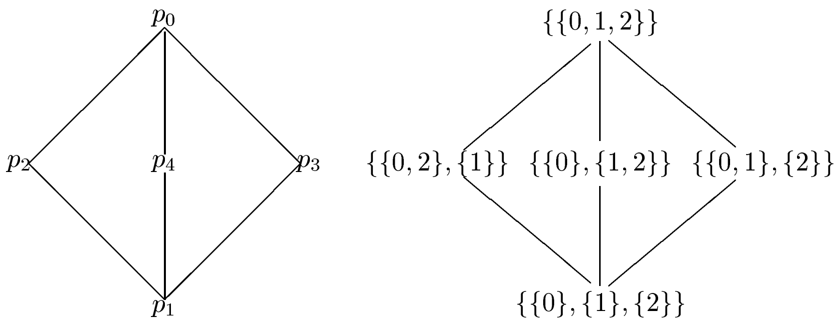


Figure 5: Lattice-theoretic representation corresponding to representation of Example 9.

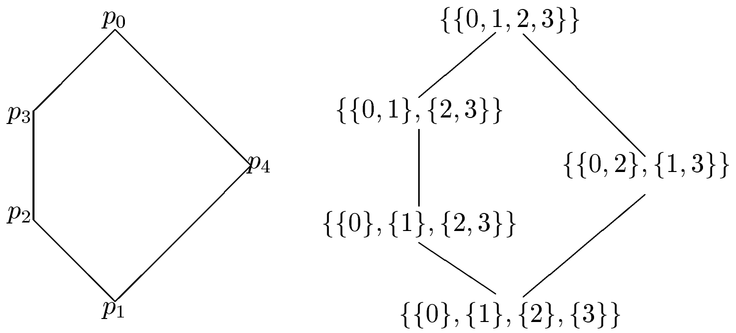


Figure 6: Lattice-theoretic representation corresponding to representation of Example 10.

Theorem 15 (Whitman [17]) *Every lattice has a representation.*

Whitman's proof was simplified and the result sharpened somewhat by Jónsson in 1953.

Theorem 16 (Jónsson [5]) *Every lattice has a representation of type 3.*

Jónsson's paper contains several other interesting results. He shows that if a lattice has a representation of type 2, then it is modular. Since there are nonmodular lattices, this shows that Theorem 16 is the best possible in terms of the type of the representation. Jónsson also shows that every modular lattice has a representation of type 2, so having a representation of type 2 characterizes modularity. By another result of Jónsson, there are modular lattices that do not have any representation of type 1, so his result about representations of modular lattices is also best possible. (The question of which lattices have representations of type 1 is left open in [5]. In [6], Jónsson gives a nice characterization of the complemented lattices that have type 1 representations, and in [7] he gives a complicated characterization of all the lattices having type 1 representations (namely, a lattice has a type 1 representation if and only if it is a model of a certain infinite, computable set of universal Horn formulas). It is still an open problem whether there is a set of equalities such that a lattice has a type 1 representation if and only if it satisfies all of the equalities in the set.) A good overview of the area of lattice representations from the point of view of general algebra is Lampe [8].

Whitman's and Jónsson's constructions yield infinite representations, in general, even when the lattice is finite. In [17], Whitman expressed the hope that every finite lattice has a finite representation. This problem remained open for a long time. It was finally solved by Pudlák and Tůma in 1980.

Theorem 17 (Pudlák and Tůma [12]) *Every finite lattice has a finite representation.*

In the same paper, Pudlák and Tůma state that the following strengthening of the previous theorem follows easily from their methods, but they do not give the proof.

Theorem 18 (Pudlák and Tůma [12]) *For every finite lattice $\mathcal{L}$, there is a number n_0 such that for all finite sets A with $|A| \geq n_0$, there is a lattice representation $\eta : L \to Eq(A)$ that preserves 0 and 1.*

An obvious question is whether Theorems 16 and 17 can be combined to show that every finite lattice has a finite representation of type 3; this is still an open problem.

Pudlák and Tůma's construction is quite difficult and indirect. In any computability-theoretic application of the theorem, if a finite representation with some additional properties is needed, the original finite representation has to be taken as if from a black box and the additional properties have to be obtained by generic transformations applicable to any representation. This is the approach used in Theorem 28. Jónsson's construction, on the other hand, is simple and direct and has been used by Thomason in [15] to show that every finite lattice has a computable representation and by Shore in [13] to show the more general result that every computably presented lattice has a computable representation. Both of these constructions are nothing more than the translation of Jónsson's construction into computability-theoretic terms. (We give Shore's construction in Theorem 20 below.) Jónsson's construction is also the basis for constructions of representations with additional properties. A simple example is our Theorem 27. Fejer [2] has a slightly more complicated example and in Lerman [10] and [11] are much more complicated constructions using Jónsson's method as a starting point.

Constructions in computability theory that use representations are often simpler if a finite representation is used, but can be modified to work with a computable representation, so it can be convenient, but is not essential, to invoke Pudlák and Túma's difficult result. Being able to assume that the representation has type 3 can also make constructions somewhat easier to express, but these constructions will also work even if the representation used does not have any finite type.

In results in complexity theory such as Shore and Slaman [14] that use lattice representations, it is essential to have a finite representation and Pudlák and Túma's result has to be invoked. Here again, fortunately, the constructions work no matter what the type of the representation, so the fact that it is not known if every finite lattice has a finite representation of type 3 does not cause any problem. (Shore and Slaman erroneously claim in [14] that Pudlák and Túma showed that every finite lattice has a finite representation of type 3, but their constructions do not actually need the representation to be type 3.)

Before we can give Shore's translation into computability-theoretic terms of Jónsson's lattice representation construction, we need a preliminary lemma characterizing computable representations.

Lemma 19 *Let $\mathcal{L}$ be a lattice and A be a set. A computability-theoretic A-representation is computable if and only if it is equivalent to some $\Theta' = \{\theta'_a | a \in A\}$ such that each $\theta'_a : L \to \omega$ and there are enumerations without repetitions $L = \{p_i | i \in I_n\}$, $A = \{a_i | i \in I_m\}$, $n, m \in \omega^+$ with $\lambda i, j.\theta'_{a_j}(p_i)$ computable.*

Proof: First suppose that $\Theta = \{\theta_a | a \in A\}$ is a computable computability-theoretic A-representation of $\mathcal{L}$. Let $L = \{p_i | i \in I_n\}$, $A = \{a_i | i \in I_m\}$, $n, m \in \omega^+$, be enumerations such that $\{\langle i, j, k\rangle | \theta_{a_j}(p_i) = \theta_{a_k}(p_i)\}$ is computable. Define $\Theta' = \{\theta'_a | a \in A\}$ by letting $\theta'_a(p_i)$ be the least k such that $\theta_{a_k}(p_i) = \theta_a(p_i)$. Then each θ'_a takes its values in ω, Θ' is equivalent to Θ, and with respect to the given enumerations of L and A, $\lambda i, j.\theta'_{a_j}(p_i)$ is computable.

Conversely, if Θ' is equivalent to Θ and meets the given conditions, then Θ' is computable (using the enumerations of L and A with respect to which the condition is satisfied) and hence Θ is computable. ∎

In particular, if we have a representation $\Theta = \{\theta_i | i \in I_m\}$ for $\mathcal{L}$, $L = \{p_i | i \in I_n\}$, each θ_i takes values in ω and $\lambda i, j.\theta_j(p_i)$ is computable, then Θ is computable.

Theorem 20 (Shore [13], Jónsson [5]) *Every computably presented lattice has a computable representation of type 3.*

Proof: Let $\{p_i | i \in I_n\}$ be an enumeration without repetitions of L with $n \in \omega^+$ such that meet and join (and hence $\leq_{\mathcal{L}}$) are effective with respect to the enumeration. For each $i \in I_n$, define two functions $\beta_{2i}, \beta_{2i+1} : L \to \omega$ by

$$\begin{aligned} \beta_{2i}(p_j) &= 2^{2i} \\ \beta_{2i+1}(p_j) &= \begin{cases} 2^{2i} & \text{if } p_j \leq p_i \\ 2^{2i+1} & \text{if } p_j \not\leq p_i \end{cases} \end{aligned}$$

It is easy to see that the collection of functions $\{\beta_i | i < 2n\}$ (we take $2\omega = \omega$ here) satisfies conditions (1) and ($2'$) and that any collection of functions containing all the β_i's will satisfy the right-to-left implication of (1).

Now suppose that Θ' is any collection of functions with domain L satisfying the left-to-right implication of condition (1) and also condition ($2'$). Suppose in addition that p_i and p_j are incomparable, $p_i \wedge_{\mathcal{L}} p_j = p_k$ and that $\alpha(p_k) = \beta(p_k)$ for some $\alpha, \beta \in \Theta'$. We show how to add $\gamma_1, \gamma_2, \gamma_3$ to Θ' so that the augmented collection still satisfies the left to right implication of (1) and also condition ($2'$) and $(\alpha, \gamma_1, \gamma_2, \gamma_3, \beta)$ is a sequence of (p_i, p_j)-interpolants from α to β. Let w, x, y, z be objects not in the range of any of the functions in Θ' and define

$$\begin{aligned} \gamma_1(p_\ell) &= \begin{cases} \alpha(p_\ell) & \text{if } p_\ell \leq p_i \\ w & \text{if } p_\ell \not\leq p_i \end{cases} \\ \gamma_2(p_\ell) &= \begin{cases} \gamma_1(p_\ell) & \text{if } p_\ell \leq p_j \\ x & \text{if } p_\ell \leq p_i \text{ and } p_\ell \not\leq p_j \\ y & \text{otherwise} \end{cases} \\ \gamma_3(p_\ell) &= \begin{cases} \beta(p_\ell) & \text{if } p_\ell \leq p_j \\ x & \text{if } p_\ell \leq p_i \text{ and } p_\ell \not\leq p_j \\ z & \text{otherwise.} \end{cases} \end{aligned}$$

It is tedious, but not too difficult, to check that $\gamma_1, \gamma_2, \gamma_3$ have the desired properties.

Now, we construct the desired representation Θ in stages. Let $\Theta_0 = \{\beta_0\}$. Given Θ_s satisfying condition ($2'$) and the left-to-right implication of condition (1), let Θ'_{s+1} be $\Theta_s \cup \{\beta_{s+1}\}$, if $s + 2 \leq 2n$, and be Θ_s, otherwise. Obtain

Θ_{s+1} from Θ'_{s+1} as follows. For each $i, j, k \in I_n$ such that $i, j, k \leq s$, $p_i |_{\mathcal{L}} p_j$ and $p_i \wedge_{\mathcal{L}} p_j = p_k$ and each $\alpha, \beta \in \Theta'_{s+1}$ such that $\alpha(p_k) = \beta(p_k)$ and there is no sequence $(\delta_0, \ldots, \delta_4)$ of (p_i, p_j)-interpolants from α to β in Θ'_{s+1}, add to Θ'_{s+1} the three interpolants $\gamma_1, \gamma_2, \gamma_3$ constructed using the method of the previous paragraph. Each time interpolants are added, choose for w, x, y, z odd numbers that do not appear in the range of any function in Θ'_{s+1} and use different values for w, x, y, z for each sequence of interpolants added at stage $s+1$. If $\Theta_{s+1} = \Theta_s$, then stop with $\Theta = \Theta_s$. Otherwise, replace s with $s+1$ and repeat the process.

If the construction does not stop at the end of some stage, then let Θ be the union of all the Θ_s. In either case, let $\{\theta_i | i \in I_m\}$ be an enumeration of Θ in the order the elements were added. It follows easily by induction that for all s for which Θ_s is defined, $\Theta_s \cup \{\beta_{2i}, \beta_{2i+1} | i \in I_n\}$ satisfies the left-to-right implication of condition (1) and also satisfies condition $(2')$. It follows from this that Θ satisfies these same conditions. Also, Θ meets the right-to-left implication of (1) since it contains all the β_i's. Θ meets condition $(3'_3)$ by construction. (Note that if p_i and p_j are comparable, $p_i \wedge_{\mathcal{L}} p_j = p_k$ and $\alpha(p_k) = \beta(p_k)$ for $\alpha, \beta \in \Theta$, then Θ automatically contains the desired interpolant sequence. For instance, if $p_j \leq_{\mathcal{L}} p_i$, then $(\alpha, \alpha, \beta, \beta, \beta)$ is the desired sequence.) Thus, by Theorem 7, Θ is a representation for $\mathcal{L}$ of type 3. Since $\mathcal{L}$ is computably presented, the construction is effective and hence $\lambda i, j.\theta_j(p_i)$ is computable. By Lemma 19, Θ is computable. ∎

Additional Properties of Representations

Some additional useful properties that a computability-theoretic representation $\Theta = \{\theta_a | a \in A\}$ might have are the following.

$$A \text{ is an initial segment of } \omega \tag{5}$$

$$\text{The range of each } \theta_a \text{ is a subset of } A \tag{6}$$

$$(\forall p \in L)(\theta_0(p) = 0) \tag{7}$$

$$\text{If } \mathcal{L} \text{ has a least element } 0_{\mathcal{L}}, \text{ then } (\forall \alpha \in \Theta)(\alpha(0_{\mathcal{L}}) = 0) \tag{8}$$

$$(\forall a, b \in A)(a \neq b \rightarrow \theta_a \neq \theta_b) \tag{9}$$

Condition (7) of course assumes that $0 \in A$. We discuss how each of these properties can be obtained.

If A can be enumerated, then (5) is easily obtained.

Theorem 21 *Let $\mathcal{L}$ be a lattice and $\Theta = \{\theta_a | a \in A\}$ be a computability-theoretic A-representation for $\mathcal{L}$. Let $A = \{a_i | i \in I_m\}$ with $m \in \omega^+$ be an enumeration of A without repetitions. Define $\Theta' = \{\theta'_i | i \in I_m\}$ by $\theta'_i(p) = \theta_{a_i}(p)$. Then Θ' is an I_m- representation of $\mathcal{L}$. If Θ is computable with respect to the enumerations $A = \{a_i | i \in I_n\}$, $L = \{p_i | i \in I_m\}$, then Θ' is computable with respect to the same enumeration of L and the natural enumeration of I_m.*

Proof: This is immediate. ∎

Conditions (6) and (7) are dealt with by another simple construction.

Theorem 22 *Let $\mathcal{L}$ be a lattice and $\Theta = \{\theta_a | a \in A\}$ be a computability-theoretic A-representation for $\mathcal{L}$. Let a well-ordering of A be given. Define $\Theta' = \{\theta'_a | a \in A\}$ by letting $\theta'_a(p)$ be the least element b of A such that $\theta_b(p) = \theta_a(p)$. Then, Θ' is equivalent to Θ, the range of each θ' is a subset of A and if $A \neq \emptyset$ and a_0 is the least element of A, then $\theta'_{a_0}(p) = a_0$ for all $p \in L$.*

Proof: Immediate. ∎

Turning now to condition (8), we introduce a weaker condition, assuming that $\mathcal{L}$ has a least element.

$$(\forall \alpha, \beta \in \Theta)(\alpha(0_{\mathcal{L}}) = \beta(0_{\mathcal{L}})) \tag{8'}$$

It is easy to see what condition $(8')$ means in terms of the associated lattice-theoretic representation.

Theorem 23 *Let $\mathcal{L}$ be a lattice with least element $0_{\mathcal{L}}$ and Θ be a computability-theoretic A-representation for $\mathcal{L}$. Then, Θ meets condition $(8')$ if and only if $\Psi(\Theta)(0_{\mathcal{L}}) = A \times A$ (that is, if and only if $\Psi(\Theta)$, the lattice-theoretic A-representation of $\mathcal{L}^{\text{dual}}$ corresponding to Θ, preserves greatest element).*

Proof: This is immediate, since the condition $\theta_a(0_{\mathcal{L}}) = \theta_b(0_{\mathcal{L}})$ is equivalent to $(a, b) \in \Psi(\Theta)(0_{\mathcal{L}})$. ∎

We now show how one can transform a representation that does not meet condition $(8')$ into one that does. This construction is due to Ralph McKenzie and is given in [16].

Theorem 24 (McKenzie, see Thomason [16]) *Suppose that $\mathcal{L}$ is a lattice with greatest element $1_{\mathcal{L}}$ and η is a lattice-theoretic A-representation for $\mathcal{L}$. Let $A/\eta(1_{\mathcal{L}})$, the set of equivalence classes of the equivalence relation $\eta(1_{\mathcal{L}})$, be given as $\{B_i | i \in I\}$ for some index set I, where $i \neq j$ implies $B_i \neq B_j$, let C be $\prod_{i \in I} B_i$, i.e., C consists of all sequences $x = (x_i)_{i \in I}$ such that for all $i \in I$, $x_i \in B_i$, and let b be a particular element of C.*

Define A' to be the set of all $x \in C$ such that $\{i \in I | x_i \neq b_i\}$ is finite and define $\eta' : L \to Eq(A')$ by

$$\eta'(p) = \{(x, y) \in A' \times A' | (\forall i \in I)(x_i \eta(p) y_i)\}$$

for all $p \in L$. Then, η' is a lattice-theoretic A'- representation for $\mathcal{L}$ such that $\eta'(1_{\mathcal{L}}) = A' \times A'$. Further, if η is type m, then so is η'; if η is finite, then so is η'; and if η is computable, then it is possible to choose an indexing $\{B_i | i \in I\}$ for $A/\eta(1_{\mathcal{L}})$ and a $b \in C$ such that η' is computable.

Proof: Let $p, q, r \in L$. If $p \leq_{\mathcal{L}} q$, then $\eta(p) \subseteq \eta(q)$, so $\eta'(p) \subseteq \eta'(q)$. If $p \not\leq_{\mathcal{L}} q$, then $\eta(p) \not\subseteq \eta(q)$, say $(a, a') \in \eta(p) - \eta(q)$. Since $p \leq_{\mathcal{L}} 1_{\mathcal{L}}$, $(a, a') \in \eta(p) \subseteq \eta(1_{\mathcal{L}})$, say $\{a, a'\} \subseteq B_{i_0}$. Define $x, y \in A'$ by $x_{i_0} = a$, $y_{i_0} = a'$ and $x_i = y_i = b_i$ for $i \neq i_0$. Then, $(x, y) \in \eta'(p) - \eta'(q)$, so $\eta'(p) \not\subseteq \eta'(q)$.

Suppose now that $p \vee_{\mathcal{L}} q = r$. Then, $\eta'(p) \cup \eta'(q) \subseteq \eta'(r)$, so $(\eta'(p) \cup \eta'(q))^+ \subseteq \eta'(r)$. Suppose that $(x, y) \in \eta'(r)$. Let $J = \{i \in I | x_i \neq y_i\}$. By definition of A', J is finite. For each $j \in J$, $x_j \eta(r) y_j$, so, since $\eta(r) = (\eta(p) \cup \eta(q))^+$, there is $m_j \geq 0$ and a sequence $a_0^j, \ldots, a_{m_j+1}^j$ of elements of A such that $a_0^j = x_j$, $a_{m_j+1}^j = y_j$ and for all $k \leq m_j$, if k is even, then $a_k^j \eta(p) a_{k+1}^j$ and if k is odd, then $a_k^j \eta(q) a_{k+1}^j$. Let $m = \max\{m_j | j \in J\}$ and define $a_k^j = y_j$ for $j \in J$ and $m_j + 1 < k \leq m + 1$. For each $j \in J$, $a_0^j = x_j \in B_j$. If $k \leq m$ and $a_k^j \in B_j$, then, since $(a_k^j, a_{k+1}^j) \in \eta(p) \cup \eta(q) \subseteq \eta(1_{\mathcal{L}})$, $a_{k+1}^j \in B_j$. Thus, $a_k^j \in B_j$ for all $j \in J$ and $k \leq m+1$ and we may define $x^0, \ldots, x^{m+1} \in A'$ by $(x^k)_j = a_k^j$ for $j \in J$, $(x^k)_i = b_i$ for $i \in I - J$. Then, $x^0 = x$, $x^{m+1} = y$ and for all $k \leq m$, if k is even, then $x^k \eta'(p) x^{k+1}$ and if k is odd, then $x^k \eta'(q) x^{k+1}$. Thus, $\eta'(r) \subseteq (\eta'(p) \cup \eta'(q))^+$.

If $p \wedge_{\mathcal{L}} q = r$, $\eta'(p) \cap \eta'(q) = \eta'(r)$ follows easily from the corresponding fact for η.

If $x, y \in A'$, then for all $i \in I$, $x_i, y_i \in B_i$, so $x_i \eta(1_{\mathcal{L}}) y_i$. Thus, $(x, y) \in \eta'(1_{\mathcal{L}})$ and $\eta'(1_{\mathcal{L}}) = A' \times A'$.

If A is type m, then the proof given above that $p \vee_{\mathcal{L}} q = r$ implies $\eta'(r) = (\eta'(p) \cup \eta'(q))^+$ is easily modified to show that $(\eta'(p) \cup \eta'(q))^+ = P_m(\eta'(p), \eta'(q))$ and hence that η' is type m.

If A is finite, then A' is obviously finite.

If η is computable, let $L = \{p_i | i \in I_n\}$ and $A = \{a_i | i \in I_m\}$ with $n, m \in \omega^+$ be enumerations such that $\{\langle i, j, k\rangle | a_j \eta(p_i) a_k\}$ is computable. If $A/\eta(1_{\mathcal{L}})$ is finite, then η' is finite, so is computable. Otherwise, we define an infinite sequence $b = (b_i)_{i \in \omega}$ of elements of A by the following recursion: we let b_i be a_j where j is minimal so that $\neg(a_j \eta(1_{\mathcal{L}}) b_k)$ for all $k < i$. Also, define $B_i = [b_i]_{\eta(1_{\mathcal{L}})}$ for $i \in \omega$. It is not too hard to see that the set A' resulting from this choice of indexing of $A/\eta(1_{\mathcal{L}})$ and choice of b can be enumerated as $\{a'_i | i \in \omega\}$ so that $\{\langle i, j, k\rangle | a'_j \eta'(p_i) a'_k\}$ is computable. ∎

We now look at what condition (9) means lattice-theoretically.

Theorem 25 *Let $\mathcal{L}$ be a lattice, A be a set, $\Theta = \{\theta_a | a \in A\}$ be a computability-theoretic A-representation for $\mathcal{L}$ and $\Psi(\Theta)$ be the corresponding lattice-theoretic A-representation for $\mathcal{L}^{\mathrm{dual}}$.*

1. Θ *satisfies condition (9) if and only* $\Psi(\Theta)$ *satisfies:*

$$(\forall a, b \in A)(a \neq b \rightarrow (\exists p \in L)((a,b) \notin \Psi(\Theta)(p))).$$

2. If $\mathcal{L}$ *has a greatest element* $1_{\mathcal{L}}$*, then* Θ *satisfies condition (9) if and only if*

$$\Psi(\Theta)(1_{\mathcal{L}}) = \{(a,a)|a \in A\}$$

(i.e., $\Psi(\Theta)$ *preserves least element).*

Proof: The first part is immediate from the definition of Ψ. For the second, note that $(\exists p \in L)((a,b) \notin \Psi(\Theta)(p))$ is equivalent to $(a,b) \notin \Psi(\Theta)(1_{\mathcal{L}})$, since $\Psi(\Theta)(1_{\mathcal{L}}) \subseteq \Psi(\Theta)(p)$ for all $p \in L$. ∎

It is obvious how to obtain condition (9). Maintaining computability is a little more delicate.

Theorem 26 *Let* $\mathcal{L}$ *be a lattice and* $\Theta = \{\theta_a|a \in A\}$ *be a computability-theoretic* A*-representation for* $\mathcal{L}$*.*

Suppose that A *is well-ordered by* $\preceq$*. Let* $A' = \{a \in A|(\forall b \in A)(b \prec a \rightarrow \theta_b \neq \theta_a)\}$ *and define* $\Theta' = \{\theta'_a|a \in A'\}$ *by* $\theta'_a = \theta_a$ *for all* $a \in A'$*. Then* Θ' *is a computability-theoretic* A'*-representation for* $\mathcal{L}$ *which meets (9).*

Further, if $\mathcal{L}$ *has a greatest element, and* Θ *is computable, say with enumerations* $A = \{a_i|i \in I_n\}$ *and* $L = \{p_i|i \in I_m\}$*, and the above process is carried out with respect to the given ordering of* A*, then* Θ' *is computable.*

Proof: The first part is obvious. For the second part, let $1_{\mathcal{L}} = p_{i_1}$. Then, $A' = \{a_j|(\forall k < j)(\theta_{a_k}(p_{i_1}) \neq \theta_{a_j}(p_{i_1}))\}$, so $B = \{x|a_x \in A'\}$ is computable, say f with domain $I_{n'}$ enumerates B in increasing order. Let $a'_i = a_{f(i)}$, so $A' = \{a'_i|i \in I_{n'}\}$. Then, $\{\langle i,j,k\rangle|\theta'_{a'_j}(p_i) = \theta'_{a'_k}(p_i)\} = \{\langle i,j,k\rangle|\theta_{a_{f(j)}}(p_i) = \theta_{a_{f(k)}}(p_i)\}$ is computable. ∎

Sharpened Existence Theorems

We can now give some sharpened existence theorems for lattice representations satisfying the additional conditions we have been discussing.

Theorem 27 *Let* $\mathcal{L}$ *be a computably presented lattice with join and meet computable with respect to the enumeration* $L = \{p_i|i \in I_n\}$*. Then there is a computability-theoretic* A*-representation* Θ *for* $\mathcal{L}$ *of type 3 that satisfies all of the conditions (5)-(9) and such that* $\lambda i, j.\theta_j(p_i)$ *is computable. (So, by Lemma 19,* Θ *is computable.)*

Proof: We begin by modifying the construction of Theorem 20 in two ways, one carried out if $\mathcal{L}$ has a least element, the other if it has a greatest element.

Towards meeting condition (8), if $\mathcal{L}$ has least element $0_{\mathcal{L}}$, then we modify the definition of the β_i's so that each β_i maps $0_{\mathcal{L}}$ to 0, but is otherwise the same. It is easy to verify that the modified collection of β_i's still satisfies conditions (1) and (2′) and that any collection of functions that contains the β_i's will meet the right-to-left implication of (1). Further, $\lambda j, i.\beta_i(p_j)$ is computable. (We don't claim, of course, that given a computably presented lattice, one can determine if the lattice has a least element and, if so, which element is the least one, but the collection of β's is still computable.) It is also easily verified that if the interpolant adding process of the proof of Theorem 20 is applied to two functions that both map $0_{\mathcal{L}}$ to 0, then the three interpolants produced have the same property. Thus, all functions in the representation produced map $0_{\mathcal{L}}$ to 0.

Towards meeting condition (9), it is easily checked that in the construction of Theorem 20, all elements added to the representation are distinct from each other with one exception: if p_{i_1} is the greatest element of $\mathcal{L}$, then β_{2i_1} and β_{2i_1+1} are the same. Thus, we modify the construction so that at stage $s+1 = 2i_1+1$, we do not add β_{s+1} to the representation.

Let $\Theta' = \{\theta'_i | i \in I_m\}$ be the representation produced by this modified construction. Then, Θ' satisfies conditions (5), (8) and (9) and $\lambda i, j.\theta'_j(p_i)$ is computable. Let Θ be obtained from Θ' by applying the process of Theorem 22 to Θ', using the natural ordering of I_m. Then, Θ meets conditions (6) and (7). Since Θ is equivalent to Θ', Θ meets conditions (5) and (9). Finally, Θ meets condition (8) and $\lambda i, j.\theta_j(p_i)$ is computable since Θ' has these properties and the process of Theorem 22 preserves these properties. ∎

Theorem 28 *Let $\mathcal{L}$ be a finite lattice. Then $\mathcal{L}$ has a finite representation satisfying all of the additional properties (5)-(9).*

Proof: If L has only one element p, then we may take $A = \{0\}$ and $\Theta = \{\theta_0\}$ where $\theta_0(p) = 0$. Thus, we may suppose that L contains at least two elements. Then, for any A-representation of $\mathcal{L}$, we must have $A \neq \emptyset$.

We first wish to produce a finite computability-theoretic A_1-representation Θ_1 for $\mathcal{L}$ that satisfies conditions (8′) and (9). If we are willing to use Theorem 18, which is claimed but not proven in [12], then we proceed as follows. Using this theorem, let η_1 be a finite lattice-theoretic A_1 representation for $\mathcal{L}^{\text{dual}}$ that preserves least and greatest elements and let $\Theta_1 = \Phi(\eta_1)$. Then, by Theorems 23 and 25 (since $\Psi(\Theta_1) = \eta_1$), Θ_1 is a finite A_1-representation for $\mathcal{L}$ that meets both conditions (8′) and (9).

Alternatively, using Theorem 17, we may begin with a lattice-theoretic A_2-representation η_2 for $\mathcal{L}^{\text{dual}}$ for some finite set A_2. Next, we may obtain η_3, a finite lattice-theoretic A_3-representation for $\mathcal{L}^{\text{dual}}$ that preserves greatest element by applying the method of Theorem 24. Then let $\Theta_0 = \Phi(\eta_3)$. By Theorem 23, Θ_0 is a finite A_3-representation that meets condition (8′). Now, obtain Θ_1 from Θ_0 using the process of Theorem 26 and an arbitrary linear ordering of A_3. Then, A_1 is finite, and Θ_1 meets condition (9) and also condition (8′), since as sets of functions Θ_0 and Θ_1 are the same.

However Θ_1 is obtained, we let $A_1 = \{a_i | i \in I_m\}$ for some $m \in \omega$ and construct Θ_2 from Θ_1 using the method of Theorem 21. Then, Θ_2 is an I_m-representation for $\mathcal{L}$, so meets condition (5). Since Θ_2 contains the same functions as Θ_1, condition ($8'$) still holds and it is easy to see that condition (9) is preserved as well.

Finally, obtain Θ_3 from Θ_2 using the method from Theorem 22, with the natural ordering of I_m. Then, Θ_3 is an I_m-representation for $\mathcal{L}$ meeting conditions (5), (6) and (7) (because $I_m \neq \emptyset$). Since Θ_3 is equivalent to Θ_2, Θ_3 meets conditions ($8'$) and (9). Since conditions (7) and (8') together imply (8), Θ_3 is the desired representation. ∎

An Application of Lattice Representations

We complete our presentation by giving a simple but prototypical application of lattice representations to proving lattice embedding theorems in computability theory. We will show that every computably presented lattice is lattice embeddable into the $\mathcal{D}(\leq \mathbf{0}')$, the Turing degrees less than or equal to $\mathbf{0}'$, preserving least element. This result is due to Lerman [11], who actually showed the much more difficult result that each such lattice could be embedded as an initial segment below $\mathbf{0}'$. The proof we give is essentially from Shore [13], although he does not take the extra steps needed to preserve least element.

Theorem 29 *Every computably presented lattice can be lattice-embedded into $\mathcal{D}(\leq \mathbf{0}')$ preserving least element.*

Proof: Let $\mathcal{L}$ be a computably presented lattice and let $L = \{p_i | i \in I_n\}$ be an enumeration of L such that join and meet are computable with respect to the enumeration. Let $\Theta = \{\theta_j | j \in I_m\}$ be a computability-theoretic representation for $\mathcal{L}$ which satisfies the properties of Theorem 27. We will define a function $f : \omega \to I_m$ such that $f \leq_T \emptyset'$. Then, for each $p \in L$, we define $f^p : \omega \to I_m$ by $f^p(x) = \theta_{f(x)}(p)$. Letting $\mathbf{a}_p = \deg_T(f^p)$, the desired embedding will be the map $p \mapsto \mathbf{a}_p$.

No matter how $f \leq_T \emptyset'$ is defined, the following properties will automatically hold for all $p, q, r \in L$.

$$\mathbf{a}_p \leq \mathbf{0}' \tag{11}$$

$$p \leq_{\mathcal{L}} q \to \mathbf{a}_p \leq_T \mathbf{a}_q \tag{12}$$

$$p \vee_{\mathcal{L}} q = r \to \mathbf{a}_p \vee_T \mathbf{a}_q = \mathbf{a}_r \tag{13}$$

$$\text{If } \mathcal{L} \text{ has a least element, then } \mathbf{a}_{0_{\mathcal{L}}} = \mathbf{0} \tag{14}$$

The correctness of (11) follows from the definition $f^p(x) = \theta_{f(x)}(p)$, the assumption that $f \leq_T \emptyset'$ and the fact that $\lambda i, j.\theta_j(p_i)$ is computable.

To establish (12), it suffices to show that if $p \leq_{\mathcal{L}} q$, then $f^p \leq_T f^q$. Given x, use an f^q oracle to get $f^q(x)$. Then, find the least $i \in I_m$ such that $f^q(x) = \theta_i(q)$. This can be done effectively, given $f^q(x)$, since $\lambda i, j.\theta_j(p_i)$ is computable. We know such an i exists since $i = f(x)$ is one such. The i we find might not be $f(x)$. Nonetheless, $\theta_i(q) = \theta_{f(x)}(q)$, so, by (1), $\theta_i(p) = \theta_{f(x)}(p) = f^p(x)$ and we have computed $f^p(x)$ effectively in f^q.

To see that (13) holds, assume that $p \vee_{\mathcal{L}} q = r$. We must show that $f^p \oplus f^q \equiv_T f^r$. We have $f^p, f^q \leq_T f^r$ from (12), so we need to show $f^r \leq_T f^p \oplus f^q$. Given x, find the least $i \in I_m$ such that $\theta_i(p) = f^p(x)$ and $\theta_i(q) = f^q(x)$. (One such i is $f(x)$.) Since θ_i and $\theta_{f(x)}$ agree on p and q, by (2), they agree on r and $\theta_i(r) = \theta_{f(x)}(r) = f^r(x)$. Thus, we have computed $f^r(x)$ effectively in $f^p \oplus f^q$.

If $\mathcal{L}$ has a least element, then by (8), for all $x \in \omega$, $f^{0_{\mathcal{L}}}(x) = \theta_{f(x)}(0_{\mathcal{L}}) = 0$, so (14) holds.

Thus, for our construction to succeed, we need to meet the following two additional conditions for all $p, q, r \in L$.

$$p \not\leq_{\mathcal{L}} q \rightarrow \mathbf{a}_p \not\leq_T \mathbf{a}_q \tag{15}$$

$$p \wedge_{\mathcal{L}} q = r \rightarrow \mathbf{a}_p \wedge_T \mathbf{a}_q = \mathbf{a}_r. \tag{16}$$

We will meet (15) if we meet all of the following requirements for $e \in \omega, p, q \in L$ such that $p \not\leq_{\mathcal{L}} q$.

$$\{e\}^{f^q} \neq f^p. \tag{17}$$

We will meet (16) if we meet all of the following requirements for $e_0, e_1 \in \omega$, $p, q, r \in L$ with $p \wedge_{\mathcal{L}} q = r$.

$$\{e_0\}^{f^p} = \{e_1\}^{f^q} = g, \text{ total} \rightarrow g \leq_T f^r. \tag{18}$$

Our construction will be a finite extension construction using a $\emptyset'$ oracle. Thus, it suffices to show that given a string $\sigma \in {I_m}^{<\omega}$ and a requirement of the form (17) or (18), one can find effectively in $\emptyset'$ a string $\tau \in {I_m}^{<\omega}$ extending σ such that if f is any function from ω to I_m that extends τ, then the given requirement is met. It will be useful to have the following notation. If $\sigma \in {I_m}^{<\omega}$ and $p \in L$, define $\sigma^p \in {I_m}^{<\omega}$ by $|\sigma^p| = |\sigma|$ and $\sigma^p(i) = \theta_{\sigma(i)}(p)$ for $i < |\sigma|$.

Suppose first that we are given $\sigma \in {I_m}^{<\omega}$ and a requirement of the form (17). We will force $\{e\}^{f^q}(t) \neq f^p(t)$ on the first available t, namely, $t = |\sigma|$. By (1), since $p \not\leq_{\mathcal{L}} q$, there are $i, j \in I_m$ such that $\theta_i(q) = \theta_j(q)$ but $\theta_i(p) \neq \theta_j(p)$. Such a pair i, j can be found effectively. Ask $\emptyset'$ if there is a $\rho \in {I_m}^{<\omega}$ such that $\rho \supseteq \sigma * i$ and $\{e\}^{\rho^q}(|\sigma|) \downarrow$. If not, set $\tau = \sigma * i$. Then, if f extends τ, we have f^q extends τ^q, so $\{e\}^{f^q}(|\sigma|) \uparrow$ and the requirement is met. If such a ρ does exist, take the first such in some standard ordering of strings. If $\{e\}^{\rho^q}(|\sigma|) \neq \rho^p(|\sigma|)$, take $\tau = \rho$. We will then have, for f extending τ, $\{e\}^{f^q}(|\sigma|) = \{e\}^{\rho^q}(|\sigma|) \neq \rho^p(|\sigma|) = f^p(|\sigma|)$. Otherwise, i.e., if $\{e\}^{\rho^q}(|\sigma|) = \rho^p(|\sigma|)$, let τ be the same as ρ but with $\tau(|\sigma|) = j$. Even though we have switched from i to j, we will still have $\tau^q = \rho^q$ since $\theta_i(q) = \theta_j(q)$, but $\tau^p(|\sigma|) = \theta_{\tau(|\sigma|)}(p) = \theta_j(p)$, while $\rho^p(|\sigma|) = \theta_{\rho(|\sigma|)}(p) = \theta_i(p)$, so $\tau^p(|\sigma|) \neq \rho^p(|\sigma|)$. Thus, if f extends τ, we have $\{e\}^{f^q}(|\sigma|) = \{e\}^{\tau^q}(|\sigma|) = \{e\}^{\rho^q}(|\sigma|) = \rho^p(|\sigma|) \neq \tau^p(|\sigma|) = f^p(|\sigma|)$ and the requirement is met.

Finally, suppose we are given $\sigma \in {I_m}^{<\omega}$ and a requirement of the form (18). Ask $\emptyset'$ if there are strings $\rho_0, \rho_1 \in {I_m}^{<\omega}$ and $x \in \omega$ such that $\sigma \subseteq \rho_0, \rho_1$, $|\rho_0| = |\rho_1|$, $\rho_0^r = \rho_1^r$ and $\{e_0\}^{\rho_0^p}(x) \downarrow \neq \{e_1\}^{\rho_1^q}(x) \downarrow$. If no such ρ_0, ρ_1, x exist, let $\tau = \sigma$. To see that the requirement is met in this case, suppose that f extends τ and that $\{e_0\}^{f^p} = \{e_1\}^{f^q} = g$, total. We show how to compute g from f^r. Given x, find a $\rho \supseteq \sigma$ such that $\rho^r \subseteq f^r$ and $\{e_0\}^{\rho^p}(x) \downarrow$. (A large enough initial segment of f is one such ρ.) We can find ρ effectively in f^r. We claim that $\{e_0\}^{\rho^p}(x) = g(x)$. If not, then since $g(x) = \{e_1\}^{f^q}(x)$, we can take ρ' to be a sufficiently long initial segment of f to get $\{e_1\}^{\rho'^q}(x) = g(x) \neq \{e_0\}^{\rho^p}(x)$. We may also assume that $|\rho| = |\rho'|$ by padding the shorter string so that the two strings agree on the extension. Then, ρ, ρ', x satisfy the conditions given above for ρ_0, ρ_1, x, contradicting the assumption that no such ρ_0, ρ_1, x exist. Thus, $\{e_0\}^{\rho^p}(x) = g(x)$ and $g \leq_T f^r$.

Now suppose that ρ_0, ρ_1, x as above do exist. Fix the first such in some effective ordering. We will show how to define τ in such a way that it meets the requirement by forcing the antecedent to be false. Let $\{e_0\}^{\rho_0^p}(x) = A, \{e_1\}^{\rho_1^q}(x) = B$. Let ν_1, ν_2, ν_3 be interpolant strings for ρ_0, ρ_1, (i.e., for $r = 1, 2, 3$, $\sigma \subseteq \nu_r, |\nu_r| = |\rho_0|$, and for $|\sigma| \leq y < |\rho_0|, (\theta_{\rho_0(y)}, \theta_{\nu_1(y)}, \theta_{\nu_2(y)}, \theta_{\nu_3(y)}, \theta_{\rho_1(y)})$ is a sequence of (p,q)-interpolants from $\theta_{\rho_0(y)}$ and $\theta_{\rho_1(y)}$). The strings ν_1, ν_2, ν_3 exist by (3_3) and can be found effectively, given ρ_0 and ρ_1. Also, $\rho_0^p = \nu_1^p, \nu_1^q = \nu_2^q, \nu_2^p = \nu_3^p, \nu_3^q = \rho_1^q$, so $\{e_0\}^{\nu_1^p}(x) = A, \{e_1\}^{\nu_3^q}(x) = B$.

Ask $\emptyset'$ if there is a ζ_1 containing ν_1 such that $\{e_1\}^{\zeta_1^q}(x) \downarrow$. If not, set $\tau = \nu_1$; if f extends τ, then $\{e_1\}^{f^q}(x) \uparrow$. If so, take the first such ζ_1. If $\{e_1\}^{\zeta_1^q}(x) \neq A$, then set $\tau = \zeta_1$; if f extends τ, then $\{e_0\}^{f^p}(x) = \{e_0\}^{{\nu_1}^p}(x) = A \neq \{e_1\}^{{\zeta_1}^q}(x) = \{e_1\}^{f^q}(x)$. Otherwise, we have $\{e_1\}^{\zeta_1^q}(x) = A = \{e_0\}^{\zeta_1^p}(x)$; let ζ_2' be obtained from ν_2 by adding the same string to ν_2 as was added to ν_1 to get ζ_1. Then ${\zeta_2'}^q = \zeta_1^q$, so $\{e_1\}^{{\zeta_2'}^q}(x) = A$. Ask $\emptyset'$ if there is an extension ζ_2 of ζ_2' such that $\{e_0\}^{\zeta_2^p}(x) \downarrow$. If not, set $\tau = \zeta_2'$; if f extends τ, then $\{e_0\}^{f^p}(x) \uparrow$. If so, take the first such ζ_2. If $\{e_0\}^{\zeta_2^p}(x) \neq A$, set $\tau = \zeta_2$; if f extends τ, then $\{e_0\}^{f^p}(x) = \{e_0\}^{\zeta_2^p}(x) \neq A = \{e_1\}^{{\zeta_2'}^q}(x) = \{e_1\}^{f^q}(x)$. Otherwise, $\{e_0\}^{\zeta_2^p}(x) = A = \{e_1\}^{\zeta_2^q}(x)$. Let ζ_3' be obtained from ν_3 by adding the same string that was added to ν_2 to get ζ_2. Then ${\zeta_3'}^p = \zeta_2^p$, so $\{e_0\}^{{\zeta_3'}^p}(x) = A$, but $\{e_1\}^{{\zeta_3'}^q}(x) = \{e\}^{\nu_3^q}(x) = B$. Set $\tau = \zeta_3'$; clearly, in this last case, we have met the requirement.

The entire process for meeting the requirement is computable from $\emptyset'$, so the proof is complete. ∎

The above proof shows how each of the conditions in the definition of lattice representation is used and also shows how most of the additional conditions (5)-(9) are used. One condition which was not utilized is (9). This condition is used in arguments where one is trying to preserve greatest element. In the notation of the previous proof, the condition ensures that $f^{1_{\mathcal{L}}} \equiv_T f$. (We have $f^{1_{\mathcal{L}}} \leq_T f$ from the equation $f^{1_{\mathcal{L}}}(x) = \theta_{f(x)}(1_{\mathcal{L}})$. Conversely, given x, use an oracle for $f^{1_{\mathcal{L}}}$ to find an i with $\theta_i(1_{\mathcal{L}}) = f^{1_{\mathcal{L}}}(x) = \theta_{f(x)}(1_{\mathcal{L}})$. Then, by (1), $\theta_i = \theta_{f(x)}$, so by (9), $i = f(x)$.) Thus, if one codes a set A into f, then A is in fact coded into the image of $1_{\mathcal{L}}$ under the embedding. For an example of the use of lattice representations to embed lattices preserving greatest element, see [2], where it is shown that every computably presented lattice can be embedded preserving both least and greatest element into $\mathcal{D}(\leq \mathbf{0}')$ (or in fact, into $\mathcal{D}(\leq \mathbf{a})$ for any non-GL_2 degree $\mathbf{a}$).

References

[1] Garrett Birkhoff, *On the structure of abstract algebras*, Math. Proc. Cambridge Philos. Soc. **31** (1935), 433–454.

[2] Peter A. Fejer, *Embedding lattices with top preserved below non-GL_2 degrees*, Z. Math. Logik Grundlag. Math. **35** (1989), 3–14.

[3] Peter A. Fejer and Richard A. Shore, *Embeddings and extensions of embeddings in the r.e. tt and wtt-degrees*, Recursion Theory Week, Proceedings of a Conference Held in Oberwolfach, West Germany (April 15-21, 1984) (Berlin) (H. D. Ebbinghaus, G. H. Müller, and G. E. Sacks, eds.), Lecture Notes in Mathematics, Number 1141, Springer-Verlag, 1985, pp. 121–140.

[4] Christine Ann Haught, *Lattice embeddings in the recursively enumerable truth table degrees*, Trans. Amer. Math. Soc. **301** (1987), 515–535.

[5] Bjarni Jónsson, *On the representation of lattices*, Math. Scand. **1** (1953), 193–206.

[6] ———, *Modular lattices and Desargues theorem*, Math. Scand. **2** (1954), 295–314.

[7] ———, *Representations of modular lattices and relation algebras*, Trans. Amer. Math. Soc. **92** (1959), 449–464.

[8] William A. Lampe, *A perspective on algebriac representations*, Algebra Universalis **31** (1994), 337–364.

[9] Manuel Lerman, *Some nondistributive lattices as initial segments of the degrees of unsolvability*, J. Symbolic Logic **34** (1969), 85–98.

[10] ———, *Initial segments of the degrees of unsolvability*, Ann. of Math. (2) **93** (1971), 365–389.

[11] ———, *Degrees of unsolvability*, Perspectives in Mathematical Logic, Ω-Series, Springer-Verlag, Berlin, 1983.

[12] Pavel Pudlák and Jiří Tůma, *Every finite lattice can be embedded in a finite partition lattice*, Algebra Universalis **10** (1980), 74–95.

[13] Richard A. Shore, *Finitely generated codings and the degrees r.e. in a degree* **d**, Proc. Amer. Math. Soc. **84** (1982), 256–263.

[14] Richard A. Shore and Theodore A. Slaman, *The p-T-degrees of the recursive sets: Lattice embeddings, extensions of embeddings and the two-quantifier theory*, Theoret. Comput. Sci. **97** (1992), 263–284.

[15] S. K. Thomason, *A proof of Whitman's representation theorem for finite lattices*, Proc. Amer. Math. Soc. **25** (1970), 618–619.

[16] ———, *Sublattices and initial segments of the degrees of unsolvability*, Canad. J. Math. **22** (1970), 569–581.

[17] Philip M. Whitman, *Lattices, equivalence relations, and subgroups*, Bull. Amer. Math. Soc. **52** (1946), 507–522.