



**HAL**  
open science

# Computing Distributed Knowledge as the Greatest Lower Bound of Knowledge

Carlos Pinzón, Santiago Quintero, Sergio Ramírez, Frank D. Valencia

► **To cite this version:**

Carlos Pinzón, Santiago Quintero, Sergio Ramírez, Frank D. Valencia. Computing Distributed Knowledge as the Greatest Lower Bound of Knowledge. Relational and Algebraic Methods in Computer Science - 19th International Conference, RAMiCS 2021, Nov 2021, Marseille, France. pp.413-432. hal-02422624v3

**HAL Id: hal-02422624**

**<https://hal.science/hal-02422624v3>**

Submitted on 19 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Computing Distributed Knowledge as the Greatest Lower Bound of Knowledge <sup>\*</sup>

Carlos Pinzón<sup>4</sup>, Santiago Quintero<sup>3</sup>, Sergio Ramírez<sup>4</sup>, Frank Valencia<sup>1,2</sup>

<sup>1</sup> CNRS-LIX, École Polytechnique de Paris

<sup>2</sup> Pontificia Universidad Javeriana-Cali

<sup>3</sup> LIX, École Polytechnique de Paris

<sup>4</sup> INRIA Saclay

**Abstract.** Let  $L$  be a distributive lattice and  $\mathcal{E}(L)$  be the set of join endomorphisms of  $L$ . We consider the problem of finding  $f \sqcap_{\mathcal{E}(L)} g$  given  $L$  and  $f, g \in \mathcal{E}(L)$  as inputs. (1) We show that it can be solved in time  $O(n)$  where  $n = |L|$ . The previous upper bound was  $O(n^2)$ . (2) We characterize the standard notion of distributed knowledge of a group as the greatest lower bound of the join-endomorphisms representing the knowledge of each member of the group. (3) We show that deciding whether an agent has the distributed knowledge of two other agents can be computed in time  $O(n^2)$  where  $n$  is the size of the underlying set of states. (4) For the special case of  $S5$  knowledge, we show that it can be decided in time  $O(n\alpha_n)$  where  $\alpha_n$  is the inverse of the Ackermann function.

**Keywords:** Distributive Knowledge · Join-endomorphisms · Lattice Algorithms.

## Introduction

Structures involving a lattice  $L$  and its set of join-endomorphisms  $\mathcal{E}(L)$  are ubiquitous in computer science. For example, in *Mathematical Morphology (MM)* [3], a well-established theory for the analysis and processing of geometrical structures founded upon lattice theory, join-endomorphisms correspond to one of its fundamental operations: *dilations*. In this and many other areas, lattices are used as rich abstract structures that capture the fundamental principles of their domain of application.

We believe that devising efficient algorithms in the abstract realm of lattice theory could be of great utility: We may benefit from many representation results and identify general properties that can be exploited in the particular domain of application of the corresponding lattices. In fact, we will use *distributivity* and *join-irreducibility* to reduce significantly the time and space needed to solve particular lattice problems. In this paper we focus on algorithms for the meet of join-endomorphisms.

We shall begin with a maximization problem: *Given a lattice  $L$  of size  $n$  and  $f, g \in \mathcal{E}(L)$ , find the greatest lower bound  $h = f \sqcap_{\mathcal{E}(L)} g$ .* Notice that the input is  $L$  not  $\mathcal{E}(L)$ . Simply taking  $h(a) = f(a) \sqcap_L g(a)$  for all  $a \in L$  does not work because the resulting  $h$  may not even be a join-endomorphism. Previous lower bounds for solving this problem are  $O(n^3)$  for arbitrary lattices and  $O(n^2)$  for distributive lattices [22]. We will show that that this problem can actually be solved in  $O(n)$  for distributive lattices.

---

<sup>\*</sup> This work has been partially supported by the ECOS-NORD project FACTS (C19M03).

*Distributed knowledge* [15] corresponds to knowledge that is distributed among the members of a group, without any of its members necessarily having it. This notion can be used to analyse the implications of the knowledge of a community if its members were to combine their knowledge, hence its importance. We will show that distributed knowledge can be seen as the meet of the join-endomorphisms representing the knowledge of each member of a group.

The standard structures in economics for multi-agent knowledge [23] involve a set of *states (or worlds)*  $\Omega$  and a *knowledge operator*  $K_i : \mathcal{P}(\Omega) \rightarrow \mathcal{P}(\Omega)$  describing the events, represented as subsets of  $\Omega$ , that an agent  $i$  knows. The event of  $i$  knowing the event  $E$  is  $K_i(E) = \{\omega \in \Omega \mid \mathcal{R}_i(\omega) \subseteq E\}$  where  $\mathcal{R}_i \subseteq \Omega^2$  is the accessibility relation of  $i$  and  $\mathcal{R}_i(\omega) = \{\omega' \mid (\omega, \omega') \in \mathcal{R}_i\}$ . The event of having distributed knowledge of  $E$  by  $i$  and  $j$  is  $D_{\{i,j\}}(E) = \{\omega \in \Omega \mid \mathcal{R}_i(\omega) \cap \mathcal{R}_j(\omega) \subseteq E\}$  [7].

Knowledge operators are join-endomorphisms of  $L = (\mathcal{P}(\Omega), \supseteq)$ . Intuitively, the lower an agent  $i$  (its knowledge function) is placed in  $\mathcal{E}(L)$ , the “wiser” (or more knowledgeable) the agent is. We will show that  $D_{\{i,j\}} = K_i \sqcap_{\mathcal{E}(L)} K_j$ , i.e.,  $D_{\{i,j\}}$  can be viewed as the *least* knowledgeable agent that is *wiser* than both  $i$  and  $j$ .

We also consider the following decision problem: *Given the knowledge of agents  $i, j, m$ , decide whether  $m$  has the distributed knowledge of  $i$  and  $j$ , i.e.,  $K_m = D_{\{i,j\}}$ .* The knowledge of an agent  $k$  can be represented by  $K_k : \mathcal{P}(\Omega) \rightarrow \mathcal{P}(\Omega)$ . If available it can also be represented, exponentially more succinctly, by  $\mathcal{R}_k \subseteq \Omega^2$ . In the first case the problem reduces to checking whether  $K_m = K_i \sqcap_{\mathcal{E}(L)} K_j$ . In the second the problem reduces to  $\mathcal{R}_m = \mathcal{R}_i \cap \mathcal{R}_j$  and this can be done in  $O(n^2)$  where  $n = |\Omega|$ .

Nevertheless, we show that even without the accessibility relations, if inputs are the knowledge operators, represented as arrays, the problem can be still be solved in  $O(n^2)$ . We obtain this result using tools from lattice theory to exponentially reduce the number of tests on the knowledge operators (arrays) needed to decide the problem.

Furthermore, if the inputs are the accessibility relations and they are equivalences (hence they can be represented as partitions), we show that the problem can be solved basically in *linear time*: More precisely, in  $O(n\alpha_n)$  where  $\alpha_n$  is an extremely slow growing function; the inverse of the Ackermann function. It is worth noticing that if accessibility relations can be represented as partitions, the structures are known as Aumann structures [2] and they characterize a standard notion of knowledge called *S5* [7].

To prove the  $O(n\alpha_n)$  bound we show a new result of independent interest using a Disjoint-Set data structure [8]: The intersection of two *partitions* of a set of size  $n$  can be computed in  $O(n\alpha_n)$ . This result may have applications beyond *knowledge*, particularly in domains where Disjoint-Set is typically used; e.g., *given two undirected graphs  $G_1$  and  $G_2$  with the same nodes, find an undirected graph  $G_3$  such that two nodes are connected in it iff they are connected in both  $G_1$  and  $G_2$ .*

**Contributions and Organization.** The main contributions are the following:

1. We prove that for distributive lattices of size  $n$ , the meet of join-endomorphisms can be computed in time  $O(n)$ . Previous upper bound was  $O(n^2)$ .
2. We show that distributed knowledge of a given group can be viewed as the meet of the join-endomorphisms representing the knowledge of each member of the group.
3. We show that the problem of whether an agent has the distributed knowledge of two other can be decided in time  $O(n^2)$  where  $n = |\Omega|$ .

4. If the agents' knowledge can be represented as partitions, the problem in (3) can be decided in  $O(n\alpha_n)$ . To obtain this we provide a procedure, interesting in its own right, that computes the intersection of two *partitions* of a set of size  $n$  in  $O(n\alpha_n)$ .

The above results are given in Sections 2 and 5. For conducting our study, in the intermediate sections (Sections 3 and 4) we will adapt some representation and duality results (e.g., Jónsson-Tarski duality [17]) to our structures. Some of these results are part of the folklore in lattice theory but for completeness we provide simple proofs of them. We also provide experimental results for the above-mentioned effective procedures.

## 1 Notation, Definitions and Elementary Facts

We list facts and notation used throughout the paper. We index joins, meets, and orders with their corresponding poset but often omit the index when it is clear from the context.

**Partially Ordered Sets and Lattices.** A poset  $L$  is a *lattice* iff each finite nonempty subset of  $L$  has a supremum and infimum in  $L$ . It is a *complete lattice* iff each subset of  $L$  has a supremum and infimum in  $L$ . A poset  $L$  is *distributive* iff for every  $a, b, c \in L$ ,  $a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$ . We write  $a \parallel b$  to denote that  $a$  and  $b$  are incomparable in the underlying poset. A *lattice of sets* is a set of sets ordered by inclusion and closed under finite unions and intersections. A *powerset lattice* is a lattice of sets that includes all the subsets of its top element.

**Definition 1 (Downsets, Covers, Join-irreducibility [5]).** Let  $L$  be a lattice and  $a, b \in L$ . We say  $b$  is covered by  $a$ , written  $b \prec a$ , if  $b \sqsubset a$  and there is no  $c \in L$  s.t.,  $b \sqsubset c \sqsubset a$ . The down-set (up-set) of  $a$  is  $\downarrow a \stackrel{\text{def}}{=} \{b \in L \mid b \sqsubseteq a\}$  ( $\uparrow a \stackrel{\text{def}}{=} \{b \in L \mid b \sqsupseteq a\}$ ), and the set of elements covered by  $a$  is  $\downarrow^1 a \stackrel{\text{def}}{=} \{b \mid b \prec a\}$ . An element  $c \in L$  is said to be join-irreducible if  $c = a \sqcup b$  implies  $c = a$  or  $c = b$ . If  $L$  is finite,  $c$  is join-irreducible if  $|\downarrow^1 c| = 1$ . The set of all join-irreducible elements of  $L$  is  $\mathcal{J}(L)$  and  $\downarrow^{\mathcal{J}} c \stackrel{\text{def}}{=} \downarrow c \cap \mathcal{J}(L)$ .

**Posets of maps.** A map  $f : X \rightarrow Y$  where  $X$  and  $Y$  are posets is *monotonic* (or *order-preserving*) if  $a \sqsubseteq_X b$  implies  $f(a) \sqsubseteq_Y f(b)$  for every  $a, b \in X$ . We say that  $f$  preserves the join of  $S \subseteq X$  iff  $f(\bigsqcup S) = \bigsqcup \{f(c) \mid c \in S\}$ . A *self-map* on  $X$  is a function  $f : X \rightarrow X$ . If  $X$  and  $Y$  are posets, we use  $\langle X \rightarrow Y \rangle$  to denote the poset of monotonic functions from  $X$  to  $Y$ . The functions in  $M = \langle X \rightarrow Y \rangle$  are ordered pointwise: i.e.,  $f \sqsubseteq_M g$  iff  $f(a) \sqsubseteq_Y g(a)$  for every  $a \in X$ .

**Definition 2 (Join-endomorphisms and  $\mathcal{E}(L)$ ).** Let  $L$  be a lattice. We say that a self-map is a (bottom preserving) join-endomorphism iff it preserves the join of every finite subset of  $L$ . Define  $\mathcal{E}(L)$  as the set of all join-endomorphisms of  $L$ . Furthermore, given  $f, g \in \mathcal{E}(L)$ , define  $f \sqsubseteq_{\mathcal{E}} g$  iff  $f(a) \sqsubseteq g(a)$  for every  $a \in L$ .

**Proposition 3 ([12,5]).** Let  $L$  be a lattice.

- P.1**  $f \in \mathcal{E}(L)$  iff  $f(\perp) = \perp$  and  $f(a \sqcup b) = f(a) \sqcup f(b)$  for all  $a, b \in L$ .  
**P.2** If  $f \in \mathcal{E}(L)$  then  $f$  is monotonic.  
**P.3** If  $L$  is a complete lattice, then  $\mathcal{E}(L)$  is a complete lattice.

**P.4**  $\mathcal{E}(L)$  is a complete distributive lattice iff  $L$  is a complete distributive lattice.

**P.5** If  $L$  is finite and distributive,  $\mathcal{E}(L) \cong \langle \mathcal{J}(L) \rightarrow L \rangle$ .

**P.6** If  $L$  is a finite lattice,  $e = \bigsqcup_L \{c \in \mathcal{J}(L) \mid c \sqsubseteq e\}$  for every  $e \in L$ .

**P.7** If  $L$  is finite and distributive,  $f \in \mathcal{E}(L)$  iff  $(\forall e \in L) f(e) = \bigsqcup \{f(e') \mid e' \in \downarrow^\mathcal{J} e\}$ .

We shall use these posets in our examples:  $\bar{n}$  is  $\{1, \dots, n\}$  with the order  $x \sqsubseteq y$  iff  $x = y$  and  $\mathbf{M}_n \stackrel{\text{def}}{=} (\bar{n}_\perp)^\top$  is the lattice that results from adding a top and bottom to  $\bar{n}$ .

## 2 Computing the Meet of Join-Endomorphisms

Join-endomorphisms and their meet arise as fundamental computational operations in computer science. We therefore believe that the problem of computing these operations in the abstract realm of lattice theory is a relevant issue: We may identify general properties that can be exploited in all instances of these lattices.

In this section, we address the problem of computing the meet of join endomorphisms. Let us consider the following *maximization* problem.

*Problem 4.* Given a lattice  $L$  of size  $n$  and two join-endomorphisms  $f, g : L \rightarrow L$ , find the *greatest* join-endomorphism  $h : L \rightarrow L$  below both  $f$  and  $g$ : i.e.,  $h = f \sqcap_{\mathcal{E}(L)} g$ .

Notice that the lattice  $\mathcal{E}(L)$ , which could be exponentially bigger than  $L$  [22], is not an input to the problem above. It may not be immediate how to find  $h$ ; e.g., see the endomorphism  $h$  in Fig. 1a for a small lattice of four elements. A *naive approach* to find  $f \sqcap_{\mathcal{E}(L)} g$  could be to attempt to compute it pointwise by taking  $h(a) = f(a) \sqcap_L g(a)$  for every  $a \in L$ . Nevertheless, the somewhat appealing equation

$$(f \sqcap_{\mathcal{E}(L)} g)(a) = f(a) \sqcap_L g(a) \tag{1}$$

does not hold in general, as illustrated in the lattices  $\mathbf{M}_2$  and  $\mathbf{M}_3$  in Fig. 1b and Fig. 1c.

A general approach in [22] for arbitrary lattices shows how to find  $h$  in Prob.4 by successive approximations  $\sigma^0 \sqsupseteq \sigma^1 \sqsupseteq \dots \sqsupseteq \sigma^i$ , starting with some self-map  $\sigma^0$  known to be smaller than both  $f$  and  $g$ , and greater than  $h$ ; while keeping the invariant  $\sigma^i \sqsupseteq h$ . The starting point is the naive approach above:  $\sigma^0(a) = f(a) \sqcap g(a)$  for all  $a \in L$ . The approach computes decreasing upper bounds of  $h$  by correcting in  $\sigma^i$  the image under  $\sigma^{i-1}$  of some values  $b, c, b \sqcup c$  violating the property  $\sigma^{i-1}(b) \sqcup \sigma^{i-1}(c) = \sigma^{i-1}(b \sqcup c)$ . The correction satisfies  $\sigma^{i-1} \sqsupseteq \sigma^i$  and maintains the invariant  $\sigma^i \sqsupseteq h$ . This approach eventually finds  $h$  in  $O(n^3)$  basic lattice operations (binary meets and joins).

Recall that in finite distributive lattices, and more generally in co-Heyting algebras [21], the subtraction operator  $\ominus$  is uniquely determined by the *Galois connection*  $b \sqsupseteq c \ominus a$  iff  $a \sqcup b \sqsupseteq c$ . Based on the following proposition it was shown in [22] that if the only basic operations are joins or meets,  $h$  can be computed in  $O(n^3)$  of them, but if we also allow subtraction as a basic operation, the bound can be improved to  $O(n^2)$ .

**Proposition 5 ([22]).** *Let  $L$  be a finite distributive lattice. Let  $h = f \sqcap_{\mathcal{E}(L)} g$ . Then (1)  $h(c) = \bigsqcup_L \{f(a) \sqcup g(b) \mid a \sqcup b \sqsupseteq c\}$ , and (2)  $h(c) = \bigsqcup_L \{f(a) \sqcup g(c \ominus a) \mid a \in \downarrow c\}$ .*

Nevertheless, it turns out that we can partly use Eq. 1 to obtain a better upper bound. The following lemma states that Eq. 1 holds if  $L$  is distributive and  $a \in \mathcal{J}(L)$ .

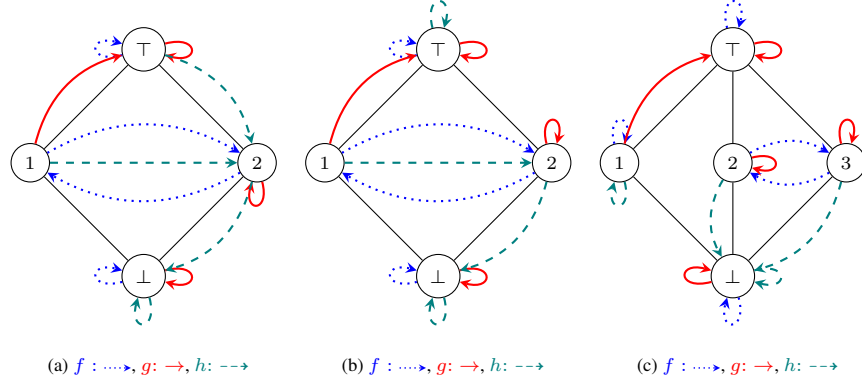


Fig. 1: (a)  $h = f \sqcap_{\mathcal{E}(L)} g$ . (b)  $h(a) \stackrel{\text{def}}{=} f(a) \sqcap g(a)$  for  $a \in \mathbf{M}_2$  is not in  $\mathcal{E}(\mathbf{M}_2)$ :  $h(1 \sqcup 2) \neq h(1) \sqcup h(2)$ . (c) Any  $h : \mathbf{M}_3 \rightarrow \mathbf{M}_3$  s.t.  $h(a) = f(a) \sqcap g(a)$  for  $a \in \mathcal{J}(\mathbf{M}_3)$  is not in  $\mathcal{E}(\mathbf{M}_3)$ :  $h(\top) = h(1 \sqcup 2) = h(1) \sqcup h(2) = 1 \neq \perp = h(2) \sqcup h(3) = h(2 \sqcup 3) = h(\top)$ .

**Lemma 6.** Let  $L$  be a finite distributive lattice and  $f, g \in \mathcal{E}(L)$ . Then the following equation holds:  $(f \sqcap_{\mathcal{E}(L)} g)(a) = f(a) \sqcap_L g(a)$  for every  $a \in \mathcal{J}(L)$ .

*Proof.* From Prop.5,  $(f \sqcap_{\mathcal{E}(L)} g)(a) = \sqcap \{f(a') \sqcup g(a \ominus a') \mid a' \in \downarrow a\}$ . Note that since  $a \in \mathcal{J}(L)$  if  $a' \in \downarrow a$  then  $a \ominus a' = a$  when  $a \neq a'$ , and  $a \ominus a' = \perp$  when  $a = a'$ . Then,  $\{f(a') \sqcup g(a \ominus a') \mid a' \in \downarrow a\} = \{f(a') \sqcup g(a \ominus a') \mid a' \sqsubset a\} \cup \{f(a) \sqcup g(\perp)\} = \{f(a') \sqcup g(a) \mid a' \sqsubset a\} \cup \{f(a)\} = \{f(a') \sqcup g(a) \mid \perp \sqsubset a' \sqsubset a\} \cup \{f(a), g(a)\}$ . By absorption, we know that  $(f(a') \sqcup g(a)) \sqcap g(a) = g(a)$ . Finally, using properties of  $\sqcap$ ,  $(f \sqcap_{\mathcal{E}(L)} g)(a) = \sqcap (\{f(a') \sqcup g(a) \mid \perp \sqsubset a' \sqsubset a\} \cup \{f(a), g(a)\}) = \sqcap \{f(a') \sqcup g(a) \mid \perp \sqsubset a' \sqsubset a\} \sqcap f(a) \sqcap g(a) = f(a) \sqcap g(a)$ .  $\square$

It is worth noting the Lem.6 may not hold for non-distributive lattices. This is illustrated in Fig.1c with the archetypal non-distributive lattice  $\mathbf{M}_3$ . Suppose that  $f$  and  $g$  are given as in Fig.1c. Let  $h = f \sqcap_{\mathcal{E}(L)} g$  with  $h(a) = f(a) \sqcap g(a)$  for all  $a \in \{1, 2, 3\} = \mathcal{J}(\mathbf{M}_3)$ . Since  $h$  is a join-endomorphism, we would have  $h(\top) = h(1 \sqcup 2) = h(1) \sqcup h(2) = 1 \neq \perp = h(2) \sqcup h(3) = h(2 \sqcup 3) = h(\top)$ , a contradiction.

Lem.6 and P.7 lead us to the following characterization of meets over  $\mathcal{E}(L)$ .

**Theorem 7.** Let  $L$  be a finite distributive lattice and  $f, g \in \mathcal{E}(L)$ . Then  $h = f \sqcap_{\mathcal{E}(L)} g$  iff  $h$  satisfies

$$h(a) = \begin{cases} f(a) \sqcap_L g(a) & \text{if } a \in \mathcal{J}(L) \text{ or } a = \perp \\ h(b) \sqcup_L h(c) & \text{if } b, c \in \downarrow^1 a \text{ with } b \neq c \end{cases} \quad (2)$$

*Proof.* The only-if direction follows from Lem.6 and Prop.P.7. For the if-direction, suppose that  $h$  satisfies Eq.2. If  $h \in \mathcal{E}(L)$  the result follows from Lem.6 and P.7. To prove  $h \in \mathcal{E}(L)$  from P.7 it suffices to show (3)  $h(e) = \sqcup \{h(e') \mid e' \in \downarrow^{\mathcal{J}} e\}$  for every  $e \in L$ . From Eq.2 and since  $f$  and  $g$  are monotonic,  $h$  is monotonic. If  $e \in \mathcal{J}(L)$  then  $h(e') \sqsubseteq h(e)$  for every  $e' \in \downarrow^{\mathcal{J}} e$ . Therefore,  $\sqcup \{h(e') \mid e' \in \downarrow^{\mathcal{J}} e\} = h(e)$ . If

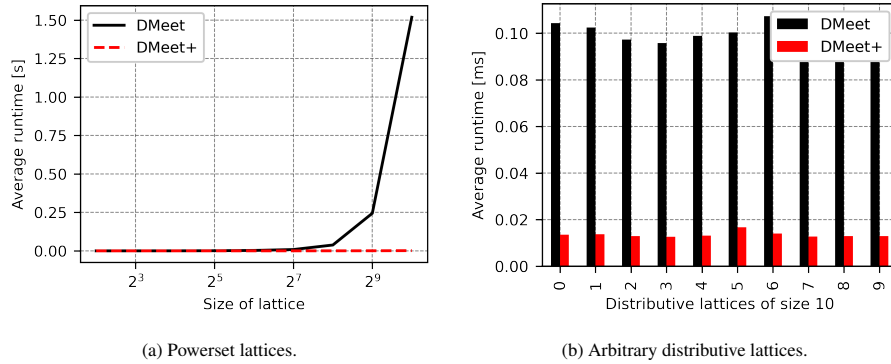


Fig. 2: Comparison between an implementation of Prop.5 (DMEET) and Th.7 (DMEET+).

$e \notin \mathcal{J}(L)$ , we proceed by induction. Assume Eq.3 holds for all  $a \in \downarrow^1 e$ . By definition,  $h(e) = h(b) \sqcup h(c)$  for any  $b, c \in \downarrow^1 e$  with  $b \neq c$ . Then, we have  $h(b) = \bigsqcup \{h(e') \mid e' \in \downarrow^\sigma b\}$  and  $h(c) = \bigsqcup \{h(e') \mid e' \in \downarrow^\sigma c\}$ . Notice that  $e' \in \downarrow^\sigma b$  or  $e' \in \downarrow^\sigma c$  iff  $e' \in \downarrow^\sigma (b \sqcup c)$ , since  $L$  is distributive. Thus,  $h(e) = h(b) \sqcup h(c) = \bigsqcup \{h(e') \mid e' \in \downarrow^\sigma (b \sqcup c)\} = \bigsqcup \{h(e') \mid e' \in \downarrow^\sigma e\}$  as wanted.  $\square$

We conclude this section by stating the time complexity  $O(n)$  to compute  $h$  in the above theorem. As in [22], the time complexity is determined by the number of basic binary lattice operations (i.e., meets and joins) performed during execution.

**Corollary 8.** *Given a distributive lattice  $L$  of size  $n$ , and functions  $f, g \in \mathcal{E}(L)$ , the function  $h = f \sqcap_{\mathcal{E}(L)} g$  can be computed in  $O(n)$  binary lattice operations.*

*Proof.* If  $a \in \mathcal{J}(L)$  then from Th.7,  $h(a)$  can be computed as  $f(a) \sqcap g(a)$ . If  $a = \perp$  then  $h(a)$  is  $\perp$ . If  $a \notin \mathcal{J}(L)$  and  $a \neq \perp$ , we pick any  $b, c \in \downarrow^1 a$  such that  $b \neq c$  and compute  $h(a)$  recursively as  $h(b) \sqcup h(c)$  by Th.7. We can use a lookup table to keep track of the values of  $a \in L$  for which  $h(a)$  has been computed, starting with all  $a \in \mathcal{J}(L)$ . Since  $h(a)$  is only computed once for each  $a \in L$ , either as a meet for elements in  $\mathcal{J}(L)$  or as a join otherwise, we only perform  $n$  binary lattice operations.

**Experimental Results.** Now we present some experimental results comparing the average runtime between the previous algorithm in [22] based on Prop.5, referred to as DMEET, and the proposed algorithm in Th.7, called DMEET+.

Fig.2 shows the average runtime of each algorithm, from 100 runs with a random pair of join-endomorphisms. For Fig.2a, we compared each algorithm against powerset lattices of sizes between  $2^2$  and  $2^{10}$ . For Fig.2b, 10 random distributive lattices of size 10 were selected. In both cases, all binary lattice operation are guaranteed a complexity in  $O(1)$  to showcase the quadratic nature of DMEET compared to the linear growth of DMEET+. The time reduction from DMEET to DMEET+ is also reflected in a reduction on the number of  $\sqcup$  and  $\sqcap$  operations performed as illustrated in Table 1. For DMEET+, given a distributive lattice  $L$  of size  $n$ ,  $\#\sqcap = |\mathcal{J}(L)|$  and  $\#\sqcup = |L| - |\mathcal{J}(L)| - 1$  ( $\perp$  is directly mapped to  $\perp$ ).

Size	DMEET Time [s]	DMEET+ Time [s]	DMEET # $\sqcup$	DMEET+ # $\sqcup$	DMEET # $\sqcap$	DMEET+ # $\sqcap$
16	0.000246	0.000024	81	11	81	4
32	0.000971	0.000059	243	26	243	5
64	0.002659	0.000094	729	57	729	6
128	0.008735	0.000163	2187	120	2187	7
256	0.038086	0.000302	6561	247	6561	8
512	0.244304	0.000645	19683	502	19683	9
1024	1.518173	0.001468	59049	1013	59049	10

Table 1: Average runtime in seconds over powerset lattices. Number of  $\sqcup$  and  $\sqcap$  operations performed for each algorithm.

### 3 A Representation of Join-Irreducible Elements of $\mathcal{E}(L)$

In this section we state a characterization of the join-irreducible elements of the lattice of join-endomorphisms  $\mathcal{E}(L)$ . We use it to prove a representation result for join-endomorphisms. Some of these results may be part of the folklore in lattice theory, our purpose here is to identify and use them as technical tools in the following section.

The following family of functions can be used to represent  $\mathcal{J}(\mathcal{E}(L))$ .

**Definition 9.** Let  $L$  be a lattice and  $a, b \in \mathcal{J}(L)$ . Let  $f_{a,b} : L \rightarrow L$  be given by  $f_{a,b}(x) \stackrel{\text{def}}{=} b$  if  $x \in \uparrow a$ , otherwise  $f_{a,b}(x) \stackrel{\text{def}}{=} \perp$ .

It is easy to verify that  $f_{a,b}(\perp) = \perp$ . On the other hand, for every  $c, d \in L$ ,  $f_{a,b}(c \sqcup d) = f_{a,b}(c) \sqcup f_{a,b}(d)$  follows from the fact that  $a \in \mathcal{J}(L)$  and by cases on  $c \sqcup d \in \uparrow a$  and  $c \sqcup d \notin \uparrow a$ . Thus, from P.1 we know that  $f_{a,b}$  is a join-endomorphism, and from P.2 it is monotone. Therefore,  $f_{a,b} \upharpoonright_{\mathcal{J}(L)} \in \langle \mathcal{J}(L) \rightarrow L \rangle$ . In addition, we point out the following rather technical lemma that gives us way to construct from a function  $g \in \langle \mathcal{J}(L) \rightarrow L \rangle$ , a function  $h \in \langle \mathcal{J}(L) \rightarrow L \rangle$  covered by  $g$ . The proof is given in the technical report available at <https://hal.archives-ouvertes.fr/hal-03323638>.

**Lemma 10.** Let  $L$  be a finite lattice. Let  $g \in \langle \mathcal{J}(L) \rightarrow L \rangle$ ,  $x_0 \in \mathcal{J}(L)$  and  $y_0 \in L$  be such that  $y_0 \in \downarrow^1 g(x_0)$  and  $g(x) \sqsubseteq y_0$  for all  $x \sqsubset x_0$ . Define  $h : \mathcal{J}(L) \rightarrow L$  as  $h(x) \stackrel{\text{def}}{=} y_0$  if  $x = x_0$  else  $h(x) \stackrel{\text{def}}{=} g(x)$ . Then  $h$  is monotonic and  $g$  covers  $h$ .

We proceed to characterize the join-irreducible elements of the lattice  $\mathcal{E}(L)$ . The next lemma, together with P.6, tell us that every join-endomorphism in  $\mathcal{E}(L)$  can be expressed solely as a join of functions of the form  $f_{a,b}$  defined in Def.9.

**Lemma 11.** Let  $L$  be a finite distributive lattice. For any join-endomorphism  $f \in \mathcal{E}(L)$ ,  $f$  is join-irreducible iff  $f = f_{a,b}$  for some  $a, b \in \mathcal{J}(L)$ .

*Proof.* For notational convenience let  $M = \langle \mathcal{J}(L) \rightarrow L \rangle$ . From P.5 it suffices to prove:  $g \in M$  is join-irreducible in  $M$  iff  $g = g_{a,b}$  for some  $a, b \in \mathcal{J}(L)$  where  $g_{a,b} = f_{a,b} \upharpoonright_{\mathcal{J}(L)}$ . We use the following immediate consequence of Lem.10.



**Property ( $\star$ ):** Let  $g \in M$ ,  $x_1, x_2 \in \mathcal{J}(L)$  and  $y_1, y_2 \in L$ , be such that for each  $i \in \{1, 2\}$ ,  $y_i \in \downarrow^1 g(x_i)$  and  $g(x) \sqsubseteq y_i$  for all  $x \sqsubset x_i$ . If  $x_1 \neq x_2$  or  $y_1 \neq y_2$ , then there are two distinct functions  $g_1, g_2 \in M$  that are covered by  $g$  in  $M$ .

1. For the only-if direction, let  $X = \{x \in \mathcal{J}(L) \mid g(x) \neq \perp\}$  and  $Y = \{g(x) \mid x \in X\}$ . If  $X = \emptyset$ , then  $g(x) = \perp$  for all  $x \in \mathcal{J}(L)$ , in which case  $g$  is not join-irreducible in  $M$ . Thus, necessarily,  $X \neq \emptyset$  and  $Y \neq \emptyset$ . Let us now prove that:
  - (a)  $X$  has a minimum element  $a \in \mathcal{J}(L)$  with  $g(a) \in \mathcal{J}(L)$ , and (b)  $Y = \{g(a)\}$ .
  - (a) Let  $x_1, x_2 \in X$  be minimal elements in  $X$ . For each  $i \in \{1, 2\}$ , let  $y_i \in \downarrow^1 g(x_i)$ . Since  $x_i$  is minimal, it follows that  $g(x) = \perp$  for all  $x \sqsubset x_i$ . From ( $\star$ ) and the fact that  $g$  is join-irreducible, we have  $x_1 = x_2$  and  $y_1 = y_2$ . Thus,  $X$  has a minimum element. We refer to such element as  $a$ . Furthermore,  $|\downarrow^1 g(a)| = 1$ , i.e.  $g(a) \in \mathcal{J}(L)$ .
  - (b) Let  $Y^* = Y \setminus \{g(a)\}$ . For the sake of contradiction, suppose  $Y^* \neq \emptyset$ . Let  $y \in Y^*$  be a minimal element and  $x^* \in X$  be a minimal of  $X^* = \{x \in X \mid g(x) = y\}$ . Since  $a \sqsubset x^*$  and  $y \neq g(a)$ , we have  $g(a) \sqsubset g(x^*) = y$ . Then there is at least one  $z \in \downarrow^1 y$  such that  $g(a) \sqsubseteq z \sqsubset y$ . Since  $g$  is monotonic,  $Im(g) = \{\perp\} \cup Y$  and  $y$  is minimal in  $Y^*$ , for all  $x \sqsubset x^*$ , we have  $g(x) \in \{\perp, g(a)\}$ . Therefore,  $g(x) \sqsubseteq z$  for all  $x \sqsubset x^*$ . From ( $\star$ ), with  $x_1 = a$ ,  $x_2 = x^*$ ,  $y_1 \in \downarrow^1 g(a)$  and  $y_2 = z$ , it follows that  $g$  is not join-irreducible in  $M$ , a contradiction.

Monotonicity of  $g$  and (a)-(b), imply  $Im(g) = \{\perp, b\}$  with  $b = g(a)$ . Thus  $g = g_{a,b}$ .

2. We prove that  $g = g_{a,b}$  has a unique cover in  $M$ . Let  $c$  be the only cover of  $b$ . Define  $g^*: \mathcal{J}(L) \rightarrow L$  as  $g^*(x) = c$  if  $x = a$  else  $g^*(x) = g(x)$ . From Lem.10, it follows that  $g^* \in M$  and  $g_{a,b}$  covers  $g^*$  in  $M$ . It suffices that for any  $h \in M$  with  $h \sqsubset_M g_{a,b}$ ,  $h \sqsubseteq_M g^*$  holds. Take any such  $h \in M$ . Since  $h(a) \neq b$ ,  $h(a) \sqsubset b$ . Thus  $h(a) \sqsubseteq c$ , so  $h(a) \sqsubseteq g^*(a)$ . Indeed, for any  $x \neq a$ ,  $h(x) \sqsubset g(x) = g^*(x)$ . Then  $h \sqsubseteq_M g^*$ .  $\square$

We conclude with a corollary of Lem.11 that provides a representation theorem for join-endomorphism on distributive lattices. We will use this result in the next section.

**Corollary 12.** *Let  $L$  be a finite distributive lattice and let  $f \in \mathcal{E}(L)$ . Then  $f = F_R$  where  $R = \{(a, b) \in \mathcal{J}(L)^2 \mid a \sqsubseteq f(b)\}$  and  $F_R : L \rightarrow L$  is the function given by  $F_R(c) \stackrel{\text{def}}{=} \bigsqcup \{a \in \mathcal{J}(L) \mid (a, b) \in R \text{ and } c \sqsupseteq b \text{ for some } b \in \mathcal{J}(L)\}$ .*

*Proof.* From P.6  $f = \bigsqcup_{\mathcal{E}(L)} \{g \in \mathcal{J}(\mathcal{E}(L)) \mid g \sqsubseteq_{\mathcal{E}} f\}$ . Thus,

$$\begin{aligned}
 f(c) &= \left( \bigsqcup_{\mathcal{E}(L)} \{g \in \mathcal{J}(\mathcal{E}(L)) \mid g \sqsubseteq_{\mathcal{E}} f\} \right) (c) = \bigsqcup \{g(c) \mid g \in \mathcal{J}(\mathcal{E}(L)) \text{ and } g \sqsubseteq_{\mathcal{E}} f\} \\
 &= \bigsqcup \{f_{b,a}(c) \mid (b, a) \in \mathcal{J}(L)^2 \text{ and } f_{b,a} \sqsubseteq_{\mathcal{E}} f\} && \text{(Lem.11)} \\
 &= \bigsqcup \{f_{b,a}(c) \mid (b, a) \in \mathcal{J}(L)^2 \text{ and } a \sqsubseteq f(b)\} \\
 &= \bigsqcup \{a \in \mathcal{J}(L) \mid (b, a) \in \mathcal{J}(L)^2, a \sqsubseteq f(b) \text{ and } c \sqsupseteq b \text{ for some } b \in \mathcal{J}(L)\} \\
 &= \bigsqcup \{a \in \mathcal{J}(L) \mid (b, a) \in R \text{ and } c \sqsupseteq b \text{ for some } b \in \mathcal{J}(L)\} = F_R(c)
 \end{aligned}$$

## 4 Distributive Lattices and Knowledge Structures

In this section, we introduce some knowledge structures from economics [2,23] and relate them to distributive lattices by adapting fundamental duality results between modal algebras and frames [17]. We will use these structures and their relation to distributive lattices in the algorithmic results in the next section. We use the term *knowledge* to encompass various epistemic concepts including *S5* knowledge and belief [7].

**Definition 13 ([23]).** A (finite) Knowledge Structure (KS) for a set of agents  $\mathcal{A}$  is a tuple  $(\Omega, \{K_i\}_{i \in \mathcal{A}})$  where  $\Omega$  is a finite set and each  $K_i : \mathcal{P}(\Omega) \rightarrow \mathcal{P}(\Omega)$  is given by  $K_i(E) = \{\omega \in \Omega \mid \mathcal{R}_i(\omega) \subseteq E\}$  where  $\mathcal{R}_i \subseteq \Omega^2$  and  $\mathcal{R}_i(\omega) = \{\omega' \mid (\omega, \omega') \in \mathcal{R}_i\}$ .

The elements  $\omega \in \Omega$  and the subsets  $E \subseteq \Omega$  are called *states* and *events*, resp. We refer to  $K_i$  and  $\mathcal{R}_i$  as the *knowledge operator* and the *accessibility relation* of agent  $i$ .

The notion of *event* may be familiar to some readers from probability theory; for example the event “*public transportation is suspended*” corresponds the set of states at which public transportation is suspended. An event  $E$  holds at  $\omega$  if  $\omega \in E$ . Thus  $\Omega$ , the event that holds at every  $\omega$ , corresponds to true in logic, union of events corresponds to disjunction, intersection to conjunction, and complementation in  $\Omega$  to negation. We use  $\bar{E}$  for  $\Omega \setminus E$ . We write  $E \Rightarrow F$  for the event  $\bar{E} \cup F$  which corresponds to classic logic implication. We say that  $E$  entails  $F$  if  $E \subseteq F$ . The event of  $i$  knowing  $E$  is  $K_i(E)$ .

The following properties hold for all events  $E$  and  $F$  of any KS  $(\Omega, \{K_i\}_{i \in \mathcal{A}})$ . It is easy to see that ( $\mathfrak{K}1$ )  $K_i(\Omega) = \Omega$ , i.e., agents know the event that holds at every state, i.e.,  $\Omega$ . A distinctive property of knowledge is ( $\mathfrak{K}2$ )  $K_i(E) \cap K_i(F) = K_i(E \cap F)$ ; i.e., if an agent knows two events, she knows their conjunction. In fact,  $\mathfrak{K}2$  implies ( $\mathfrak{K}3$ )  $(K_i(E) \cap K_i(E \Rightarrow F)) \subseteq K_i(F)$ . This property expresses *modus ponens* for knowledge. Other property implied by  $\mathfrak{K}2$  is that knowledge is monotonic: ( $\mathfrak{K}4$ ) if  $E \subseteq F$  then  $K_i(E) \subseteq K_i(F)$ , i.e., agents know the consequences of their knowledge.

An agent  $i$  is *wiser* (or *more knowledgeable*) than  $j$  iff  $K_j(E) \subseteq K_i(E)$  for every event  $E$ ; i.e., if  $j$  knows  $E$  so does  $i$ .

**Aumann Structures.** Aumann structures are the standard event-based formalism in economics and decision theory [7] for reasoning about knowledge. A (finite) Aumann structure (AS) is a KS where all the accessibility relations are equivalences.<sup>5</sup> The intended notion of knowledge of AS is *S5*; i.e., the knowledge captured by Prop. $\mathfrak{K}1$ - $\mathfrak{K}2$  and the following three fundamental properties which hold for any AS: ( $\mathfrak{K}5$ )  $K_i(E) \subseteq E$ , ( $\mathfrak{K}6$ )  $K_i(E) \subseteq K_i(K_i(E))$ , and ( $\mathfrak{K}7$ )  $\overline{K_i(E)} \subseteq K_i(\overline{K_i(E)})$ . The first says that if an agent knows  $E$  then  $E$  cannot be false, the second and third state that agents know what they know and what they do not know.

**Extended KS.** We now introduce a simple extension of KS that will allow us to give a uniform presentation of our results.

**Definition 14 (EKS).** A tuple  $(\Omega, \mathcal{S}, \{K_i\}_{i \in \mathcal{A}})$  is said to be an extended knowledge structure (EKS) if (1)  $(\Omega, \{K_i\}_{i \in \mathcal{A}})$  is a KS, and (2)  $\mathcal{S}$  is a subset of  $\mathcal{P}(\Omega)$  that contains  $\Omega$  and it is closed under union, intersection and application of  $K_i$  for every  $i \in \mathcal{A}$ .

<sup>5</sup> The presentation of AS [2] uses a partition  $\mathcal{P}_i = \{\mathcal{R}_i(\omega) \mid \omega \in \Omega\}$  of  $\Omega$  and  $K_i(E)$  is equivalently defined as  $\{\omega \in \Omega \mid \mathcal{P}_i(\omega) \subseteq E\}$  where  $\mathcal{P}_i(\omega)$  is the cell of  $\mathcal{P}_i$  containing  $\omega$ .

*Notation.* Given an underlying EKS  $(\Omega, \mathcal{S}, \{K_i\}_{i \in \mathcal{A}})$  and  $f : \mathcal{P}(\Omega) \rightarrow \mathcal{P}(\Omega)$  we shall use  $\tilde{f}$  for the function  $f \upharpoonright_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{P}(\Omega)$ , i.e.,  $\tilde{f}(E) = f(E)$  for every  $E \in \mathcal{S}$ . Because of the closure properties of  $\mathcal{S}$ , for every  $i \in \mathcal{A}$  we have  $\tilde{K}_i : \mathcal{S} \rightarrow \mathcal{S}$ .

Notice that the AS and, in general KS, are EKS where  $\mathcal{S} = \mathcal{P}(\Omega)$ . Also Kripke frames [7] can be viewed as EKS with  $\mathcal{S} = \mathcal{P}(\Omega)$ . Other structures not discussed in this paper such as set algebras with operators (SOS) [24] and general frames [4] can be represented as EKSs where  $\mathcal{S}$  is required to be closed under complement.

#### 4.1 Extended KS and Distributive Lattices

The knowledge operators of an EKS are join-endomorphisms on a distributive lattice. This is an easy consequence of  $\mathfrak{K}1$  and  $\mathfrak{K}2$ , and the closure properties of EKS. The next proposition tells us that the *wiser* the agent, the lower that (its knowledge operator) is placed in the corresponding lattice.

**Proposition 15.** *Let  $(\Omega, \mathcal{S}, \{K_i\}_{i \in \mathcal{A}})$  be an EKS. Then  $L = (\mathcal{S}, \supseteq)$  is a distributive lattice and for each  $i \in \mathcal{A}$ ,  $\tilde{K}_i \in \mathcal{E}(L)$ .*

Conversely, the join-endomorphisms of distributive lattices correspond to knowledge operators of EKS. Recall that every distributive lattice is isomorphic to (the dual of) a lattice of sets. The next proposition is an adaptation to finite distributive lattices of Jónsson-Tarski duality for general-frames and boolean algebras with operators [17].

**Proposition 16.** *Let  $L$  be dual to a finite lattice of sets with a family  $\{f_i \in \mathcal{E}(L)\}_{i \in I}$ . Then  $(\Omega, \mathcal{S}, \{K_i\}_{i \in I})$  is an EKS where  $\mathcal{S} = L$ ,  $\Omega = \perp_L$ , and for every  $i \in I$ ,  $\mathcal{R}_i = \{(\omega, \omega') \in \Omega^2 \mid \text{for all } E \in \mathcal{S}, \omega \in f_i(E) \text{ implies } \omega' \in E\}$ . Also, for  $i \in I$ ,  $\tilde{K}_i = f_i$ .*

*Proof.* Notice that  $L = \mathcal{S}$  is closed under union and intersection since  $L$  is the dual of a lattice of sets. Showing  $\tilde{K}_i = f_i$  also proves that  $\mathcal{S}$  is closed under  $K_i$ . Recall that  $\tilde{K}_i(E) = K_i(E)$  for each  $E \in \mathcal{S}$ . Thus, it remains to prove  $K_i(E) = f_i(E)$  for all  $E \in \mathcal{S}$ . From  $\mathfrak{K}1$  and the fact that  $f_i$  is a join-endomorphism,  $K_i(E) = f_i(E) = \Omega$  for  $E = \Omega$ . Hence, choose an arbitrary  $E \neq \Omega$ . First suppose that  $\tau \in f_i(E)$ . From the definition of  $\mathcal{R}_i$  if  $(\tau, \tau') \in \mathcal{R}_i$ ,  $\tau' \in E$ . Hence  $\mathcal{R}_i(\tau) \subseteq E$ , so  $\tau \in K_i(E)$ .

Now suppose that  $\tau \in K_i(E)$  but  $\tau \notin f_i(E)$ . From  $\tau \in K_i(E)$  we obtain:

$$\text{for all } \tau' \in \Omega \text{ if } (\tau, \tau') \in \mathcal{R}_i \text{ then } \tau' \in E. \quad (4)$$

From the assumption  $\tau \notin f_i(E)$  and the monotonicity of join-endomorphisms (P.2):

$$\text{for every } F \in \mathcal{S} \text{ if } F \subseteq E \text{ then } \tau \notin f_i(F). \quad (5)$$

Let  $X = \{E' \in \mathcal{S} \mid \tau \in f_i(E')\}$ . If  $X = \emptyset$  then from the definition of  $\mathcal{R}_i$  we conclude  $\mathcal{R}_i(\tau) = \Omega$  which contradicts (4) since  $E \neq \Omega$ . If  $X \neq \emptyset$  take  $S = \bigcap X$ . Since  $f_i$  is a join-endomorphism, it distributes over intersection (i.e., the join in  $L$ ), we conclude  $\tau \in f_i(S)$ . Thus, if  $S \subseteq E$  we obtain a contradiction with (5). If  $S \not\subseteq E$  then there exists  $\tau' \in S$  such that  $\tau' \notin E$ . From the definition of  $S$ ,  $\tau' \in E'$  for each  $E' \in X$  such that  $\tau \in f_i(E')$ . But this implies  $(\tau, \tau') \in \mathcal{R}_i$  and  $\tau' \notin E$ , a contradiction with (4).  $\square$

Nevertheless, we can use our general characterization of join endomorphisms in the previous section (Cor.12) to obtain a simpler relational construction for join endomorphisms of powerset lattices (boolean algebras). Unlike the construction in Prop.16, this characterization of  $\mathcal{R}_i$  does not appeal to universal quantification.

**Proposition 17.** *Let  $L$  be dual to a finite powerset lattice with a family  $\{f_i \in \mathcal{E}(L)\}_{i \in I}$ . Let  $(\Omega, \{K_i\}_{i \in I})$  be the KS where  $\Omega = \perp_L$  and  $\mathcal{R}_i = \{(\omega, \omega') \mid \omega \in f_i(\overline{\{\omega'\}})\}$ . Then, for every  $i \in \mathcal{A}$ ,  $K_i = f_i$ .*

*Proof.* Since  $L$  is dual to a powerset lattice,  $\sqcup = \cap$ ,  $\sqsubseteq = \supseteq$ , and  $\mathcal{J}(L) = \{\overline{\{\tau\}} \mid \tau \in \Omega\}$ . Let  $Q = \{(\overline{\{\sigma\}}, \overline{\{\tau\}}) \mid (\sigma, \tau) \in \mathcal{R}_i\}$ . Notice that for every  $(\overline{\{\sigma\}}, \overline{\{\tau\}}) \in Q$ , we have  $\sigma \in f_i(\overline{\{\tau\}})$ . Equivalently,  $\{\sigma\} \subseteq f_i(\overline{\{\tau\}})$  and  $f_i(\overline{\{\tau\}}) \subseteq \{\sigma\}$ . Therefore, from Cor.12, it follows that for every  $E \in L$ ,  $f_i(E) = \bigcap \{\overline{\{\sigma\}} \in \mathcal{J}(L) \mid (\overline{\{\sigma\}}, \overline{\{\tau\}}) \in Q_i \text{ and } E \subseteq \overline{\{\tau\}}\}$ . We complete the proof as follows:

$$\begin{aligned}
 f_i(E) &= \bigcap \{\overline{\{\sigma\}} \in \mathcal{J}(L) \mid \exists \overline{\{\tau\}} \in \mathcal{J}(L) : ((\overline{\{\sigma\}}, \overline{\{\tau\}}) \in Q \text{ and } E \subseteq \overline{\{\tau\}})\} \\
 &= \bigcap \{\overline{\{\sigma\}} \in \mathcal{J}(L) \mid \neg \forall \overline{\{\tau\}} \in \mathcal{J}(L) : ((\overline{\{\sigma\}}, \overline{\{\tau\}}) \in Q \implies E \subseteq \overline{\{\tau\}})\} \\
 &= \bigcap \{\Omega \setminus \{\sigma\} \in \mathcal{J}(L) \mid \neg \forall \tau \in \Omega : ((\sigma, \tau) \in \mathcal{R}_i \implies \tau \in E)\} \\
 &= \bigcap \{\Omega \setminus \{\sigma\} \in \mathcal{J}(L) \mid \neg(\mathcal{R}_i(\sigma) \subseteq E)\} \\
 &= \Omega \setminus \{\sigma \in \Omega \mid \neg(\mathcal{R}_i(\sigma) \subseteq E)\} = \{\sigma \in \Omega \mid \mathcal{R}_i(\sigma) \subseteq E\} = K_i(E)
 \end{aligned}$$

We conclude this section by pointing out that accessibility relations can be obtained from knowledge operators. We can use Prop.17 to compute  $\mathcal{R}_i$  from  $K_i$ . For AS we can obtain the equivalence class  $\mathcal{R}_i(\omega)$  directly from  $K_i$ .

**Corollary 18.** *Let  $\mathcal{K} = (\Omega, \{K_i\}_{i \in \mathcal{A}})$  be a KS. Then (1)  $\mathcal{R}_i = \{(\omega, \omega') \mid \omega \in K_i(\overline{\{\omega'\}})\}$ . (2) If  $\mathcal{K}$  is an AS then  $\mathcal{R}_i(\omega) = K_i(\overline{\{\omega\}})$  for every  $\omega \in \Omega$ .*

## 5 Distributed Knowledge.

The notion of *distributed knowledge* represents the information that two or more agents may have as a group but not necessarily individually. Intuitively, it is what someone who knows what each agent, in a given group, knows. As described in [7], while common knowledge can be viewed as what ‘‘any fool’’ knows, distributed knowledge can be viewed as what a ‘‘wise man’’ would know.

Let  $(\Omega, \{K_i\}_{i \in \mathcal{A}})$  be a KS and  $i, j \in \mathcal{A}$ . The *distributed knowledge* of  $i$  and  $j$  is represented by  $D_{\{i,j\}} : \mathcal{P}(\Omega) \rightarrow \mathcal{P}(\Omega)$  defined as  $D_{\{i,j\}}(E) = \{\omega \in \Omega \mid \mathcal{R}_i(\omega) \cap \mathcal{R}_j(\omega) \subseteq E\}$  where  $\mathcal{R}_i$  and  $\mathcal{R}_j$  are the accessibility relations for  $i$  and  $j$ .

The following property captures the notion of distributed knowledge by relating group to individual knowledge: (K8)  $(K_i(E) \cap K_j(E \implies F)) \subseteq D_{\{i,j\}}(F)$ . It says that if one agents knows  $E$  and the other knows that  $E$  implies  $F$ , together they have the distributed knowledge of  $F$  even if neither agent knew  $F$ .

*Example 19.* Let  $E$  be the event “Bob’s boss is working from home” and  $F$  be the event “public transportation is suspended”. Suppose that agent Alice knows that Bob’s boss is working from home (i.e.,  $K_A(E)$ ), and that agent Bob knows that his boss works from home only when public transportation is suspended (i.e.,  $K_B(E \Rightarrow F)$ ). Thus, if they told each other what they knew, they would have distributed knowledge of  $F$  (i.e.,  $D_{\{A,B\}}(F)$ ). Indeed,  $K_A(E) \cap K_B(E \Rightarrow F)$  entails  $D_{\{A,B\}}(F)$  from  $\mathfrak{K}8$ .

A self-explanatory property relating individual and distributed knowledge is  $(\mathfrak{K}9)$   $K_i(E) \subseteq D_{\{i,j\}}(E)$ . Furthermore, the above basic properties of knowledge Prop. $\mathfrak{K}1$ - $\mathfrak{K}2$  also hold if we replace the  $K_i$  with  $D_{\{i,j\}}$ : Intuitively, distributed knowledge is knowledge. Indeed, imagine an agent  $m$  that combines  $i$  and  $j$ ’s knowledge by having an accessibility relation  $\mathcal{R}_m = \mathcal{R}_i \cap \mathcal{R}_j$ . In this case we would have  $K_m = D_{\{i,j\}}$ . Therefore, any KS may include distributed knowledge as one of its knowledge operators. For simplicity, we are considering distributed knowledge of two agents but this can be easily extended to arbitrary groups of agents. E.g. if  $K_m = D_{\{i,j\}}$  then  $D_{\{k,m\}}$  represents the distributed knowledge of three agents  $i$ ,  $j$  and  $k$ .

### 5.1 The Meet of Knowledge.

In Sec.4.1 we identified knowledge operators and join endomorphisms. We now show that the notion of distributed knowledge corresponds exactly to the meet of the knowledge operators in the lattice of all join-endomorphisms in  $(\mathcal{S}, \supseteq)$ .

**Theorem 20.** *Let  $(\Omega, \mathcal{S}, \{K_i\}_{i \in \mathcal{A}})$  be an EKS and let  $L$  be the lattice  $(\mathcal{S}, \supseteq)$ . Let us suppose that  $K_m = D_{\{i,j\}}$  for some  $i, j, m \in \mathcal{A}$ . Then  $\tilde{K}_m = \tilde{K}_i \sqcap_{\mathcal{E}(L)} \tilde{K}_j$ .*

*Proof.* Let us assume  $K_m = D_{\{i,j\}}$ . Then from the closure properties of  $\mathcal{S}$ , we have  $\tilde{D}_{\{i,j\}} = \tilde{K}_m : \mathcal{S} \rightarrow \mathcal{S}$ . Let  $f = \tilde{K}_i \sqcap_{\mathcal{E}(L)} \tilde{K}_j$ . (Recall that the order relation  $\sqsubseteq_L$  over  $L$  is reversed inclusion  $\supseteq$ , joins are intersections and meets are unions.)

From Prop. $\mathfrak{K}9$ , for every  $E \in \mathcal{S}$ ,  $D_{\{i,j\}}(E) \sqsubseteq_L K_i(E), K_j(E)$ . Thus  $\tilde{D}_{\{i,j\}}$  is a lower bound of both  $\tilde{K}_i$  and  $\tilde{K}_j$  in  $\mathcal{E}(L)$ , so  $\tilde{D}_{\{i,j\}} \sqsubseteq_{\mathcal{E}(L)} f$ .

To prove  $f \sqsubseteq_{\mathcal{E}(L)} \tilde{D}_{\{i,j\}}$ , take  $\tau \in \tilde{D}_{\{i,j\}}(E) = D_{\{i,j\}}(E)$  for an arbitrary  $E \in \mathcal{S}$ . By definition of  $D_{\{i,j\}}$ , we have (6)  $\mathcal{R}_i(\tau) \cap \mathcal{R}_j(\tau) \subseteq E$ . From Prop.5

$$f(E) = \bigcup \{K_i(F) \cap K_j(H) \mid F, H \in \mathcal{S} \text{ and } F \cap H \subseteq E\} \quad (7)$$

Take  $F = \mathcal{R}_i(\tau)$  and  $H = \mathcal{R}_j(\tau)$ , from (6),  $F \cap H \subseteq E$ . By definition of knowledge operator,  $\tau \in K_i(F)$  and  $\tau \in K_j(H)$ . From (7),  $\tau \in f(E)$ . Thus  $f \sqsubseteq_{\mathcal{E}(L)} \tilde{D}_{\{i,j\}}$ .  $\square$

The theorem above allows us to characterize an agent  $m$  having the distributed knowledge of  $i$  and  $j$  as the *least knowledgeable* agent wiser than both  $i$  and  $j$ . In the next section we consider the decision problem of whether a given  $m$  indeed has the distributed knowledge of  $i$  and  $j$ .

## 5.2 The Distributed Knowledge Problem.

In what follows, let  $(\Omega, \{K_i\}_{i \in \mathcal{A}})$  be a KS and let  $n = |\Omega|$ . Let us now consider the following decision: *Given the knowledge of agents  $i, j, m$ , decide whether  $m$  has the distributed knowledge of  $i$  and  $j$ , i.e.,  $K_m = D_{\{i,j\}}$ .*

The input for this problem is the knowledge of the agents and it can be represented using either knowledge operators  $K_i, K_j, K_m$  or accessibility relations  $\mathcal{R}_i, \mathcal{R}_j, \mathcal{R}_m$ . For each representation, the algorithm that solves the problem  $K_m = D_{\{i,j\}}$  can be implemented differently. For the first representation, it follows from Th.20 that  $K_m = D_{\{i,j\}}$  holds if and only if  $K_m = K_i \sqcap_{\mathcal{E}(L)} K_j$  where  $L = (\mathcal{P}(\Omega), \supseteq)$ . For the second one, we can verify  $\mathcal{R}_m = \mathcal{R}_i \cap \mathcal{R}_j$  instead. Indeed, as stated in Cor.18, one representation can be obtained from the other, hence an alternative solution for the decision problem is to translate the input from the given representation into the other one before solving.

Accessibility relations represent knowledge much more compactly than knowledge operators because the former are relations on  $\Omega^2$  while the latter are relations on  $\mathcal{P}(\Omega)^2$ . For this reason, it would seem in principle that the algorithm for handling the knowledge operator would be slower by several orders of magnitude. Nevertheless, we can use our lattice theoretical results from previous sections to show that this is not necessarily the case, thus it is worth considering both types of representations.

*From Knowledge Operators.* We wish to determine  $K_m = D_{\{i,j\}}$  by establishing whether  $K_m = K_i \sqcap_{\mathcal{E}(L)} K_j$  where  $L = (\mathcal{P}(\Omega), \supseteq)$ . Let us assume the following bitwise representation of knowledge operators. The states in  $\Omega$  are numbered as  $\omega_1, \dots, \omega_n$ . Each event  $E$  is represented as a number  $\#E \in [0..2^n - 1]$  whose binary representation has its  $k$ -th bit set to 1 iff  $\omega_k \in E$ . Each input knowledge operator  $K_i$  is represented as an array  $K_i$  of size  $2^n$  that stores  $\#K_i(E)$  at position  $\#E$ , i.e.,  $K_i[\#E] = \#K_i(E)$ .

From Lem.6,  $K_m = K_i \sqcap_{\mathcal{E}(L)} K_j$  iff  $K_m(E) = K_i(E) \cup K_j(E)$  for every join-irreducible element  $E$  in  $L$ . Notice that  $E \in \mathcal{J}(L)$  iff  $E$  has the form  $\overline{\{\omega_k\}}$  for some  $\omega_k \in \Omega$ . Moreover,  $\#\overline{\{\omega_k\}} = (2^n - 1) - 2^k$ . These facts lead us to the following result.

**Theorem 21.** *Given the arrays  $K_i, K_j, K_m$  where  $i, j, m \in I$ , there is an effective procedure that can decide  $K_m = D_{\{i,j\}}$  in time  $O(n^2)$  where  $n = |\Omega|$ .*

*Proof.* Let  $L = (\mathcal{P}(\Omega), \supseteq)$ . We have  $K_m = D_{\{i,j\}}$  iff  $K_m = K_i \sqcap_{\mathcal{E}(L)} K_j$  (Th.20) iff  $K_m(E) = K_i(E) \cup K_j(E)$  for every  $E \in \mathcal{J}(L)$  (Lem.6). Furthermore,  $E \in \mathcal{J}(L)$  iff  $E = \overline{\{\omega\}}$  for some  $\omega \in \Omega$ . Then we can conclude that  $E \in \mathcal{J}(L)$  iff  $\#E = (2^n - 1) - 2^k$  for some  $k \in [0..n-1]$ . Therefore,  $K_m = D_{\{i,j\}}$  iff for every  $k \in [0..n-1]$

$$K_m[p_k] = K_i[p_k] \mid K_j[p_k] \quad (8)$$

where  $p_k = (2^n - 1) - 2^k$  and  $\mid$  is the OR operation over the bitwise representation of  $K_i[p_k]$  and  $K_j[p_k]$ . For each  $k \in [0..n-1]$ , the equality test and the OR operation in Eq.8 can be computed in  $O(n)$ . Hence the total cost is  $O(n^2)$ .  $\square$

*From Accessibility Relations.* A very natural encoding for accessibility relations is to use a binary  $n \times n$  matrix. If the input is encoded using three matrices  $M_i, M_j$  and  $M_m$ , we can test whether  $\mathcal{R}_m = \mathcal{R}_i \cap \mathcal{R}_j$  (a proxy for  $K_m = D_{\{i,j\}}$ ) in  $O(n^2)$  by checking pointwise if  $M_m[a, b] = M_i[a, b] \cdot M_j[a, b]$ .

It suggests that for AS we can use a different encoding and check  $\mathcal{R}_m = \mathcal{R}_i \cap \mathcal{R}_j$  practically in *linear time*: More precisely in  $O(\alpha_n n)$  where  $\alpha_n$  is the inverse of the Ackermann function<sup>6</sup>. The key point is that the relations of AS are equivalences so they can be represented as *partitions*. The proof of the following result, which is interesting in its own right, shows an  $O(n\alpha_n)$  procedure for deciding  $\mathcal{R}_m = \mathcal{R}_i \cap \mathcal{R}_j$ .

**Theorem 22.** *Let  $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3 \subseteq \Omega^2$  be equivalences over a set  $\Omega$  of  $n = |\Omega|$  elements. There is an  $O(\alpha_n n)$  algorithm for the following problem:*

**Input:** *Each  $\mathcal{R}_i$  in partition form, i.e. an array of disjoint arrays of elements of  $\Omega$ , whose concatenation produces  $\Omega$ . This is readable in  $O(n)$ .*

**Output:** *Boolean answer to whether  $\mathcal{R}_3 = \mathcal{R}_1 \cap \mathcal{R}_2$ .*

*Proof.* We use the Disjoint-Sets data structure [8] whose details are included in the technical report <https://hal.archives-ouvertes.fr/hal-03323638>. We can view a disjoint-set as a function  $r : I \rightarrow I$  that satisfies  $r \circ r = r$  and can be evaluated at a particular index in  $O(\alpha_n)$ . The element  $r(i)$  corresponds to the class representative of  $i$  for each  $i \in I$ , so that  $i \sim_r j$  if and only if  $r(i) = r(j)$ .

If we let  $r_i$  denote a disjoint-set for  $\mathcal{R}_i$  for each  $i \in \{1, 2, 3\}$ , and we let  $q$  denote the disjoint-set for  $\mathcal{R}_1 \cap \mathcal{R}_2$ , then the problem can be divided into computing the disjoint-set  $q$  in  $O(n\alpha_n)$  and verifying whether  $\sim_q = \sim_{r_3}$  also in  $O(n\alpha_n)$ . To organize these claims, let us consider the following algorithm descriptions.

**Intersection.** Takes two disjoint-sets  $r_1$  and  $r_2$ , and produces a disjoint-set  $q$  such that  $i \sim_q j$  iff  $i \sim_{r_1} j$  and  $i \sim_{r_2} j$ .

**Canonical.** Takes a disjoint-set  $r$  and produces another  $\hat{r}$  with  $\sim_r = \sim_{\hat{r}}$ , but such that  $\hat{r}(i) \leq i$  for all  $i \in I$ .

**Equality.** Takes two disjoint-sets  $r_1, r_2$  and determines if  $i \sim_{r_1} j$  iff  $i \sim_{r_2} j$  for all  $i, j \in I$ . This problem is reduced simply to checking if  $\hat{r}_1 = \hat{r}_2$ .

We proceed to show that Algorithms 1 and 2 compute  $q$  and  $\hat{r}$  (in array form) in  $O(n\alpha_n)$ . The complexity follows from the fact that they must read the input function(s) pointwise and all other operations are linear. It remains to show correctness only.

The array  $g$  in Algo.1 is any version of the inverse image of  $\mathbf{f}$ , i.e.  $\mathbf{f}[g[y]] = y$  for every  $y \in \text{Im}(\mathbf{f})$ . This guarantees  $\mathbf{f} \circ g \circ \mathbf{f} = \mathbf{f}$  and hence  $q \circ q = g \circ \mathbf{f} \circ g \circ \mathbf{f} = g \circ \mathbf{f} = q$ . Moreover, for any  $i, j \in I$ ,  $q[i] = q[j]$  iff  $g[\mathbf{f}[i]] = g[\mathbf{f}[j]]$  by definition; iff  $\mathbf{f}[i] = \mathbf{f}[j]$  because  $\mathbf{f}$  is injective; iff  $r_1(i) = r_1(j)$  and  $r_2(i) = r_2(j)$ ; iff  $i \sim_{r_1} j$  and  $i \sim_{r_2} j$ .

Regarding Algo.2, for all  $i \in I$ ,  $i \sim t[r(i)]$ , thus  $r(i) = r(t[r(i)])$ . This is,  $r = r \circ t \circ r$ . Thus,  $\hat{r} \circ \hat{r} = t \circ r \circ t \circ r = t \circ r = \hat{r}$ . Moreover, for any  $i, j \in I$ ,  $i \sim j$  iff  $r(i) = r(j)$ ; iff  $t[r(i)] = t[r(j)]$  since  $t$  is injective on  $J$ ; iff  $\hat{r}[i] = \hat{r}[j]$  by definition.

**Experimental Results.** Fig.3 shows the average runtime (100 random executions) of the four algorithms listed below for the distributed knowledge problem. Fixing the number of elements  $n = |\Omega|$  elements, the input for each execution consisted of three randomly generated partitions  $P_i, P_j$  and  $P_m$ . The first two are generated independently

<sup>6</sup>  $\alpha_n \stackrel{\text{def}}{=} \min\{k : A(k, k) \geq n\}$ , where  $A$  is the Ackermann function. The growth of  $\alpha_n$  is negligible in practice, e.g.,  $\alpha_n = 4$  for  $n = 2^{2^{65536}} - 3$ .

---

**Algorithm 1** Intersection of disjoint sets in  $O(n\alpha_n)$

---

```

1: procedure INTERSECTION( $r_1, r_2$ )
2: Let  $\mathbf{f} : I \rightarrow I \times I$  be an array
3: For each  $i \in I$  do
4:    $\mathbf{f}[i] \leftarrow (r_1(i), r_2(i))$ 
5: Let  $\mathbf{g} : \text{Im}(\mathbf{f}) \rightarrow I$  be a hash map
6: For each  $i \in I$  do  $\mathbf{g}[\mathbf{f}[i]] \leftarrow i$ 
7: Let  $\mathbf{q} : I \rightarrow I$  be an array
8: For each  $i \in I$  do  $\mathbf{q}[i] \leftarrow \mathbf{g}[\mathbf{f}[i]]$ 
9: return  $\mathbf{q}$ 

```

---



---

**Algorithm 2** Equality of disjoint sets in  $O(n\alpha_n)$

---

```

1: procedure CANONICAL( $r$ )
2: (Comment)  $J \stackrel{\text{def}}{=} \{r(i) : i \in I\}$ .
3: Let  $\mathbf{t} : J \rightarrow I$  be a hash map.
4: For each  $i \in I$  do  $\mathbf{t}[r(i)] \leftarrow r(i)$ .
5: For each  $i \in I$  do
6:    $\mathbf{t}[r(i)] \leftarrow \min(\mathbf{t}[r(i)], i)$ 
7: Let  $\hat{\mathbf{r}} : I \rightarrow I$  be an array
8: For each  $i \in I$  do  $\hat{\mathbf{r}}[i] \leftarrow \mathbf{t}_{r(i)}$ 
9: return  $\hat{\mathbf{r}}$ 

```

---

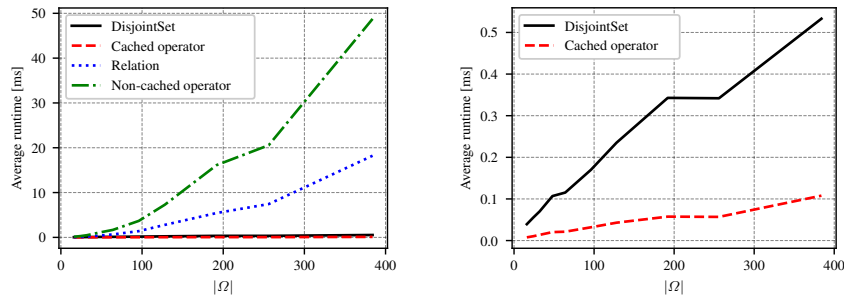


Fig. 3: Runtime comparison of several algorithms that solve the distributed knowledge problem.

and uniformly over the set of all possible partitions of  $n$  elements. The third,  $P_m$ , corresponds with 50% probability to the intersection of the relations of the first two, and to a different but very similar partition otherwise, so as to increase the problem difficulty.

1. The “Cached operator” algorithm is the one described in Th.21. It assumes that the input knowledge operators can be evaluated in  $O(1)$  at any join-irreducible input  $E \subseteq \Omega$ . Its complexity is  $O(n^2)$ , because bit-mask operations are linear w.r.t. the number of bits. However, this is compensated heavily in practice by the speed of bit-masking operations, at least for the sizes depicted.
2. The “Disjoint set” algorithm is the one described in Th.22 ( $O(n\alpha_n)$ ). It takes the accessibility relations in partition form as input.
3. The “Relation” algorithm ( $O(n^2)$ ) takes as input the accessibility relations in the form of  $n \times n$  binary matrices, and simply verifies if the pointwise-and matches.
4. The “Non-cached operator” ( $O(n^2)$ ) algorithm is that of the “Cached operator” when the cost of evaluating  $K_i(\cdot)$  is taken into account. It shows that although the “Cached operator” algorithm is very fast, its speed depends heavily on the assumption that the knowledge operators are pre-computed.



## 6 Concluding Remarks and Related Work.

We have used some standard tools from lattice theory to characterize the notion of distributed knowledge and provide efficient procedures to compute the meet of join-endomorphisms. Furthermore, we provide an algorithm to compute the intersection of *partitions* of a set of size  $n$  in  $O(n\alpha_n)$ . As illustrated in the introduction, this algorithm may have applications for graph connected components and other domains where the notion of partition and intersection arise naturally.

In [22] we proposed algorithms to compute  $f \sqcap_{\mathcal{E}(L)} g$  with time complexities  $O(n^3)$  for arbitrary lattices and  $O(n^2)$  for distributive lattices. Here we have improved the bound to  $O(n)$  for distributive lattices. The authors in [13] gave a method of logarithmic time complexity (in the size of the lattice) for meet operations. Since  $\mathcal{E}(L)$  is isomorphic to  $\mathcal{O}(\mathcal{J}(L) \times \mathcal{J}(L)^{op})$  for a distributive lattice  $L$ , finding  $f \sqcap_{\mathcal{E}(L)} g$  with their algorithm would be in  $O(\log_2(2^{n^2})) = O(n^2)$  in contrast to our linear bound. Furthermore, we would need a lattice isomorphic to  $\mathcal{E}(L)$  to find  $f \sqcap_{\mathcal{E}(L)} g$  using their algorithm. This lattice can be exponentially bigger than  $L$  [22] which is the input to our algorithm. We also provided experimental results illustrating the performance of our procedures. We followed the work in [16] for generating random distributive lattices.

The finite representation results we used in Sections 3 and 4 to obtain our main results are adaptations from standard results from duality theory. Jónsson and Tarski [19,17] originally presented an extension of boolean algebras with operators (BAO), called canonical extensions, provided with some representation theorems. Roughly speaking, the representation theorems state that (1) every relation algebra is isomorphic to a complete and atomic relation algebra and (2) every boolean algebra with operators is isomorphic to a complex algebra that is complete and atomic. The idea behind this result, as was presented later by Kripke in [20], basically says that the operators can be recovered from certain binary relations and vice versa. Another approach to this duality was given by Goldblatt [11] where it is stated that the variety of normal modal algebras coincides with the class of subalgebras defined on the class of all frames. Canonical extensions have been useful for the development of duality and algebra. Jónsson proved an important result for modal logic in [18] and the authors of [10,9,6] have generalized canonical extensions for BAOs to distributive and arbitrary bounded lattices and posets.

Distributed knowledge was introduced in [15] and various axiomatization and expressiveness for it have been provided, e.g., in [14,1]. In terms of computational complexity, the satisfiability problem for epistemic logic with distributed knowledge ( $S5^D$ ) has been shown to be PSPACE-complete [7]. Nevertheless, we are not aware of any lattice theoretical characterization of distributed knowledge nor algorithms to decide if an agent has the distributed knowledge of others.

**Acknowledgments.** We are indebted to the anonymous referees and editors of RAM-iCS 2021 for helping us to improve the overall quality of the paper. We thank Luigi Santocanale for his constructive comments.

## References

1. Ågotnes, T., Wáng, Y.N.: Resolving distributed knowledge. *Artif. Intell.* **252**, 1–21 (2017)

2. Aumann, R.J.: Agreeing to disagree. *The Annals of Statistics* **4**, 1236–1239 (1976)
3. Bloch, I., Heijmans, H., Ronse, C.: Mathematical morphology. In: *Handbook of Spatial Logics*. pp. 857–944. Springer Netherlands (2007)
4. Chagrov, A., Zakharyashev, M.: *Modal Logic*, vol. 35. Oxford University Press (1997)
5. Davey, B.A., Priestley, H.A.: *Introduction to Lattices and Order*. Cambridge University Press, 2 edn. (2002)
6. Dunn, J.M., Gehrke, M., Palmigiano, A.: Canonical extensions and relational completeness of some substructural logics. *Journal of Symbolic Logic* **70**(3), 713–740 (2005)
7. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: *Reasoning about knowledge*. MIT press Cambridge, 4th edn. (1995)
8. Galler, B.A., Fisher, M.J.: An improved equivalence algorithm. *Communications of the ACM* **7**(5), 301–303 (1964)
9. Gehrke, M., Harding, J.: Bounded lattice expansions. *Journal of Algebra* pp. 345–371 (2001)
10. Gehrke, M., Jónsson, B.: Bounded distributive lattice expansions. *Mathematica Scandinavica* **94**(1), 13–45 (2004), <http://www.jstor.org/stable/24493402>
11. Goldblatt, R.: Varieties of complex algebras. *Annals of Pure and Applied Logic* **44**(3), 173–242 (1989). [https://doi.org/10.1016/0168-0072\(89\)90032-8](https://doi.org/10.1016/0168-0072(89)90032-8)
12. Grätzer, G., Schmidt, E.: On the lattice of all join-endomorphisms of a lattice. *Proceedings of The American Mathematical Society* **9**, 722–722 (1958)
13. Habib, M., Nourine, L.: Tree structure for distributive lattices and its applications. *Theoretical Computer Science* **165**(2), 391–405 (1996)
14. Hakli, R., Negri, S.: Proof theory for distributed knowledge. In: *CLIMA. Lecture Notes in Computer Science*, vol. 5056, pp. 100–116. Springer (2007)
15. Halpern, J.Y., Moses, Y.: Knowledge and common knowledge in a distributed environment. *J. ACM* **37**(3), 549–587 (1990)
16. Jipsen, P., Lawless, N.: Generating all finite modular lattices of a given size. *Algebra universalis* **74**(3), 253–264 (2015)
17. Jónsson, B., Tarski, A.: Boolean algebras with operators. Part II. *American Journal of Mathematics* **74**(1), 127–162 (1952), <http://www.jstor.org/stable/2372074>
18. Jónsson, B.: On the canonicity of sahlqvist identities. *Studia Logica* **53**(4), 473–491 (1994)
19. Jónsson, B., Tarski, A.: Boolean algebras with operators. Part I. *American Journal of Mathematics* **73**(4), 891–939 (1951), <http://www.jstor.org/stable/2372123>
20. Kripke, S.A.: A completeness theorem in modal logic. *The Journal of Symbolic Logic* **24**(1), 1–14 (1959)
21. McKinsey, J.C.C., Tarski, A.: On closed elements in closure algebras. *Annals of Mathematics* **47**(1), 122–162 (1946), <http://www.jstor.org/stable/1969038>
22. Quintero, S., Ramírez, S., Rueda, C., Valencia, F.: Counting and computing join-endomorphisms in lattices. In: *RAMiCS. Lecture Notes in Computer Science*, vol. 12062, pp. 253–269. Springer (2020)
23. Samet, D.: Agreeing to disagree: The non-probabilistic case. *Games and Economic Behavior* **69**(1), 169–174 (2010). <https://doi.org/10.1016/j.geb.2008.09.032>
24. Samet, D.: S5 knowledge without partitions. *Synthese* **172**(1), 145–155 (2010)