Algebraic Structures from Concurrent Constraint Programming Calculi for Distributed Information in Multi-Agent Systems

Michell Guzmán\textsuperscript{d,1}, Sophia Knight\textsuperscript{c}, Santiago Quintero\textsuperscript{d}, Sergio Ramírez\textsuperscript{a}, Camilo Rueda\textsuperscript{a}, Frank Valencia\textsuperscript{a,b,*}

\textsuperscript{a}Department of Electronics and Computer Science
Pontificia Universidad Javeriana
Cali, Colombia
\textsuperscript{b}CNRS, LIX École Polytechnique de Paris, France
\textsuperscript{c}University of Minnesota Duluth, Minnesota, USA
\textsuperscript{d}LIX École Polytechnique de Paris, France

Abstract

Spatial constraint systems (scs) are semantic structures for reasoning about spatial and epistemic information in concurrent systems. We develop the theory of scs to reason about the distributed information of potentially infinite groups. We characterize the notion of distributed information of a group of agents as the infimum of the set of join-preserving functions that represent the spaces of the agents in the group. We provide an alternative characterization of this notion as the greatest family of join-preserving functions that satisfy certain basic properties. For completely distributive lattices, we establish that distributed information of a group is the greatest information below all possible combinations of information in the spaces of the agents in the group that derive a given piece of information. We show compositionality results for these characterizations and conditions under which information that can be obtained by an infinite group can also be obtained by a finite group. Finally, we provide an application on mathematical morphology where dilations, one of its fundamental operations, define an scs on a powerset lattice. We show that distributed information represents a particular dilation in such scs.

Keywords: Reasoning about Groups, Distributed Knowledge, Infinitely Many Agents, Reasoning about Space, Mathematical Morphology, Algebraic Modeling.

Contents

1 Introduction \hfill \textsuperscript{2}

2 Background \hfill \textsuperscript{5}

\textsuperscript{*}This work has been partially supported by the ECOS-NORD project FACTS (C19M03)
\textsuperscript{1}Present address: fortiss - Research Institute of the Free State of Bavaria
Email addresses: guzman@fortiss.org (Michell Guzmán), sknight@umn.edu (Sophia Knight), sqquinter@polytechnique.fr (Santiago Quintero), sergio@javerianacali.edu.co (Sergio Ramírez), crueda@javerianacali.edu.co (Camilo Rueda), frank.valencia@lix.polytechnique.fr (Frank Valencia)
1. Introduction

In current distributed systems such as social networks, actors behave more as members of a certain group than as isolated individuals. Information, opinions, and beliefs of a particular actor are frequently the result of an evolving process of interchanges with other actors in a group. This suggests a reified notion of group as a single actor operating within the context of the collective information of its members. It also conveys two notions of information, one spatial and the other epistemic. In the former, information is localized in compartments associated with a user or group. In the latter, it refers to something known or believed by a single agent or collectively by a group.

In this paper we pursue the development of a principled account of a reified notion of group by taking inspiration from the epistemic notion of distributed knowledge [1]. A group has its information distributed among its member agents. We thus develop a theory about what exactly is the information available to agents as a group when considering all that is distributed among its members.

In our account a group acts itself as an agent carrying the collective information of its members. We can interrogate, for instance, whether there is a potential contradiction or unwanted distributed
information that a group might be involved in among its members or by integrating a certain agent. This is a fundamental question since it may predict or prevent potentially dangerous evolutions of the system.

Furthermore, in many real life multi-agent systems, the agents are unknown in advance. New agents can subscribe to the system in unpredictable ways. Thus, there is usually no a-priori bound on the number of agents in the system. It is then often convenient to model the group of agents as an infinite set. In fact, in models from economics and epistemic logic \cite{2,3}, groups of agents have been represented as infinite, even uncountable, sets. In accordance with this fact, in this paper we consider that groups of agents can also be infinite. This raises interesting issues about the distributed information of such groups. In particular, that of group compactness: information that when obtained by an infinite group can also be obtained by one of its finite subgroups.

Context. Constraint systems (cs)\footnote{For simplicity we use cs for both constraint system and its plural form.} are algebraic structures for the semantics of process calculi from concurrent constraint programming (ccp) \cite{4}. In this paper we shall study cs as semantic structures for distributed information of a group of agents.

A cs can be formalized as a complete lattice \((\mathbb{C}, \sqsubseteq)\). The elements of \(\mathbb{C}\) represent partial information and we shall think of them as being assertions. They are traditionally referred to as constraints since they naturally express partial information (e.g., \(x > 42\)). The order \(\sqsubseteq\) corresponds to entailment between constraints, \(c \sqsubseteq d\), often written \(d \sqsupseteq c\), means \(c\) can be derived from \(d\), or that \(d\) represents as much information as \(c\). The join \(\sqcup\), the bottom \(\sqcap\), and the top \(\top\) of the lattice correspond to conjunction, the empty information, and the join of all (possibly inconsistent) information, respectively.

The notion of computational space and the epistemic notion of belief in the spatial ccp (scp) process calculi \cite{1} is represented as a family of join-preserving maps \(s_i : \mathbb{C} \rightarrow \mathbb{C}\) called space functions. A cs equipped with space functions is called a spatial constraint system (scs). From a computational point of view \(s_i(c)\) can be interpreted as an assertion specifying that \(c\) resides within the space of agent \(i\). From an epistemic point of view, \(s_i(c)\) specifies that \(i\) considers \(c\) to be true. An alternative epistemic view is that \(i\) interprets \(c\) as \(s_i(c)\). All these interpretations convey the idea of \(c\) being local or subjective to agent \(i\).

This work. In the spatial ccp process calculus scp \cite{5}, scs are used to specify the spatial distribution of information in configurations \((P,c)\) where \(P\) is a process and \(c\) is a constraint, called the store, representing the current partial information. E.g., a reduction \(\langle P, s_1(a) \sqcup s_2(b) \rangle \rightarrow \langle Q, s_1(a) \sqcup s_2(b \sqcup c) \rangle\) means that \(P\), with \(a\) in the space of agent 1 and \(b\) in the space of agent 2, can evolve to \(Q\) while adding \(c\) to the space of agent 2.

Given the above reduction, assume that \(e\) is some piece of information resulting from the combination (join) of the three constraints above, i.e., \(e = a \sqcup b \sqcup c\), but strictly above the join of any two of them. We are then in the situation where neither agent has \(e\) in their spaces, but as a group they could potentially have \(e\) by combining their information. Intuitively, \(e\) is distributed in the spaces of the group \(I = \{1, 2\}\). Being able to predict the information that agents 1 and 2 may derive as group is a relevant issue in multi-agent concurrent systems, particularly if \(e\) represents unwanted or conflicting information (e.g., \(e = \text{false}\)).

In this work we develop the theory of group space functions \(D_I : \mathbb{C} \rightarrow \mathbb{C}\) to reason about information distributed among the members of a potentially infinite group \(I\). We shall refer to \(D_I\) as the distributed space of group \(I\). In our theory \(d \sqsupseteq D_I(c)\) holds exactly when we can derive
from $d$ that $e$ is distributed among the agents in $I$. For example, for $e = a \cup b \cup c$ given above, we will have $d = s_1(a) \cup s_2(b \cup c) \supseteq \mathbb{D}_{\{1,2\}}(e)$ meaning that from the information $s_1(a) \cup s_2(b \cup c)$ we can derive that $e$ is distributed among the group $I = \{1,2\}$. Furthermore, $\mathbb{D}_J(e) \supseteq \mathbb{D}_J(e)$ holds whenever $I \subseteq J$ since if $e$ is distributed among a group $I$, it should also be distributed in a group that includes the agents of $I$.

Distributed information of infinite groups can be used to reason about multi-agent computations with unboundedly many agents. For example, a computation in sccp is a possibly infinite reduction sequence $\gamma$ of the form $\langle P_0, c_0 \rangle \rightarrow \langle P_1, c_1 \rangle \rightarrow \cdots$ with $c_0 \subseteq c_1 \subseteq \cdots$. The result of $\gamma$ is $\bigsqcup_{n \geq 0} c_n$, the join of all the stores in the computation. In sccp all fair computations from a configuration have the same result $[3]$. Thus, the observable behaviour of $P$ with initial store $c$, written $O(P,c)$, is defined as the result of any fair computation starting from $\langle P, c \rangle$. Now consider a setting where in addition to their sccp capabilities in $[5]$, processes can also create new agents. Hence, unboundedly many agents, say agents $1,2,\ldots$, may be created during an infinite computation. In this case, $O(P,c) \supseteq \mathbb{D}_N(false)$, where $N$ is the set of natural numbers, would imply that some (finite or infinite) set of agents in any fair computation from $\langle P, c \rangle$ may reach contradictory local information among them. Notice that from the above-mentioned properties of distributed spaces, the existence of a finite set of agents $H \subseteq N$ such that $O(P,c) \supseteq \mathbb{D}_H(false)$ implies $O(P,c) \supseteq \mathbb{D}_N(false)$. The converse of this implication will be called group compactness and we will provide meaningful sufficient conditions for it to hold.

Contributions and Organization.

The paper starts with some background on lattice theory in Section 2 and on spatial constraint systems in Section 3. The main contributions are given in Sections 4 and 5 and are listed below:

1. We characterize the distributed space $\mathbb{D}_I$ as the greatest space function below the space functions that represent the spaces (or beliefs) of the agents of a possibly infinite group $I$ (Section 4.2).
2. We provide an alternative characterization of a distributed space as the greatest function that satisfies certain basic properties (Section 4.3).
3. We show that distributed spaces have an inherent compositional nature: The information of a group is determined by that of its subgroups (Section 4.9).
4. We provide a group compactness result: Given an infinite group $I$, we identify a meaningful condition under which $c \supseteq \mathbb{D}_I(e)$ implies $c \supseteq \mathbb{D}_J(e)$ for some finite group $J \subseteq I$ (Section 4.7).
5. We then show that without this meaningful condition we cannot guarantee that $c \supseteq \mathbb{D}_J(e)$ implies $c \supseteq \mathbb{D}_J(e)$ for some finite group $J \subseteq I$ (Section 4.8).
6. We provide a characterization of distributed spaces for distributive lattices: Given an infinite group $I$, $\mathbb{D}_I(e)$ can be viewed as the greatest information below all possible combinations of information in the spaces of the agents in $I$ that derive $e$ (Section 4.9).
7. Finally, we investigate applications of the theory developed in this paper to geometry and mathematical morphology (MM) (Section 5). Below we use $A, B, \ldots$ to denote sets in a vector space. In geometry, the Minkowski addition is given by $A \oplus B = \{a + b \mid a \in A, b \in B\}$ [6]. It is well-known that the distribution law $A \oplus (B \cap C) = (A \cap B) \oplus (A \cap C)$ holds for convex sets.\footnote{A convex set is a set of points such that, given any two points in that set, the line segment joining them lies entirely within that set.}

4
but not in general. As a simple application of our theory, we identify a novel and pleasant law for $A \oplus (B \cap C)$: Namely, $A \oplus (B \cap C) = \bigcap_{X \subseteq A} (X \oplus B) \cup ((A \setminus X) \oplus C)$.

In MM, a dilation by a structuring element $A$ can be seen as a function $\delta_A$ that transforms every input image $X$ into the image $\delta_A(X) = A \oplus X$. We show that dilations are space functions and that the distributed space corresponding to these dilations is the dilation that arises from the intersection of their structuring elements: i.e., if $s_1 = \delta_A$ and $s_2 = \delta_B$ then $D\{1, 2\} = \delta_{A \cap B}$.

All in all, in this paper we put forward an algebraic theory for group reasoning in the context of ccp that can also be applied to other domains. The theory here developed can be used in the semantics of the spatial ccp process calculus to reason about or prevent potential unwanted evolutions of ccp processes. One could imagine the incorporation of group reasoning in a variety of process algebraic settings and indeed we expect that such formalisms will appear in due course. We will also show that our algebraic theory can be applied to prove new results in other realms such as geometry and mathematical morphology.

Remark 1. This paper is the extended version of the CONCUR’19 paper in [8] with full proofs and the contributions described above in the points 5, 6 and 7. For the sake of the reviewers we also include contents, index and subject tables and for Sections 4 and 5, which contain the main contributions, we include a summary at the end of each section.
(v) $P$ is said to be a completely distributive lattice if it is a complete lattice and for any doubly indexed subset $\{x_{ij}\}_{i \in I, j \in J_i}$ of $P$

$$\bigsqcup_{i \in I} \left( \bigcap_{j \in J_i} x_{ij} \right) = \bigcap_{f \in F} \left( \bigsqcup_{i \in I} x_{i f(i)} \right)$$

where $F$ is the class of choice functions $f$ choosing for each index $i \in I$ some index $f(i) \in J_i$.

Our space functions will be defined as self-maps with some structural properties intended to capture our notion of space.

**Definition 2 ([11])**. Let $(L, \sqsubseteq)$ be a complete lattice. A self-map on $L$ is a function $f$ from $L$ to $L$. Let $f$ be a self-map on $L$.

(i) $f$ is monotonic if for every $a, b \in L$ such that $a \sqsubseteq b$, then $f(a) \sqsubseteq f(b)$.
(ii) We say that $f$ preserves the join of a set $S \subseteq L$ if and only if $f(\bigsqcup S) = \bigsqcup \{f(c) \mid c \in S\}$.
(iii) We say that $f$ preserves arbitrary joins if and only if it preserves the join of any arbitrary set.
(iv) $f$ is continuous if and only if it preserves the join of any directed set on $L$.

We conclude this background section with a well-known fact about continuous functions.

**Proposition 1 ([13])**. Let $(P, \sqsubseteq)$ be a poset where $P$ is a countable set. Let $f$ be a self-map that preserves the join of increasing chains, i.e., for every $S = \{c_1, c_2, \ldots\} \subseteq P$ such that $c_1 \sqsubseteq c_2 \sqsubseteq \cdots$, we have $f(\bigsqcup S) = \bigsqcup \{f(c) \mid c \in S\}$. Then $f$ is continuous.

3. Spatial and Standard Constraint Systems

In this section we recall the notion of constraint system and its spatial extension [5]. Furthermore, we generalize this extension to allow for infinitely many agents and state some results that will be used in later sections.

3.1. Constraint Systems

Constraint systems [4] are semantic structures to specify partial information. They can be formalized as complete lattices [14].

**Definition 3 (Constraint Systems [14])**. A constraint system (cs) is a complete lattice $(C, \sqsubseteq)$. The elements of $C$ are called constraints. The symbols $\sqcup$, true and false will be used to denote the least upper bound (lub) operation, the bottom, and the top element of $C$.

The elements of a cs $(C, \sqsubseteq)$, the constraints, represent (partial) information. A constraint $c$ can be viewed as an assertion. The lattice order $\sqsubseteq$ is meant to capture entailment of information: $c \sqsubseteq d$, alternatively written $d \sqsupseteq c$, means that the assertion $d$ represents at least as much information as $c$. We think of $d \sqsupseteq c$ as saying that $d$ entails $c$ or that $c$ can be derived from $d$. The operator $\sqcap$ represents join of information; $c \sqcap d$ can be seen as an assertion stating that both $c$ and $d$ hold. We can think of $\sqcup$ as representing conjunction of assertions. The top element represents the join of all, possibly inconsistent, information, hence it is referred to as false. The bottom
reason about possibly infinite groups of agents. We say that $c$ is consistent if $c \neq \text{false}$, otherwise we say that $c$ is inconsistent. Similarly, we say that $c$ is consistent/inconsistent with $d$ if $c \sqcup d$ is consistent/inconsistent.

Distributivity is ubiquitous in order theory and it plays a fundamental role in the results of this paper. We consider three forms of distribution.

**Definition 4 (Distributive Constraint Systems).** A cs $(C, \sqsubseteq)$ is said to be distributive (completely distributive) iff it is a distributive (completely distributive) lattice. It is said to be a constraint frame iff its joins distribute over arbitrary meets. More precisely, $c \sqcup \bigwedge S = \bigwedge \{c \sqcup e \mid e \in S\}$ for every $c \in C$ and $S \subseteq C$.

Clearly every completely distributive cs is a constraint frame and every constraint frame is also distributive cs. For finite constraint systems all the three notions of distributivity are equivalent.

Constraint frames allow us to define a general form of implication by adapting the corresponding notion from Heyting Algebras to constraint systems. A Heyting implication $c \rightarrow d$ in our setting corresponds to the weakest constraint one needs to join $c$ with to derive $d$.

**Definition 5 (Heyting Implication).** Let $(C, \sqsubseteq)$ be a cs. Define $c \rightarrow d$ as $\bigwedge \{e \in C \mid c \sqcup e \sqsupseteq d\}$.

The following properties of Heyting implication correspond to standard logical properties with $\rightarrow$, $\sqcup$, and $\sqsubseteq$ interpreted as implication, conjunction, and entailment.

**Proposition 2 ([15]).** Let $(C, \sqsubseteq)$ be a constraint frame. For every $c, d \in C$ the following holds: (1) $c \sqcup (c \rightarrow d) = c \sqcup d$, (2) $(c \rightarrow d) \sqsubseteq d$, (3) $c \rightarrow d = \text{true}$ iff $c \sqsubseteq d$.

We conclude this section with a simple cs example.

**Example 1 (Powerset Constraint System).** The power set of any set $S$ ordered by inclusion $(\mathcal{P}(S), \subseteq)$ is a cs. In fact it is the stereotypical example of completely distributive cs. In this case true = $\emptyset$, false = $S$, for every $A, B \subseteq S$, $A \sqcup B = A \cup B$, $A \sqcap B = A \cap B$, $A \rightarrow B = B \setminus A$.

### 3.2. Spatial Constraint Systems

The authors of [5] extended the notion of cs to account for distributed and multi-agent scenarios with a finite number of agents, each having their own space for local information and their computations. The extended structures are called spatial constraint systems (scs). Here we adapt scs to reason about possibly infinite groups of agents.

A group $G$ is a set of agents. Each $i \in G$ has a space function $s_i : C \rightarrow C$ satisfying some structural conditions. Recall that constraints can be viewed as assertions. Thus given $c \in C$, we can then think of the constraint $s_i(c)$ as an assertion stating that $c$ is a piece of information residing within the space of agent $i$. Some alternative epistemic interpretations of $s_i(c)$ is that it is an assertion stating that agent $i$ believes $c$, that $c$ holds within the space of agent $i$, or that agent $i$ interprets $c$ as the constraint $s_i(c)$. All these interpretations convey the idea that $c$ is local or subjective to agent $i$.

In [5] scs are used to specify the spatial distribution of information in configurations $\langle P, c \rangle$ where $P$ is a process and $c$ is a constraint. E.g., a reduction $\langle P, s_i(c) \sqcup s_j(d) \rangle \rightarrow \langle Q, s_i(c) \sqcup s_j(d \sqcup e) \rangle$ means that $P$ with $c$ in the space of agent $i$ and $d$ in the space of agent $j$ can evolve to $Q$ while adding $e$ to the space of agent $j$.

We now introduce the notion of space function.
Definition 6 (Space Functions). A space function over a constraint system \((C, \sqsubseteq)\) is a continuous self-map \(f : C \to C\) such that for every \(c, d \in C\):

(S.1) \(f(\text{true}) = \text{true}\), and
(S.2) \(f(c \sqcup d) = f(c) \sqcup f(d)\).

We shall use \(S(C)\) to denote the set of all space functions over \(C\).

The assertion \(f(c)\) can be viewed as saying that \(c\) is in the space represented by \(f\). Property (S.1) states that having an empty local space amounts to nothing. Property (S.2) allows us to join and distribute the information in the space represented by \(f\).

Remark 2 (Continuity). In [5] space functions were not required to be continuous. Nevertheless, we will argue later, in Remark 5, that continuity comes naturally in the intended phenomena we wish to capture: modeling information of possibly infinite groups. In fact, in [5] scs could only have finitely many agents. In this work we also extend scs to allow arbitrary, possibly infinite, sets of agents.

The continuity and preservation of finite joins by space functions will provide us with their preservation of arbitrary joins. The following proposition gives us sufficient conditions for the existence of the join of an arbitrary set on a given poset. A sketch of its proof is presented in [16], for the sake of completeness we present our own proof of it.

Proposition 3 ([16]). Let \((P, \sqsubseteq)\) be a poset. Suppose that \(\bigsqcup F\) and \(\bigsqcup D\) exist for every finite set \(F \subseteq P\) and for every directed set \(D \subseteq P\). Then \(\bigsqcup A\) also exists for every \(A \subseteq P\).

Proof. Let \(A \subseteq P\) be an arbitrary set and let \(D = \{\bigsqcup F \mid F \subseteq A\text{ and } F\text{ is finite}\}\). First we prove that \(D\) is directed. For \(F_1, F_2 \subseteq A\), both finite sets, the elements \(\bigsqcup F_1, \bigsqcup F_2 \in D\) and the set \(F_3 = F_1 \sqcup F_2 \subseteq A\) is finite. Then \(\bigsqcup F_3 \in D\) and, both \(\bigsqcup F_1 \sqsubseteq \bigsqcup F_3\) and \(\bigsqcup F_2 \sqsubseteq \bigsqcup F_3\) hold.

To complete the proof we show that for every \(c \in P\):

\[c\text{ is an upper bound of } D\text{ if and only if } c\text{ is an upper bound of } A.\]

For the “only if” direction, we prove its contrapositive. Let \(c \in P\). If \(c\) is not an upper bound of \(A\), there is an \(a \in A\) such that \(a \not\sqsubseteq c\). Notice that \(\{a\} \subseteq A\) and thus \(\bigsqcup \{a\} = a \in D\) but \(a \not\sqsubseteq c\). Hence \(c\) is not an upper bound of \(D\).

For the other direction, assume that \(c \in P\) is an upper bound of \(A\). Let \(F\) be any finite subset of \(A\). Since \(e \sqsubseteq c\) for every \(e \in F\), then \(\bigsqcup F \sqsubseteq c\). Therefore, \(c\) is an upper bound of \(D\).

We then conclude \(\bigsqcup A = \bigsqcup D\) as wanted. 

The following proposition states two useful properties of space functions: monotonicity and preservation of arbitrary joins.

Proposition 4. Let \(f : C \to C\) be a function over a cs \((C, \sqsubseteq)\). Then

1. If \(f\) is space function then \(f\) is monotonic.
2. \(f\) is space function if and only if it preserves arbitrary joins.

Proof. Let \(f : C \to C\) be a function over a cs \((C, \sqsubseteq)\).

1. Suppose \(f : C \to C\) is a space function and let \(c, d \in C\) such that \(c \sqsubseteq d\). From axiom (S.2) in Def. 6, \(f(c \sqcup d) = f(c) \sqcup f(d)\). Since \(c \sqcup d = d\), we know that \(f(c \sqcup d) = f(d)\). Therefore \(f(d) = f(c) \sqcup f(d)\), that implies \(f(c) \sqsubseteq f(d)\).
2. The “if” direction is immediate. For the “only if” direction, assume that $f$ is a space function. Let $A \subseteq C$ be an arbitrary set and define $D = \{ \bigsqcup F \mid F \subseteq A \text{ and } F \text{ is finite} \}$. We can prove $f(\bigsqcup A) = \bigsqcup \{ f(a) \mid a \in A \}$ by using the following derivation:

$$
f \left( \bigsqcup A \right)
= \langle \bigsqcup A = \bigsqcup D \text{ from Prop. 4} \rangle
f \left( \bigsqcup D \right)
= \langle f \text{ is continuous and } D \text{ is directed.} \rangle
\bigsqcup \{ f \left( \bigsqcup F \right) \mid F \subseteq A \text{ and } F \text{ is finite} \}
= \langle f \text{ preserves finite joins.} \rangle
\bigsqcup \{ \bigsqcup \{ f(e) \mid e \in F \} \mid F \subseteq A \text{ and } F \text{ is finite} \}
= \langle \text{Simplifying} \rangle
\bigsqcup \{ f(e) \mid e \in A \}
$$

A spatial constraint system is a constraint system with a possibly infinite group of agents each one having a space function. We specify such a group as a tuple of space functions.

**Definition 7 (Spatial Constraint Systems).** A spatial constraint system (scs) is a constraint system $(C, \sqsubseteq)$ equipped with a possibly infinite tuple $s = (s_i)_{i \in G}$ of space functions from $\mathcal{S}(C)$.

We shall use $(C, \sqsubseteq, (s_i)_{i \in G})$ to denote an scs with a tuple $(s_i)_{i \in G}$. We refer to $G$ and $s$ as the group of agents and space tuple of $C$ and to each $s_i$ as the space function in $C$ of agent $i$. Subsets of $G$ are also referred to as groups of agents (or sub-groups of $G$).

Let us illustrate a simple scs that will be used throughout the paper.

**Example 2.** The scs $(C, \sqsubseteq, (s_i)_{i \in \{1, 2\}})$ in Fig. 1 is given by the complete lattice $\mathbf{M}_2$ and two agents. We have $C = \{ p \lor \neg p, p, \neg p, p \land \neg p \}$ and $c \sqsubseteq d$ holds if $c$ is a logical consequence of $d$. The top element (false) is $p \lor \neg p$, the bottom element (true) is $p \lor \neg p$, and the constraints $p$ and $\neg p$ are incomparable with each other.

The set of agents is $\{1, 2\}$ with space functions $s_1$ and $s_2$: For agent 1, $s_1(p) = \neg p$, $s_1(\neg p) = p$, $s_1(\text{false}) = \text{false}$, $s_1(\text{true}) = \text{true}$, and for agent 2, $s_2(p) = \text{false} = s_2(\text{false})$, $s_2(\neg p) = \neg p$, $s_2(\text{true}) = \text{true}$. The intuition is that the agent 2 sees no difference between $p$ and false while agent 1 interprets $\neg p$ as $p$ and vice versa.

More involved examples of scs include meaningful families of structures from logic and economics such as Kripke structures and Aumann structures (see [5]). We illustrate scs with infinite groups in the next section.
4. Distributed Information

This section contains the main technical contributions of this paper. In particular, we will characterize the notion of collective information of a group of agents. Roughly speaking, the distributed (or collective) information of a group $I$ is the join of each piece of information that resides in the space of an agent $i \in I$. The distributed information of $I$ w.r.t. $c$ is the distributive information of $I$ that can be derived from $c$. We wish to formalize whether a given $e$ can be derived from the collective information of the group $I$ w.r.t. $c$.

The following examples, which we will use throughout this paper, illustrate the above intuition.

Example 3. Consider an scs $(C, \sqsubseteq, (s_i)_{i \in G})$ where $G = \mathbb{N}$ and $(C, \sqsubseteq)$ is a constraint frame. Let $c = s_1(a) \sqcup s_2(a \rightarrow b) \sqcup s_3(b \rightarrow e)$. The constraint $c$ specifies the situation where $a$, $a \rightarrow b$ and $b \rightarrow e$ are in the spaces of agent 1, 2 and 3, respectively. Neither agent necessarily holds $e$ in their space w.r.t. $c$. Nevertheless, the information $e$ can be derived from the collective information of the three agents w.r.t. $c$, since from Prop. 2 we have $a \sqcup (a \rightarrow b) \sqcup (b \rightarrow e) \sqsupseteq e$.

Let us now consider an example with infinitely many agents. Let $c' = \bigsqcup_{i \in \mathbb{N}} s_i(a_i)$ for some infinite increasing chain $a_0 \sqsubseteq a_1 \sqsubseteq \cdots$. Take $c'$ such that $\bigsqcup_{i \in \mathbb{N}} a_i \sqsupseteq c'$. Notice that unless $c'$ is compact (see Def. 1), it may be the case that no agent $i \in \mathbb{N}$ holds $c'$ in their space; e.g., if $c' \sqsupseteq a_i$ for any $i \in \mathbb{N}$. Yet, from our assumption, $c'$ can be derived from the collective information w.r.t. $c'$ of all the agents in $\mathbb{N}$, i.e., $\bigsqcup_{i \in \mathbb{N}} a_i$.

The above example may suggest that distributed information can be obtained by joining individual local information derived from $c$. Such information can be characterized as the $i$-projection of agent $i$ w.r.t. $c$.

Definition 8 (Agent and Join Projections). Let $(C, \sqsubseteq, (s_i)_{i \in G})$ be an scs. Given $i \in G$, the $i$-agent projection of $c \in C$ is defined as $\pi_i(c) \equiv \bigsqcup \{ e \mid c \sqsupseteq s_i(e) \}$. We say that $e$ is $i$-agent derivable from $c$ if and only if $\pi_i(c) \supseteq e$. Given $I \subseteq G$ the $I$-join projection of a group $I$ of $c$ is defined as $\pi_I(c) \equiv \bigsqcup \{ \pi_i(c) \mid i \in I \}$. Similarly, we say that $e$ is $I$-join derivable from $c$ if and only if $\pi_I(c) \supseteq e$.

The $i$-agent projection of $i \in G$ of $c$ naturally represents the join of all the information that agent $i$ has in $c$. The $I$-join projection of group $I$ joins individual $i$-agent projections of $c$ for $i \in I$. This projection can be used as a sound mechanism for reasoning about distributed-information: If $e$ is $I$-join derivable from $c$ then it follows from the distributed-information of $I$ w.r.t. $c$. 

![Figure 1: Cs given by lattice $M_2$ ordered by logical implication and space functions $s_1$ and $s_2$.](image-url)
Example 4. Let \(c\) be as in Ex. 3. We have \(\pi_1(c) \sqsupseteq a, \pi_2(c) \sqsupseteq (a \rightarrow b)\) and \(\pi_3(c) \sqsupseteq (b \rightarrow e)\). Indeed, \(e\) is \(I\)-join derivable from \(c\) since \(\pi_{(1,2,3)}(c) = \pi_1(c) \sqcup \pi_2(c) \sqcup \pi_3(c) \sqsupseteq e\). Similarly, we conclude that \(e'\) is \(I\)-join derivable from \(c'\) in Ex. 3 since \(\pi_{N}(c') = \bigsqcup_{i \in \mathbb{N}} \pi_i(c) \supseteq \bigsqcup_{i \in \mathbb{N}} a_i \supseteq e'\).

Nevertheless, \(I\)-join projections do not provide a complete mechanism for reasoning about distributed information as illustrated below.

Example 5. Let \(d \triangleq s_1(b) \sqcap s_2(b)\). Recall that we think of \(\sqcup\) and \(\sqcap\) as conjunction and disjunction of assertions: \(d\) specifies that \(b\) is present in the space of agent 1 or in the space of agent 2 though not exactly in which one. Thus from \(d\) we should be able to conclude that \(b\) belongs to the space of some agent in \(\{1, 2\}\). Nevertheless, \(b\) is not necessarily \(I\)-join derivable from \(d\) since from \(\pi_{(1,2)}(d) = \pi_1(d) \sqcup \pi_2(d)\) we cannot, in general, derive \(b\). To see this consider the scs in Fig. 2(a) taking \(b = \neg p\). We have \(\pi_{(1,2)}(d) = \pi_1(d) \sqcup \pi_2(d) = true \sqcup true = true \not\sqsupseteq b\). One can generalize the example to infinitely many agents. Consider the scs in Ex. 3 and let \(d' \triangleq \bigsqcap_{i \in \mathbb{N}} s_i(b')\). We should be able to conclude from \(d'\) that \(b'\) is in the space of some agent in \(\mathbb{N}\) but, in general, \(b'\) is not \(\mathbb{N}\)-join derivable from \(d'\).

4.1. Distributed Spaces

We have just illustrated in Ex. 3 that the \(I\)-join projection of \(c, \pi_I(c)\), the join of individual projections, may not project all distributed information of a group \(I\). To solve this problem we develop the notion of \(I\)-group projection of \(c\), written as \(\Pi_I(c)\). We will first define a space function \(D_I\) called the distributed space of group \(I\). The function \(D_I\) can be thought of as a virtual space including all the information that can be in the space of a member of \(I\). We will then define an \(I\)-projection, \(\Pi_I\), in terms of \(D_I\) much like \(\pi_i\) is defined in terms of \(s_i\).
Set of Space Functions

We now introduce a new partial order induced by \( \mathbb{C} \): The set of space functions ordered point-wise. Recall that \( \mathcal{S}(\mathbb{C}) \) denotes the set of all space functions over a cs \( \mathbb{C} \) (Def. 6). For notational convenience, we shall use \((f_I)_{I \subseteq \mathbb{G}}\) to denote the tuple \((f_I)_{I \in \mathbb{P}(\mathbb{G})}\) of elements of \( \mathcal{S}(\mathbb{C}) \).

**Definition 9 (Function Order).** Let \((\mathbb{C}, \sqsubseteq)\) be a cs. Given \( f, g : \mathbb{C} \rightarrow \mathbb{C} \) define \( f \sqsubseteq_s g \) iff \( f(c) \sqsubseteq g(c) \) for every \( c \in \mathbb{C} \).

An important design aspect of our structure is that the set of space functions \( \mathcal{S}(\mathbb{C}) \) can be made into a complete lattice.

**Lemma 1 ([17]).** Let \((\mathbb{C}, \sqsubseteq)\) be a cs. Then \((\mathcal{S}(\mathbb{C}), \sqsubseteq_s)\) is a complete lattice.

**Proof.** Let \( \mathcal{S} \subseteq \mathcal{S}(\mathbb{C}) \) be subset of space functions. For every constraint \( c \in \mathbb{C} \), we let \( \bigcup_{c \in \mathcal{S}} S \) denote the tuple \( (\bigcup_{c \in \mathcal{S}} f(c) \mid f \in \mathcal{S}) \). It is easy to verify from its definition that if \( \mathcal{S} \subseteq \mathcal{S}(\mathbb{C}) \) then it is the least upper bound of \( \mathcal{S} \). Let us then show that \( \bigcup_{c \in \mathcal{S}} S \in \mathcal{S}(\mathbb{C}) \), i.e., it is a space function.

- \( \bigcup_{c \in \mathcal{S}} S \) (true) = \( \bigcup_{c \in \mathcal{S}} \{ true \} = \bigcup_{c \in \mathcal{S}} \{ true \} = true. \)

- \( \bigcup_{c \in \mathcal{S}} S \) (c \sqcup d) = \( \bigcup_{c \in \mathcal{S}} S \) (c) \sqcup \( \bigcup_{c \in \mathcal{S}} S \) (d).

\[
\begin{align*}
\left( \bigcup_{c \in \mathcal{S}} S \right) (c \sqcup d) &= \bigcup_{c \in \mathcal{S}} \{ f(c \sqcup d) \mid f \in \mathcal{S} \} \\
&= \bigcup_{c \in \mathcal{S}} \{ f(c) \sqcup f(d) \mid f \in \mathcal{S} \} \\
&= \left( \bigcup_{c \in \mathcal{S}} \{ f(c) \mid f \in \mathcal{S} \} \right) \sqcup \left( \bigcup_{c \in \mathcal{S}} \{ f(d) \mid f \in \mathcal{S} \} \right) \\
&= \left( \bigcup_{c \in \mathcal{S}} S \right) (c) \sqcup \left( \bigcup_{c \in \mathcal{S}} S \right) (d)
\end{align*}
\]

- Continuity: \( \bigcup_{c \in \mathcal{S}} S \) \( (\sqcup_{D} D) \) = \( \bigcup_{c \in \mathcal{S}} \left\{ \left( \bigcup_{c \in \mathcal{S}} S \right) (d) \mid d \in D \right\} \) for any directed set \( D \).

Let \( D \) be a directed set. From definition of \( \bigcup_{c \in \mathcal{S}} S \) and by the continuity of each space function \( f \in \mathcal{S} \), we have

\[
\begin{align*}
\left( \bigcup_{c \in \mathcal{S}} S \right) \left( \sqcup_{D} D \right) &= \bigcup_{c \in \mathcal{S}} \left\{ f \left( \bigcup_{c \in \mathcal{S}} D \right) \mid f \in \mathcal{S} \right\} \\
&= \bigcup_{c \in \mathcal{S}} \left\{ \bigcup_{c \in \mathcal{S}} \{ f(d) \mid d \in D \} \mid f \in \mathcal{S} \right\} \\
&= \bigcup_{c \in \mathcal{S}} \left\{ \bigcup_{c \in \mathcal{S}} \{ f(d) \mid f \in \mathcal{S} \} \mid d \in D \right\} \\
&= \bigcup_{c \in \mathcal{S}} \left\{ \left( \bigcup_{c \in \mathcal{S}} S \right) (d) \mid d \in D \right\}
\end{align*}
\]

as required. \( \square \)

In the next section we use the properties of \((\mathcal{S}(\mathbb{C}), \sqsubseteq_s)\) to formalize distributed spaces of a group \( I \) as the greatest space function below every space function \( s_i \) with \( i \in I \).
4.2. Distributed Spaces as Max Spaces

We can now give the definition of distributed spaces. It is convenient to give the following intuition first.

**Remark 3.** Suppose that \( f \) and \( g \) are space functions in \( S(C) \) with \( f \sqsubseteq g \), i.e., \( f(c) \subseteq g(c) \) for every \( c \in C \). Intuitively, every piece of information \( c \) in the space represented by \( g \) is also in the space represented by \( f \). This can be interpreted as saying that the space represented by \( g \) is included in the space represented by \( f \); in other words the bigger the space, the smaller the function that represents it. \( \square \)

Following the above intuition, the order relation \( \sqsubseteq \) of \( S(C) \) represents (reverse) space inclusion and the join and meet operations in \( S(C) \) represent intersection and union of spaces. The biggest and the smallest spaces are represented by the bottom and the top elements of the lattice \( S(C) \), here called \( \lambda_\bot \) and \( \lambda_\top \), respectively, and defined as follows.

**Definition 10 (Top and Bottom Spaces).** Let \( S(C) \) be the lattice of space functions. Define \( \lambda_\bot \) and \( \lambda_\top \) in \( S(C) \) as follows: \( \lambda_\bot(c) \equiv true \) for every \( c \in C \); and \( \lambda_\top(c) \equiv true \) if \( c = true \) and \( \lambda_\top(c) \equiv false \) if \( c \neq true \).

The distributed space \( D_I \) of a group \( I \) can be viewed as the function that represents the smallest space that includes all the local information of the agents in \( I \). From Remark 3, \( D_I \) should be the *greatest space function* below the space functions of the agents in \( I \). The existence of such a function follows from completeness of \(( S(C), \sqsubseteq) \) stated in Lemma 1.

**Definition 11 (Distributed Spaces).** Let \( (C, \sqsubseteq, (s_i)_{i \in G}) \) be an scs. The distributed spaces of \( C \) is given by \( D = (D_I)_{I \subseteq G} \) where

\[
D_I \equiv def \ max \ \{ f \in S(C) \mid f \sqsubseteq s_i \text{ for every } i \in I \}.
\]

We shall say that \( e \) is distributed among \( I \subseteq G \) w.r.t. \( c \) if and only if \( c \sqsupseteq D_I(e) \). We shall refer to each \( D_I \) as the (distributed) space of the group \( I \).

**Remark 4.** From Lemma 1, \( D_I = \bigsqcup_{S(C)} \{ f \in S(C) \mid f \sqsubseteq s_i \text{ for each } i \in I \} = \bigsqcap_{S(C)} \{ s_i \mid i \in I \} \) where \( \bigsqcup_{S(C)} \) and \( \bigsqcap_{S(C)} \) are the join and meet in the complete lattice \(( S(C), \sqsubseteq) \).

Let us consider a concrete example.

**Example 6.** Fig. 2b illustrates an scs with space functions \( s_1 \) and \( s_2 \), and their distributed space \( D_{\{1,2\}} \). The reader can verify that \( D_{\{1,2\}} \) is indeed the greatest function such that \( D_{\{1,2\}} \sqsubseteq s_1 \) and \( D_{\{1,2\}} \sqsubseteq s_2 \). Notice that \( s_1(p \sqcup s_2(\neg p)) \sqsubseteq D_{\{1,2\}}(p \sqcup \neg p) = D_{\{1,2\}}(false) \) meaning that if agents 1 and 2 had \( p \) and \( \neg p \) in their corresponding spaces, as a group they could derive an inconsistency.

4.3. Compositionality of Distributed Spaces

Distributed spaces have pleasant compositional properties. They capture the intuition that the distributed information of a group \( I \) can be obtained from the the distributive information of its subgroups.
Theorem 1. Let $\langle D_I \rangle_{I \subseteq G}$ be the distributed spaces of an scs $(C, \subseteq, (s_i)_{i \in G})$. Suppose that $K, J \subseteq I \subseteq G$.

1. $D_I = \lambda_T$ if $I = \emptyset$.
2. $D_I = s_i$ if $I = \{i\}$.
3. $D_J(a) \sqcup D_K(b) \supseteq D_I(a \sqcup b)$.
4. $D_J(a) \sqcup D_K(a \rightarrow e) \supseteq D_I(c)$ if $(C, \subseteq)$ is a constraint frame.

Proof. 1. It follows directly from Def. [10] and Def. [11]
2. Let $I = \{i\}$, from Def. [11] $D_I = \max\{f \in S(C) \mid f \subseteq s_i\} = s_i$.
3. Assume $K, J \subseteq I$. From Def. [11] we conclude $D_I \subseteq_s D_J$ and $D_I \subseteq_s D_K$. Thus $D_J(a) \supseteq D_I(a), D_K(b) \supseteq D_I(b)$ and therefore $D_J(a) \sqcup D_K(b) \supseteq D_I(a) \sqcup D_I(b)$. Since $D_I$ is a space function, $D_I(a) \sqcup D_I(b) = D_J(a \sqcup b)$, then we obtain $D_J(a) \sqcup D_K(b) \supseteq D_J(a \sqcup b)$ as wanted.
4. It follows from part (3) with $a = a$ and $b = a \rightarrow c$, and Prop. [2].

Recall that $\lambda_T$ corresponds to the empty space (see Def. [10]). The first property realizes the intuition that the empty subgroup $\emptyset$ does not have any information whatsoever distributed w.r.t. a consistent $c$: if $c \supseteq D_I(c)$ and $c \neq false$ then $e = true$. Intuitively, the second property says that the function $D_I$ for the group of one agent must be the agent’s space function. The third property states that a group can join the information of its subgroups. The last property uses constraint implication, hence the constraint frame condition, to express that by joining the information $a$ and $a \rightarrow c$ of their subgroups, the group $I$ can obtain $c$.

Let us illustrate how to derive information of a group from smaller ones using Th. [1].

Example 7. Let $c = s_1(a) \sqcup s_2(a \rightarrow b) \sqcup s_3(b \rightarrow e)$ as in Ex. [3]. We want to prove that $c$ is distributed among $I = \{1, 2, 3\}$ w.r.t. $c$, i.e., $c \supseteq D_{\{1,2,3\}}(c)$. Using Properties (2) and (4) in Th. [1] we obtain $c \supseteq s_1(a) \sqcup s_2(a \rightarrow b) = D_{\{1\}}(a) \sqcup D_{\{2\}}(a \rightarrow b) \supseteq D_{\{1,2\}}(b)$, and then $c \supseteq D_{\{1,2\}}(b) \sqcup s_3(b \rightarrow e) = D_{\{1,2\}}(b) \sqcup D_{\{3\}}(b \rightarrow e) \supseteq D_{\{1,2,3\}}(e)$ as wanted.

Remark 5 (Continuity Revisited). The example with infinitely many agents in Ex. [3] illustrates well why we require our spaces to be continuous in the presence of possibly infinite groups. Clearly $c' = \bigsqcup_{i \in \mathbb{N}} s_i(a_i) \supseteq \bigsqcup_{i \in \mathbb{N}} D_N(a_i)$. By continuity, $\bigsqcup_{i \in \mathbb{N}} D_N(a_i) = D_N(\bigsqcup_{i \in \mathbb{N}} a_i)$ which indeed captures the idea that each $a_i$ is in the distributed space $D_N$.

4.4. Distributed Spaces in Aumann Structures

We now consider an important structure from mathematical economics used for group epistemic reasoning: Aumann structures [3]. We illustrate that the notion of distributed knowledge in these structures is an instance of a distributed space.

Example 8. Aumann Constraint Systems. Aumann structures are an event-based approach to modelling knowledge. An Aumann structure is a tuple $A = (S, P_1, \ldots, P_n)$ where $S$ is a set of states and each $P_i$ is a partition on $S$ for agent $i$. The partitions are called information sets.

If two states $t$ and $u$ are in the same information set for agent $i$, it means that in state $t$ agent $i$ considers state $u$ possible, and vice versa. An event in an Aumann structure is any subset of $S$. Event $e$ holds at state $t$ if $t \in e$. The set $P_i(s)$ denotes the information set of $P_i$ containing $s$. The
event of agent \(i\) knowing \(e\) is defined as \(K_i(e) = \{s \in S \mid P_i(s) \subseteq e\}\), and the distributed knowledge of an event \(e\) among the agents in a group \(I\) is defined as \(D_I(e) = \{s \in S \mid \bigcap_{i \in I} P_i(s) \subseteq e\}\).

An Aumann structure can be seen as a spatial constraint system \(C(A)\) with events as constraints, i.e., \(C = \{e \mid e\ \text{is an event in } A\}\), and for every \(e_1, e_2 \in C, e_1 \subseteq e_2\) iff \(e_2 \subseteq e_1\). The operators join (\(\sqcup\)) and meet (\(\sqcap\)) are intersection (\(\cap\)) and union (\(\cup\)) of events, respectively; true = \(S\) and false = \(\emptyset\).

The space functions are the knowledge operators, i.e., \(s_i(c) = K_i(c)\). From these definitions and since meets are unions one can verify that \(D_I(c) = D_I(e)\) which shows that distributed knowledge is an instance of distributed information.

In Th.\(\mathbb{I}\) we listed some useful properties about \((D_I)_{I \subseteq G}\). In the next section we shall see that \((D_I)_{I \subseteq G}\) is the greatest solution of three basic properties.

### 4.5. Distributed Spaces as Group Distributions Candidates.

We now wish to single out a few fundamental properties on tuples of self-maps that can be used to characterize distributed spaces.

**Definition 12 (Distribution Candidates).** Let \((C, \sqsubseteq, (s_i)_{i \in G})\) an scs. A group distribution candidate (gdc) of \(C\) is a tuple \(d = (d_I)_{I \subseteq G}\) of self-maps on \(C\) such that for each \(I, J \subseteq G\):

1. (D.1) \(d_I\) is a space function in \(C\),
2. (D.2) \(d_I = s_i\) if \(I = \{i\}\),
3. (D.3) \(d_I \sqsupseteq s_j\) if \(I \subseteq J\).

Property (D.1) requires each \(d_I\) to be a space function. This is trivially met for \(d_I = D_I\). Property (D.2) says that the function \(d_I\) for a group of one agent must be the agent’s space function. Clearly, \(d_{\{i\}} = D_{\{i\}}\) satisfies (D.2); indeed the distributed space of a single agent is their own space. Finally, Property (D.3) states that \(d_I(c) \sqsubseteq d_J(c)\), if \(I \subseteq J\). This is also trivially satisfied if we take \(d_I = D_I\) and \(d_J = D_J\). Indeed if a group \(I\) has some distributed information \(c\) then any group \(J\), that includes \(I\), should also have \(c\). This realizes the intuition in Remark\(\mathbb{B}\). The bigger the group, the bigger the space and thus the smaller the space function that represents it.

Properties (D.1)-(D.3), however, do not determine \(D\) uniquely. In fact, there could be infinitely-many tuples of space functions that satisfy them. For example, if we were to chose \(d_\emptyset = \lambda_\emptyset\), \(d_{\{i\}} = s_i\) for every \(i \in G\), and \(d_I = \lambda_I\) whenever \(|I| > 1\) then (D.1)-(D.3) would be trivially met. But these space functions would not capture our intended meaning of distributed spaces: E.g., we would have \(true \sqsupseteq d_I(e)\) for every \(e\) thus implying that any \(e\) would be distributed in the empty information \(true\) amongst the agents in \(I \neq \emptyset\).

Nevertheless, we prove that \((D_I)_{I \subseteq G}\) is the greatest solution satisfying (D.1)-(D.3).

**Theorem 2 (Max gdc).** Let \((D_I)_{I \subseteq G}\) be the distributed spaces of \((C, \sqsubseteq, (s_i)_{i \in G})\). Then

1. \((D_I)_{I \subseteq G}\) is a gdc of \(C\).
2. If \((d_I)_{I \subseteq G}\) is a gdc of \(C\) then \(d_I \sqsubseteq_s D_I\) for each \(I \subseteq G\).

**Proof.** Let \((D_I)_{I \subseteq G}\) be the distributed spaces of \(C\).

1. We need to prove that \((D_I)_{I \subseteq G}\) satisfies properties (D.1)-(D.3) in Def.\(\mathbb{I}\).
   Property (D.1) follows from definition of \(D_I\) (see Def.\(\mathbb{I}\)). Property (D.2) is proven in Th.\(\mathbb{I}\) part (2). For property (D.3), let \(I, J \subseteq G\) such that \(I \subseteq J\). Notice that \(\{f \in S(C) \mid f \sqsubseteq_s s_i\\} \subseteq \{f \in S(C) \mid f \sqsubseteq_s s_j\ \text{for every } j \in J\}\). Then \(D_J \sqsubseteq_s D_I\).
Example 9. Let \( g \) be the greatest space functions, \( I \) bigger the group, the bigger the projection. The last property says that whatever is \( c \in d_1 \) with \( \{1, 2, 3\} \). We want to prove \( c \sqsupseteq D_I(e) \) for \( I = \{1, 2, 3\} \). From (D.2) we have \( c \sqsubseteq D_{\{1\}}(a) \sqcup D_{\{2\}}(a \to b) \sqcup D_{\{3\}}(b \to e) \). We can then use (D.3) to obtain \( c \sqsupseteq D_I(a) \sqcup D_J(a \to b) \sqcup D_{\varnothing}(b \to e) \). Finally, by D.1 and Prop. 2 we infer \( c \sqsupseteq D_I(a \cup (a \to b) \cup (b \to e)) \sqsupseteq D_I(e) \), thus \( c \sqsupseteq D_I(e) \) as wanted. Now consider our counterexample in Ex. 3 with \( d = s_1(b) \sqcap s_2(b) \). We wish to prove \( d \sqsubseteq D_I(b) \) for \( I = \{1, 2\} \), i.e., \( b \) can be derived from \( d \) as being in a space of a member of \( \{1, 2\} \). Using (D.2) and (D.3) we obtain \( d = s_1(b) \sqcap s_2(b) = D_{\{1\}}(b) \sqcap D_{\{2\}}(b) \sqsubseteq D_{\{1, 2\}}(b) \sqsubseteq D_{\varnothing}(b) \) as wanted.

We shall use the characterization of distributed spaces in Th. 2 in the proofs of Prop. 5 and Th. 6.

4.6. Group Projections

As promised at the beginning of Section 4.2 we now give a definition of Group Projection. The function \( \Pi_I(c) \) extracts exactly all information that the group \( I \) may have distributed w.r.t. \( c \).

Definition 13 (Group Projection). Let \( (D_I)_{I \subseteq G} \) be the distributed spaces of an ssc \((C; \sqsubseteq, (s_i)_{i \in G})\). Given the set \( I \subseteq G \), the \( I \)-group projection of \( c \in C \) is defined as \( \Pi_I(c) = \bigcup\{e \mid c \sqsupseteq D_I(e)\} \). We say that \( c \) is \( I \)-group derivable from \( e \) if and only if \( \Pi_I(c) \supseteq e \).

Much like space functions and agent projections, group projections and distributed spaces also form a pleasant correspondence: a Galois connection 9.

Proposition 5. Let \( (D_I)_{I \subseteq G} \) be the distributed spaces of an ssc \((C; \sqsubseteq, (s_i)_{i \in G})\). For every \( c, e \in C \),

1. \( c \sqsupseteq D_I(e) \) if and only if \( \Pi_I(c) \supseteq e \).
2. \( \Pi_I(c) \supseteq \Pi_J(c) \) if \( J \subseteq I \).
3. \( \Pi_I(c) \supseteq \pi_I(c) \).

Proof. Let \( (D_I)_{I \subseteq G} \) be the distributed spaces of \( C \) and let \( c, e \in C \).

1. Let \( S = \{d \mid c \sqsupseteq D_I(d)\} \). First, assume that \( c \sqsupseteq D_I(e) \). Since \( e \in S \), by Def. 13 \( \Pi_I(c) = \bigcup S \supseteq e \). Second, assume \( \Pi_I(c) \supseteq e \). Then by monotonicity, \( D_I(\Pi_I(c)) \supseteq D_I(e) \). From continuity of \( D_I \), we know that \( D_I(\Pi_I(c)) = D_I(\bigcup S) = \bigcup \{D_I(d) \mid d \in S\} \) and by definition of \( S \), for every \( d \in S \), we have \( c \sqsupseteq D_I(d) \), then \( c \supseteq \bigcup \{D_I(d) \mid d \in S\} \). Therefore, \( c \supseteq D_I(e) \).
2. Given that \( (D_I)_{I \subseteq G} \) is a gdc (see Th. 2), if \( J \subseteq I \), then \( D_J \supseteq_a D_I \). Hence \( \{d \mid c \sqsupseteq D_J(d)\} \subseteq \{d \mid c \sqsupseteq D_I(d)\} \) and thus \( \Pi_I(c) \supseteq \Pi_J(c) \) for every \( c \in C \).
3. By part (2), for every \( \{i\} \subseteq I \) and every \( c \in C \), we have \( \Pi_I(c) \supseteq \Pi_{\{i\}}(c) \). It implies, \( \Pi_I(c) \supseteq \bigcup_{i \in I} \Pi_{\{i\}}(c) \), for every \( c \in C \). Then \( \bigcup_{i \in I} \Pi_{\{i\}}(c) = \bigcup_{i \in I} \bigcup \{d \mid c \sqsupseteq D_{\{i\}}(d)\} \) = \( \bigcup_{i \in I} \{c \in \pi_I(c)\} = \pi_I(c) \). Therefore, \( \Pi_I(c) \supseteq \pi_I(c) \), for every \( c \in C \).

The first property in Prop. 5, a Galois connection, states that we can conclude from \( c \) that \( e \) is in the distributed space of \( I \) exactly when \( e \) is \( I \)-group derivable from \( c \). The second says that the bigger the group, the bigger the projection. The last property says that whatever is \( I \)-join derivable is \( I \)-group derivable, although the opposite is not true as shown in Ex. 5.
4.7. Group Compactness.

Suppose that an infinite group of agents $I$ can derive $e$ from $c$ (i.e., $c \supseteq \mathbb{D}_I(e)$). A legitimate question is whether there exists a finite sub-group $J$ of agents from $I$ that can also derive $e$ from $c$. The following theorem provides a positive answer to this question given that $e$ is a compact element (see Section 2) and $I$-join derivable from $c$.

**Theorem 3** (Group Compactness). Let $(\mathbb{D}_I)_{I \subseteq G}$ be the distributed spaces of an scs $(C, \sqsubseteq, (s_i)_{i \in G})$. Suppose that $c \supseteq \mathbb{D}_I(e)$. If $e$ is compact and $I$-join derivable from $c$ then there exists a finite set $J \subseteq I$ such that $c \supseteq \mathbb{D}_J(e)$.

**Proof.** Suppose that $c \supseteq \mathbb{D}_I(e)$. If $I$ is finite then take $J = I$. If $I$ is not finite, since $e$ is $I$-join derivable from $c$ we have $\pi_I(c) = \bigcup S \supseteq e$ where $S = \{\pi_i(c) \mid i \in I\}$.

Define $D_I = \{\pi_J(c) \mid J \subseteq I \land J \text{ finite}\}$. Take any $\pi_H(c), \pi_K(c) \in D_I$. Since $H$ and $K$ are finite, their union $K \cup H$ must also be finite and included in $I$. Hence $\pi_{H \cup K}(c) \in D_I$. Therefore, $D_I$ is a directed set.

Since $S = \{\pi_i(c) \mid i \in I\} = \{\pi_J(c) \mid i \in I\}$ is included in $D_I$, we obtain $\bigcup D_I \supseteq \bigcup S \supseteq e$.

But $e$ is compact and $D_I$ directed hence there must be $\pi_J(c) \in D_I$, with $J$ a finite set, such that $\pi_J(c) \supseteq e$. From Prop. 5 (3) and Prop. 3 (1), we conclude $c \supseteq \mathbb{D}_J(e)$ as wanted. \qed

Let us illustrate Th. 3 with our recurrent example.

**Example 10.** Consider the example with infinitely many agents in Ex. 3. We have $c' = \bigcup_{i \in \mathbb{N}} s_i(a_i)$ for some increasing chain $a_0 \subseteq a_1 \subseteq \cdots$, and $e'$ such that $e' \sqsubseteq \bigcup_{i \in \mathbb{N}} a_i$. Notice that $e' \supseteq \mathbb{D}_N(e')$ and $\pi_N(e') \supseteq e'$. Hence $e'$ is $N$-join derivable from $c'$. If $e'$ is compact, by Th. 3 there must be a finite subset $J \subseteq \mathbb{N}$ such that $c' \supseteq \mathbb{D}_J(e')$.

4.8. Group-Compactness without $I$-join derivability

Let us assume $c \supseteq \mathbb{D}_I(e)$ as in Th. 3. By Prop. 3 (1), we know that $e$ is $I$-group derivable from $c$ but not necessarily $I$-join derivable from $c$. The problem in establishing group compactness in the absence of $I$-join derivability has to do with $d'$ in the infinite case. In Ex. 5 we have $d' = \bigcap_{i \in \mathbb{N}} s_i(b')$. Notice that we cannot guarantee that $b'$ is $N$-join derivable from $d'$ ($\pi_N(d') \supseteq b'$). One can verify that $d' \supseteq \mathbb{D}_N(b')$, i.e., $b'$ resides in the space of agent $i$ for some $i \in \mathbb{N}$. Then, $b'$ is $I$-group derivable from $d'$ ($\pi_N(d') \supseteq b'$). Nevertheless we cannot guarantee the existence of a finite $J \subseteq \mathbb{N}$ such that $d' \supseteq \mathbb{D}_J(b')$. In fact, the existence of such a $J$ cannot be guaranteed even if $e$ ($b'$ in Ex. 5) is compact as stated in the next theorem.

**Theorem 4** (Non-Compactness). There exists an scs $(C, \sqsubseteq, (s_i)_{i \in G})$ with distributed spaces $(\mathbb{D}_I)_{I \subseteq G}$ such that for some $c, e \in C$ and $I \subseteq G$: (1) $e$ is compact, (2) $c \supseteq \mathbb{D}_I(e)$ but (3) there is no finite subset $J \subseteq I$ with $c \supseteq \mathbb{D}_J(e)$.

**Proof.** Consider the scs $(C, \leq, (s_n)_{n \in \mathbb{N}})$ (Fig. 3) defined by

$$C = \left\{ \frac{1}{2}, \frac{1}{2} \right\} \cup \left\{ \frac{1}{2} + \frac{1}{2n} \mid n \geq 1 \right\} \quad \text{and} \quad s_n(x) = \begin{cases} \frac{1}{2} + \frac{1}{2n} & x \geq \frac{1}{2} \\ 0 & x < \frac{1}{2} \end{cases}$$

where $s_n$ is a self-map on $C$ for every $n \geq 1$. For $n = 0$, $s_n(x) = 0$ for every $x \in C$.

First we prove that for every $n \in \mathbb{N}$, $s_n$ is a space function. From Def. 6 we prove:
1. 1/2 is compact. This follows directly from the definition of our constraint system \((C, \leq)\).

2. \(1/2 \geq D_I(1/2)\). By definition of \(D_I\), \(D_I(1/2) \leq s_n(1/2)\) for every \(n \in I\). Then \(D_I(1/2) \leq \bigcap_{n \geq 1} s_n(1/2)\). Since for every \(n \geq 1\), \(s_n(1/2) = 1/2 + 1/2n \geq 1/2\) then \(\bigcap_{n \geq 1} s_n(1/2) = 1/2\). Thus, \(1/2 \geq D_I(1/2)\).

3. \(1/2 \not\geq D_N(1/2)\) for any finite set \(N \subseteq I\). Notice that, for any finite set \(N \subseteq I\), \(D_N(1/2) = s_m(1/2) > 1/2\), where \(m = \max(N)\). Hence, \(1/2 \not\geq D_N(1/2)\).
4.9. Distributed Spaces in Completely Distributive Lattices

In this section we present another characterization of distributed spaces for distributive lattices: For a group \( I \), \( \mathbb{D}_I(c) \) can be understood as the greatest information below all possible combinations of information in the spaces of the agents in \( I \) that derive \( c \). We also provide compositionality properties capturing the intuition that just like distributed information of a group \( I \) is the collective information from all its members, it is also the collective information of its subgroups. We shall argue that the following results can be used to produce algorithms to efficiently compute \( \mathbb{D}_I(c) \) for finite constraint systems.

We recall \( J \)-tuples, a general form of tuples that allows for an arbitrary index set \( J \).

**Definition 14 ([18]).** Let \( J \) be an index set. Given a set \( X \), a \( J \)-tuple of elements of \( X \) is a function \( x : J \to X \). If \( j \in J \), we denote \( x(j) \) by \( x_j \) and refer to it as the \( j \)-th coordinate of \( x \). The function \( x \) is denoted itself by \((x_j)_{j \in J}\). We use \( X^J \) to denote the set of all \( J \)-tuples.

The next theorem is one of main results of this paper. It establishes that for completely distributive lattices, \( \mathbb{D}_K(c) \) is the greatest information below all possible combinations of information in the spaces of the agents in \( K \) that derive \( c \).

**Theorem 5.** Let \( (\mathbb{D}_I)_{I \subseteq G} \) be the distributed spaces of an scs \((C, \sqsubseteq, (s_i)_{i \in G})\). Suppose that \((C, \sqsubseteq)\) is completely distributive. Let \( \mathbb{D}_K : C \to C \), with \( K \subseteq G \), be the function defined as follows:

\[
\mathbb{D}_K(c) \equiv \bigcap \left\{ \bigcup_{k \in K} s_k(a_k) \mid (a_k)_{k \in K} \in C^K \text{ and } \bigcup_{k \in K} a_k \sqsubseteq c \right\}.
\]

Then \( \mathbb{D}_K = \mathbb{D}_K \).

The above theorem is presented in [3] for finite cs. Here we extend this for completely distributive lattices.

For the sake of the presentation, we give the proof of the above theorem in Section 6. Nevertheless, we would like to mention that the central and non-obvious property used in the proof is that of \( \mathbb{D}_K \) being a continuous function. The distributivity of \((C, \sqsubseteq)\) is crucial for this. In fact without it the equality \( \mathbb{D}_K = \mathbb{D}_K \) does not necessarily hold as shown by the following counter-example.

**Example 11.** Consider the lattice \( M_3 \), which is not (completely) distributive, and the space functions \( s_1 \) and \( s_2 \) in Fig. 4b. We obtain \( \mathbb{D}_I(b \sqcup c) = \mathbb{D}_I(e) = a \) and \( \mathbb{D}_I(b) \sqcup \mathbb{D}_I(c) = b \sqcup a = b \). Then, \( \mathbb{D}_I(b \sqcup c) \neq \mathbb{D}_I(b) \sqcup \mathbb{D}_I(c) \), i.e., \( \mathbb{D}_I \) is not a space function.

We can use Th.5 to prove the following properties characterizing the information of a group from that of its subgroups.

**Theorem 6.** Let \( (\mathbb{D}_I)_{I \subseteq G} \) be the distributed spaces of an scs \((C, \sqsubseteq, (s_i)_{i \in G})\). Suppose that \((C, \sqsubseteq)\) is completely distributive. Let \( I, J, K \subseteq G \) be such that \( I = J \cup K \). Then the following equalities hold:

1. \( \mathbb{D}_I(c) = \bigcap \{ \mathbb{D}_J(a) \sqcup \mathbb{D}_K(b) \mid a, b \in C \text{ and } a \sqcup b \sqsubseteq c \}. \)  \\
2. \( \mathbb{D}_J(c) = \bigcap \{ \mathbb{D}_J(a) \sqcup \mathbb{D}_K(a \to c) \mid a \in C \}. \)  \\
3. \( \mathbb{D}_K(c) = \bigcap \{ \mathbb{D}_J(a) \sqcup \mathbb{D}_K(a \to c) \mid a \in C \text{ and } a \subseteq c \}. \)
(a) For $I = \{1, 2\}$, $\sigma_I(c) = \bigcap_{i \in I} s_i(c)$ is not a space function: $\sigma_I(p \lor \neg p) \neq \sigma_I(p) \lor \sigma_I(\neg p)$.

(b) For $I = \{1, 2\}$, $s_1(\rightarrow)$ and $s_2(\rightarrow)$ are space functions. The function $\Box(s \rightarrow)$ in Th. 3 is not a space function: $\Box_I(b) \lor \Box_I(c) = b \neq a = \Box_I(b \lor c)$.

Figure 4: Counter-examples over lattice $M_2$ (a) and non-distributive lattice $M_3$ (b).

We find it convenient to give the proof of Th. 6 in Section 6. The properties in this theorem bear witness to the inherent compositional nature of our notion of distributed space. The first property in Th. 6 essentially reformulates Th. 5 in terms of subgroups rather than agents. It can be proven by replacing $D_J(a)$ and $D_K(b)$ by $D_J(a)$ and $D_K(b)$, defined in Th. 5 and using distributivity of joins over meets. The second and third properties in Th. 6 are pleasant simplifications of the first one using Heyting implication. These properties realize the intuition that by joining the information $a$ and $a \rightarrow c$ of their subgroups, the group $I$ can obtain $c$.

In Section 6 we use Th. 5 to prove Th. 6. We now conclude this section with a brief discussion on how to use Th. 5 to solve a computational lattice problem.

Computing Distributed Information. Let us assume that $C$ is finite and distributive. We wish to compute $\Box_I$. Notice that under this finiteness assumption, space functions are exactly those that preserve the join of finite sets, also known as join-endomorphisms \[17\]. Recall that $S(C)$ denotes the set of space functions (join-endomorphism in this case) over $C$.

From Remark 4 computing the distributed space $\Box_I$ is then equivalent to the following lattice problem: Given a finite set $S = \{s_i \mid i \in I\}$ of join-endomorphisms over the finite distributive lattice $C$, find its meet $\bigcap_{s \in S(C)} S$. Even in small lattices with four elements and two space functions, finding $\Box_I = \bigcap_{s \in S(C)} S$ may not be immediate, e.g., consider $S = \{s_1, s_2\}$ in Fig. 2b.

A naive approach would be to compute each $D_I(c)$ by taking the point-wise meet construction $\sigma_I(c) = \bigcap_{i \in I} s_i(c) \mid i \in I$. But this does not work in general since $\bigcap_{i \in I} \{s_i \mid i \in I\}(c)$ is not necessarily equal to $\sigma_I(c)$. In fact $\sigma_I \nsubseteq D_I$ but $\sigma_I$ may not even be a space function as shown in Fig. 4a.

A brute force solution to computing $D_I(c)$ is to generate the set $\{f(c) \mid f \in S(C)\} \cap s_i$ for all $i \in I$ and then take its join (see Remark 4). This approach works since for any set $S$ of join-endomorphisms $(\bigcup_{S(C)} S)(c) = \bigcup_{S(C)} \{f(c) \mid f \in S\}$. The problem, however, is that the number of
join-endomorphisms over a distributive lattice can be non-polynomial in the size of the lattice.

Nevertheless we can use Th. 5 to obtain a worst-case polynomial bound for computing $D_I$. The next proposition shows this.

**Proposition 6.** Let $C$ be a distributive lattice of size $n$. Let $S = \{s_i \mid i \in I\}$, where $I = \{1, \ldots, m\}$, be a set of join-endomorphisms over $C$. Assuming that binary meets and joins over $C$ can be computed in $O(1)$, the meet $D_I = \bigcap_{s_i \in C} S$ can be computed in $O(mn^3)$ worst-case time complexity.

**Proof.** Let $C$ be a distributive cs (lattice) of size $n$ and let $s_j$ and $s_k$ be space functions (join-endomorphisms) over $C$. From Th. 5, the value $D_K(c) = (s_j \cap s_k)(c) = D_K(c)$, with $K = \{j, k\}$, can be computed in $O(n^3)$ by performing $O(n^2)$ joins and $O(n^2)$ meets. Hence the function $D_K$ can be computed in $O(n^3)$ whenever $|K| = 2$.

We now proceed by induction on the size of $S = \{s_i \mid i \in I\}$ where $I = \{1, \ldots, m\}$. Suppose that $m = 1$. We can then compute $D_{\{1\}} = s_1$ in $O(n)$ and hence in $O(n^3)$. Assume that $D_{\{1, \ldots, m-1\}}$ can be computed in $O((m-1)n^3)$. From the associativity of the meet operation, $D_I = D_{\{1, \ldots, m-1\}} \cap C(I)$, $S_m$. Thus, we can compute first $D_{\{1, \ldots, m-1\}}$ in $O((m-1)n^3)$ and then $D_K$, with $s_j = D_{\{1, \ldots, m-1\}}$ and $s_k = s_m$, in $O(n^3)$. The total worst-case time complexity for computing $D_I$ is then in $O(mn^3)$. □

### 4.10. Summary of Section 4

In this section we presented the main technical results of this paper. We have formalized and developed the theory of the collective information of a group of agents $I$ as the space function $D_I$. Intuitively, the space function $D_I$ represents the smallest space that includes all the local information of the agents in $I$.

We first constructed the complete lattice $(S(C), \subseteq)$ (Lemma 1) where $S(C)$ is the set of all space functions defined on the complete lattice $(C, \subseteq)$. We then defined $D_I$ as the greatest space function in $S(C)$ below the space functions of agents in $I$ (Def. 11) and presented some of its basic compositional properties (Th. 1). We showed that $D_I$ could also be alternative defined as the greatest group distribution candidate (gdc) (Th. 2). We illustrated in Ex. 8 that $D_I$ can be interpreted as Distributed Knowledge in Aumann structures, a representative model for epistemic group reasoning.

We also defined agent, join and group projections (Def. 8 Def. 13). Group (agent) projection of a given $c \in C$ represents the join of all the information that the group (agent) has in $c$. Join projections are the join of individual agent projections. We stated that group projections and distributed spaces form a Galois connection (Prop. 6). We then provided a group compactness result: Given an infinite group $I$, we identified join-derivability (Def. 8) as a condition under which $c \sqsupseteq D_I(c)$ implies $c \sqsupseteq D_J(c)$ for some finite group $J \subseteq I$ (Th. 3). We then showed that without this condition we cannot guarantee the existence of such finite set $J \subseteq I$ (Th. 4).

Finally we showed that if $C$ is completely-distributive, $D_I(c)$ can be characterized as the greatest information below all possible combinations of information in the spaces of the agents in $I$ (Th. 5) that derive $c$ and, more succinctly, as the combination of the information of its subgroups (Th. 6) that derive $c$. For the finite-case we briefly explained how Th. 5 can be used to compute distributed space functions in polynomial time.

### 5. Applications to Minkowski Addition and Mathematical Morphology

In this section we shall show that some fundamental operations from Mathematical Morphology (MM) have a counterpart in the theory we developed in the previous sections. In particular we
shall show that distributed spaces, the central notion of this paper, have a natural interpretation in MM. Furthermore, we shall use our results on distributed information to provide new constructions and results for MM.

5.1. Modules

We assume that the reader is familiar with basic concepts of abstract algebra [20, 21]. To present the results in this section uniformly we shall use a fundamental structure from algebra, namely, that of a module. Recall that a module $M$ over a ring $R$ is a generalization of the notion of vector space. In a vector space the ring $R$ needs to be a field. We shall take the liberty of referring to the elements of $M$ and $R$ as vectors and scalars, resp. The former will be written in boldface to distinguish them from the latter.

More precisely, a module $M$ over a ring $R$ [21], also called an (left) $R$-module $M$, is a set that satisfies the following three conditions. It must be closed under addition and scalar multiplication: $u + v \in M$ and $ru \in M$ whenever $u, v \in M$ and $r \in R$. It must also form an abelian group under addition: $+$ is a commutative and associative operator with $0$ as additive identity and with an additive inverse $-u$ for every $u \in M$. Finally, it must also satisfy the following axioms for scalar multiplication: For every $u, v \in M$ and every $r, s \in R$, $r(su) = (rs)u$, $r(u + v) = ru + rv$, and $(r + s)u = ru + su$.

We shall use the following basic properties of modules. The additive identity for any module $M$ and the additive inverse for every $u \in M$ are unique. If $R$ is a field then the $R$-module $M$ is a vector space over $R$. If $M$ is a vector space over $R$ then $1$ is the only multiplicative identity for $M$.

The following examples of modules are fundamental in Mathematical Morphology. One of them is not a vector space; it justifies using modules rather than vector spaces as the underlying structure.

Example 12. The set $R^n$ of all $n$-tuples of elements of a ring $R$ can be made into an $R$-module. Given $u = (p_1, \ldots, p_n) \in R^n$, $v = (q_1, \ldots, q_n) \in R^n$ and $r \in R$, define $u + v = (p_1 + q_1, \ldots, p_n + q_n)$ and $r \cdot v = (rp_1, \ldots, rp_n)$. The module additive identity $0$ is $(0, \ldots, 0) \in R^n$, where $0$ is the ring additive identity, and the module inverse additive $-u$ is $(-p_1, \ldots, -p_n)$ where $-p_i$ is the ring additive inverse of $p_i$.

The Euclidean $n$-dimensional space $\mathbb{R}^n$ is obtained by taking $R$ as the set of reals numbers $\mathbb{R}$ in the above example. Since $\mathbb{R}$ is a field, $\mathbb{R}^n$ is also a vector space. The $n$-dimensional grid $\mathbb{Z}^n$ is obtained by taking $R$ as the set of integers $\mathbb{Z}$. This is an example of a module that it is not a vector space since the ring $\mathbb{Z}$ is not a field.

5.2. Minkowski Addition

In geometry, vector addition is extended to addition of sets of vectors in an operation known as Minkowski addition. From now on we shall omit mentioning the ring of the module when it is unimportant or clear from the context.

Definition 15 (Minkowski Sum [6]). Let $M$ be a module and $A, B \subseteq M$. The Minkowski addition of $A$ and $B$ is defined thus $A \oplus B = \{u + v \mid u \in A \text{ and } v \in B\}$.

It is easy to see that $\oplus$ is associative and commutative, it has $\{0\}$ and $\emptyset$ as identity and absorbent elements, resp., and that it distributes over set union.

Proposition 7 ([6]). Let $M$ be a module. Then $(\mathcal{P}(M), \oplus)$ is a commutative monoid with zero element $\emptyset$ and identity $\{0\}$. Furthermore, $X \oplus (A \cup B) = (X \oplus A) \cup (X \oplus B)$ for every $X, A, B \subseteq M$. 
Recall that a convex set is a set of points such that, given any two points in that set, the line segment joining them lies entirely within that set. It is well-known that the distribution law $A \oplus (B \cap C) = (A \cap B) \oplus (A \cap C)$ holds for convex sets \[6\]. Nevertheless, in general, $\oplus$ does not distribute over set intersection as illustrated next.

**Example 13.** Take the Euclidean one-dimensional vector space $\mathbb{R}$ and let $X = \{0, 1\}$, $A = \{1\}$ and $B = \{2\}$. One can verify that $\emptyset = X \oplus (A \cap B) \neq (X \oplus A) \cap (X \oplus B) = \{2\}$.

However, as part of the application results in this section we shall establish a pleasant new equation for $X \oplus (A \cap B)$. Namely, for every module $M$ and for all $X, A, B \subseteq M$ we have

$$X \oplus (A \cap B) = \bigcap_{Y \subseteq X} (Y \oplus A) \cup ((X \setminus Y) \oplus B). \quad (4)$$

The Minkowski sum has been applied in mathematical morphology as well as in collision detection, robot motion planing, aggregation theory \[22, 23, 24\]. In this section we focus on applications to mathematical morphology.

### 5.3. Mathematical Morphology

Mathematical morphology (MM) is a theory developed for the analysis of geometric structures \[25\]. It is founded upon, among others, set theory, lattice theory, geometry, topology and probability. Basically, this theory considers an arbitrary space $M$ where its objects are transformed by two fundamental operations: **dilation** and **erosion**.

In \[7\], dilations and erosions are typically defined in terms of Minkowski additions over the modules $\mathbb{R}^n$ or $\mathbb{Z}^n$ given in Ex. \[12\]. Here we generalize the definition in \[7\] to arbitrary modules.

**Definition 16 (Dilations and Erosions in Modules).** Let $M$ be a module. A dilation by $S \subseteq M$ is a function $\delta_S : \mathcal{P}(M) \to \mathcal{P}(M)$ given by $\delta_S(X) = X \oplus S = \bigcup_{u \in S} X \oplus \{u\}$. An erosion by $S \subseteq M$ is a function $\varepsilon_S : \mathcal{P}(M) \to \mathcal{P}(M)$ given by $\varepsilon_S(X) = X \ominus S$ where $X \ominus S \equiv \bigcap_{u \in S} X \ominus \{-u\}$.

In MM, a binary image $X$ is typically represented as a subset of the module $M = \mathbb{Z}^2$ where a pixel is activated if its corresponding coordinate (or position vector) is in $X$. The **translation** of a vector $u$ by a vector $v$ is given by $u + v$. The dilation $\delta_S(X)$ describes the interaction of $X$ with another image $S$ referred to as a **structuring element** and typically assumed to include the center, i.e., $0 = (0, 0) \in S$. The dilated image $\delta_S(X)$ "inflates" the original one by including $X$ and adding the pixels from the translation of every $v$ in $X$ by each $u$ in $S$. Intuitively, $\delta_S(X)$ can be viewed as redrawing the image $X$ with the brush $S$ \[7\]. This is illustrated in Fig. \[5\] where an image $X$ is dilated by the structuring element $S = \{(0, 0), (0, -1)\}$, and in Fig. \[6\] where an image $X$ is dilated by two different structuring elements $A = \{(1, 1), (0, 0), (1, 0), (-1, -1), (0, -1)\}$ and $B = \{(-1, 1), (-1, 0), (0, 0), (0, -1), (1, -1)\}$.

Furthermore, the erosion $\varepsilon_S(X)$ in $\mathbb{R}^n$ or $\mathbb{Z}^n$ can be defined in terms of the translations of $S$ that are contained in $X$. The next proposition states this result for modules.

**Proposition 8.** Let $M$ be a module and $S, X \subseteq M$. Then $\varepsilon_S(X) = \{u \in M \mid S \ominus \{u\} \subseteq X\}$.  

23
Figure 5: Example of Galois connection between dilations and erosions: $\delta_S(X) \subseteq Y$ iff $X \subseteq \varepsilon_S(Y)$. Here $X$ and $Y$ are images, $S$ is the structuring element (centered at the origin), $\delta_S(X)$ is a dilation and $\varepsilon_S(Y)$ is an erosion. Pixels added/removed by dilation/erosion are depicted in gray.

Proof. Let $M$ be a module and $S, X \subseteq M$. From Def. 16 we will prove that $\bigcap_{u \in S} X \oplus \{-u\} = \{u \in M \mid S \oplus \{u\} \subseteq X\}$. Take any $v \in \bigcap_{u \in S} X \oplus \{-u\}$, then for every $u \in S$, $v = w + (-u)$ for some $w \in X$. Since for every $u \in S$, $u + v \in S \oplus \{v\}$ and $u + v = w \in X$ (see Prop. 7), we have $S \oplus \{v\} \subseteq X$. Therefore, $v \in \{u \in M \mid S \oplus \{u\} \subseteq X\}$.

Now, let $v \in \{u \in M \mid S \oplus \{u\} \subseteq X\}$, then $S \oplus \{v\} \subseteq X$. Notice that for any $w \in S$, $w + v \in S \oplus \{v\} \subseteq X$ and therefore $v = (w + v) + (-w) \in X \oplus \{-w\}$ for every $w \in S$. Then $v \in \bigcap_{u \in S} X \oplus \{-u\}$. □

Assume that $S$ includes the center. The above proposition tells us that an erosion $\varepsilon_S(X)$ reduces the image $X$ by erasing the pixels in $X$ whose translation by some element of $S$ is not within $X$. This can also be easily seen from the fact that erosions satisfy the equation $\varepsilon_S(X) = \{u \in M \mid S \ominus \{u\} \subseteq X\}$ for each $v \in S : u + v \in X$ which follows directly from Prop. 8. Fig. 5 illustrates the erosion of an image $Y$ by the structuring element $S$.

5.4. Dilations as Space Functions

We now state that the power set with the usual order and the set of all dilations over it form a (completely distributive) spatial constraint system; i.e., an scs whose underlying lattice is completely distributive.

**Theorem 7.** Let $M$ be a module. Then $(\mathcal{P}(M), \subseteq, (\delta_S)_{S \subseteq M})$ is a completely distributive scs.

**Proof.** The power set of any set ordered by inclusion is a completely distributive lattice with join $\sqcup = \bigcup$ and meet $\sqcap = \bigcap$. Hence $(\mathcal{P}(M), \subseteq)$ is a completely distributive cs.

It remains to prove that for any $S \in \mathcal{P}(M)$, the dilation $\delta_S$ is a space function. From Prop. 4 part (2) it suffices to show that $\delta_S(\bigcup_i A_i) = \bigcup_i \delta_S(A_i)$ for every arbitrary union $\bigcup_i A_i \in \mathcal{P}(M)$. 24
This follows from the following equations:

\[
\delta_S \left( \bigcup_i A_i \right) = \left\{ x + e \mid x \in \bigcup_i A_i \text{ and } e \in S \right\} = \left\{ x + e \mid x \in A_i \text{ for some } i \text{ and } e \in S \right\} = \left\{ x + e \mid x + e \in \delta_S (A_i) \text{ for some } i \right\} = \bigcup_i \delta_S (A_i)
\]

Thus, \((P(M), \subseteq, (\delta_S)_{S \subseteq M})\) is a completely distributive scs. \( \square \)

The above theorem states that dilations are space functions. Erosions, on the other hand, are space projections (see Def. 8).

**Proposition 9.** Let \( M \) be a module. For every \( S \subseteq M \), the function \( \varepsilon_S \) is the \( S \)-projection in the scs \((P(M), \subseteq, (\delta_S)_{S \subseteq M})\).

**Proof.** Let \( M \) be a module and \( S \subseteq M \). To prove that \( \varepsilon_S \) is an \( S \)-projection we show that dilations and erosions form a Galois connection, i.e., for every \( X, Y \subseteq P(M) \), \( \delta_S (X) \subseteq Y \) iff \( X \subseteq \varepsilon_S (Y) \). It is known that a Galois connection determines each function uniquely. Therefore from Prop. 5 (1) it follows that the erosion \( \varepsilon_S \) must then be a projection.

Pick arbitrary \( X, Y, S \subseteq P(M) \). We have \( \delta_S (X) = \bigcup_{v \in S} X \oplus \{ v \} \subseteq Y \) iff for every \( u \in S \), \( X \oplus \{ u \} \subseteq Y \). Furthermore, with the help of the monoid laws for \( \oplus \) (Prop. 7) we can show that for every \( u \in S \):

\[
X \oplus \{ u \} \subseteq Y
\]

iff \( \langle \text{Property of } \subseteq \rangle \)

\[
(X \oplus \{ u \}) \cup Y = Y
\]

iff \( \langle Z = Z \oplus \{ 0 \} \text{ and } \{ 0 \} = \{ -u \} \oplus \{ u \} \rangle \)

\[
(X \oplus \{ u \}) \cup (Y \oplus \{ -u \} \oplus \{ u \}) = Y
\]

iff \( \langle X \oplus (A \cup B) = (X \oplus A) \cup (X \oplus B) \rangle \)

\[
(X \cup (Y \oplus \{ -u \})) \oplus \{ u \} = Y
\]

iff \( \langle \text{Adding } \{ -u \} \text{ by } \oplus \rangle \)

\[
(X \cup (Y \oplus \{ -u \})) \oplus \{ u \} \oplus \{ -u \} = Y \oplus \{ -u \}
\]

iff \( \langle \{ -u \} \oplus \{ u \} = \{ 0 \} \text{ and } \{ 0 \} = \{ u \} \oplus \{ -u \} = \{ -u \} \rangle \)

\[
X \cup (Y \oplus \{ -u \}) = Y \oplus \{ -u \}
\]

iff \( \langle \text{Property of } \subseteq \rangle \)

\[
X \subseteq Y \oplus \{ -u \}.
\]

Clearly for every \( u \in S \), \( X \subseteq Y \oplus \{ -u \} \) iff \( X \subseteq \bigcap_{v \in S} Y \oplus \{ -v \} = \varepsilon_S (Y) \). We have then established that \( \delta_S (X) \subseteq Y \) iff \( X \subseteq \varepsilon_S (Y) \) as wanted. \( \square \)
Figure 6: From left to right, image \( X \), structuring elements \( A, B \) and \( A \cap B \), dilations \( \delta_A(X) \) and \( \delta_B(X) \), and dilation \( D_{(A,B)}(X) \). Structuring elements are centered at the origin. Pixels added by dilation are depicted in gray.

In the proof of the above proposition, we show that dilations and erosions form a Galois connection. This is a known fact in the MM community for \( \mathbb{R}^n \) or \( \mathbb{Z}^n \) space \[26\]. Here we proved a more general version of it for modules.

**Remark 6 (An Epistemic Interpretation of MM operations).** Since we have shown that dilation is a space function, and erosion is a space projection, we can now think of these functions in terms of information available to an agent. Thus, for example in Fig. 5, we can interpret \( X \) as the information that agent \( S \) has, when \( \delta_S(X) \) is the real situation. Similarly, when \( Y \) is the actual state \( \epsilon_S(Y) \) shows what \( S \) perceives. Thus, the agent only perceives some of the real information, and some parts of the actual state of the world are hidden from the agent. If we are considering visual perception, we can consider this as an agent with blurred vision who does not perceive some of the edges of objects.

One may wonder if every space function over \( (\mathcal{P}(M), \subseteq) \) is a dilation \( \delta_S \) for some \( S \subseteq M \). Prop. \([11]\) answers this question negatively. First, we need to introduce a new family of functions over modules.

**Definition 17 (Scale Function).** Let \( M \) be a module over a ring \( R \). Given \( r \in R \), a scale by \( r \) is a function \( s_r : \mathcal{P}(M) \to \mathcal{P}(M) \) defined as \( s_r(X) = \{ ru \mid u \in X \} \).

It is easy to see that \( s_r \) is a space function over \( (\mathcal{P}(M), \subseteq) \).

**Proposition 10.** Let \( M \) be a module over a ring \( R \). Then for every \( r \in R \), \( s_r \) is a space function over \( (\mathcal{P}(M), \subseteq) \).

**Proof.** Let \( M \) be a module over a ring \( R \). From Prop. \([4]\)(2), it suffices to show that given \( r \in R \) and an arbitrary \( \bigcup_i A_i \in \mathcal{P}(M) \), \( s_r(\bigcup_i A_i) = \bigcup_i s_r(A_i) \). Indeed, we have \( s_r(\bigcup_i A_i) = \{ ru \mid u \in \bigcup_i A_i \} = \{ ru \mid u \in A_i \text{ for some } i \} = \{ ru \mid ru \in s_r(A_i) \text{ for some } i \} = \bigcup_i s_r(A_i) \) as wanted.

The following proposition gives us a necessary and sufficient condition for a scale function to be a dilation. In particular, it tells us that we can have infinitely many space functions that are not dilations by some structuring element if the underlying module is, for example, the Euclidean vector space \( \mathbb{R}^n \) or the grid \( \mathbb{Z}^n \) (see Ex. \([12]\)).
Theorem 8. Let $M$ be a module over a ring $R$. Then for each $r \in R$, $s_r = \delta_S$ for some $S \subseteq M$ if and only if $r$ is a multiplicative identity for $M$.

Proof. Suppose that $r$ is a multiplicative identity for $M$. Take $S = \{0\}$. Clearly $s_r = \delta_{\{0\}}$. For the other direction we proceed by contradiction. Let us suppose that $r$ is not a multiplicative identity for $M$ but that there exists $S$ such that $s_r = \delta_S$. By applying both space functions to $\{0\}$, we obtain $s_r(\{0\}) = \{r0\} = \{0\} = \delta_S(\{0\})$. Then for every $v \in S$, $0 + v = 0$, hence $v = 0$. Thus $S = \{0\}$. It follows that for every $u \in M$, $s_r(\{u\}) = \{ru\} = \delta_{\{0\}}(\{u\}) = \{u\}$. This implies that for every $u \in M$, $ru = u$, thus $r$ is a multiplicative identity for $M$, a contradiction.

\[\square\]

5.5. The Distributed Spaces and Dilations

We have shown that dilations are space functions while erosions are space projections. However, the main construction of this paper is that of distributed spaces: The greatest space function below a given set of space functions. The problem we shall address is the following: Given two dilations $\delta_A$ and $\delta_B$, find the greatest space function $D_{\{A,B\}}$ below them. Let us consider some issues regarding this question.

Recall that join and meet operations of the power set $\mathcal{P}(M)$, $\subseteq$ are set union and intersection. Notice that simply taking the point-wise greatest lower bound does not work, i.e., in general, the equation $D_{\{A,B\}}(X) = \delta_A(X) \cap \delta_B(X)$ does not hold.

Example 14. Consider Ex. 13 with the Euclidean one-dimensional vector space $M = \mathbb{R}$, $X = \{0,1\}$, $A = \{1\}$ and $B = \{2\}$. Let $f(Y) = \delta_A(Y) \cap \delta_B(Y)$ for every $Y \subseteq M$. One can verify that $f(\{0\} \cup \{1\}) = \{2\} \neq \emptyset = f(\{0\}) \cup f(\{1\})$, hence $f$ is not even a space function.

Furthermore, notice Prop. 11 tells us that there are space functions over $(\mathcal{P}(M), \subseteq)$ that are not dilations. Thus, in principle it is not clear if $D_{\{A,B\}}$ is itself a dilation and if it is, we would like to identify what its structuring element should be.

The main result of this section, given next, addresses the above issues.

Theorem 8 (Distributed Spaces as Dilations). Suppose that $M$ is a module. Let $(D_S)_{S \subseteq M}$ be the distributed spaces of the scs $(\mathcal{P}(M), \subseteq, \{S\} \subseteq M)$. Then $D_{\{A,B\}} = \delta_{A \cap B}$ for every $A, B \subseteq M$.

Therefore, the greatest space function below two dilations is a dilation by the intersection of their structuring elements. According to our intuition about dilations, the theorem tells us that given $\delta_A$ and $\delta_B$, the dilation $D_{\{A,B\}}$ applied to an image $X$ can be intuitively described as the image obtained by re-drawing $X$ with (the brush) $A \cap B$. This is illustrated in Fig. 6 by showing $D_{\{A,B\}}$ applied to an image $X$.

We now devote the final part of this section to illustrate the application of the theory developed in previous sections to prove the above result.

---

\footnote{The result of $D_{\{A,B\}}(X)$ was computed using the $O(n^2)$ procedure mentioned in the beginning of the proof of Prop. 6 from Section 4.9}
5.6. Application: Proof of Theorem 8

We wish to prove that $D_{\{A,B\}} = \delta_{A\cap B}$ for $A, B \subseteq M$. From Def. 10, we have $\delta_{A\cap B} (X) = X \oplus (A \cap B)$ and, from Th. 6, we have $D_{\{A,B\}}(X) = \bigcap_{Y \subseteq X} (Y \oplus A) \cup ((X \setminus Y) \oplus B)$. Therefore, it suffices to show that

$$X \oplus (A \cap B) = \bigcap_{Y \subseteq X} (Y \oplus A) \cup ((X \setminus Y) \oplus B)$$

for all $X \subseteq M$. Recall that the above equality is the distributivity equation for the Minkowski addition discussed in Eq. 4. It is important also to recall the equation $X \oplus (A \cap B) = (X \oplus A) \cap (X \oplus B)$ does not hold in general. Nevertheless, it does if $X$ is a singleton set as shown next.

Let us consider the singleton case: Suppose that $X$ is an arbitrary set of the form $\{v\}$. We obtain the following equations:

$$D_{\{A,B\}}(\{v\}) = \bigcap_{Y \subseteq X} (Y \oplus A) \cup ((X \setminus Y) \oplus B)$$

$$= (\emptyset \oplus A) \cup (\{v\} \oplus B) \cap (\{v\} \oplus A) \cup (\emptyset \oplus B)$$

$$= (\{v\} \oplus B) \cap (\{v\} \oplus A)$$

$$= \{v + w \mid w \in B\} \cap \{v + w \mid w \in A\}$$

$$= \{v + w \mid w \in A \cap B\}$$

$$= \delta_{A \cap B} (\{v\})$$

Now for the general case we will use the continuity of space functions. Suppose that $X$ is an arbitrary set. From Def. 11 and Th. 7, we know that $D_{\{A,B\}}$ and $\delta_{A \cap B}$ are space functions. Furthermore from Prop. 4, it follows that space functions preserve arbitrary joins. Thus, with the help of this preservation of arbitrary joins (unions) and the singleton case above, we can obtain the desired result in a simple way; namely $D_{\{A,B\}}(X) =$

$$D_{\{A,B\}} \left( \bigcup_{v \in X} \{v\} \right) = \bigcup_{v \in X} D_{\{A,B\}}(\{v\}) = \bigcup_{v \in X} \delta_{A \cap B} (\{v\}) = \delta_{A \cap B} \left( \bigcup_{v \in X} \{v\} \right) = \delta_{A \cap B} (X).$$

5.7. Summary of Section 5

In this section we provided an application of the theory developed in Section 4 to geometry and Mathematical Morphology. First, we recalled the notion of Minkowski addition of sets in vector spaces as an operation in the algebraic structure of modules (Section 5.1) and showed some of its basic properties (Sections 5.2 and 5.3). We then proved that given a module $M$, the structure $(\mathcal{P}(M), \subseteq, (\delta_S)_{S \subseteq M})$ is an ssc where dilations are space functions (Section 5.4). Furthermore, we showed that erosions are space projections (Prop. 5). Since space functions and projections can also be viewed as the information a given agent sees, we then gave a natural epistemic interpretation of these MM operations as an agent’s perception of a given image (Remark 6).

In Section 5.5, we proved that given two dilations $\delta_A$ and $\delta_B$, the distributed space function of the group $\{A, B\}$ (i.e., $D_{\{A,B\}}$) corresponds to the dilation $\delta_{A \cap B}$. Finally, in Section 5.6 we used the theory developed in previous sections to prove Th. 8 and a novel law for $X \oplus (A \cap B)$: i.e., $X \oplus (A \cap B) = \bigcap_{Y \subseteq X} (Y \oplus A) \cup ((X \setminus Y) \oplus B)$.

28
6. Proofs of Section 4.9

This section is devoted to the proofs of the compositionality properties of distributed spaces given in Section 4.9. To simplify our notation, we define sets of \( J \)-tuples whose join derive a given constraint \( c \).

Definition 18. Let \((\mathcal{C}, \sqsubseteq)\) be a cs and \( J \) some index set. For every \( c \in \mathcal{C} \), let \( T^J_c = \{(a_j)_{j \in J} \in S^J | \bigsqcup_{j \in J} a_j \sqsupseteq c\} \). For simplicity, we use \( T_c \) instead of \( T^J_c \) when no confusion arises.

6.1. Proof of Theorem 5

The function \( D_K \) is defined in Th. 5. Here we find it convenient to use the following simplified version.

Proposition 12. \( D_K(c) = \bigcap \{ \bigsqcup_{k \in K} s_k(a_k) | (a_k)_{k \in K} \in T_c \} \).

The following is an immediate consequence of the above definition.

Proposition 13. The function \( D_K \) is monotonic.

Proof. Let \( c \sqsupseteq d \). We have \( \{ \bigsqcup_{k \in K} s_k(a_k) | (a_k)_{k \in K} \in T_c \} \subseteq \{ \bigsqcup_{k \in K} s_k(a_k) | (a_k)_{k \in K} \in T_d \} \). Thus \( D_K(c) \sqsupseteq D_K(d) \).

Next lemma states that \( \overline{D}_K \) is a space function. As pointed out in Section 4.9 the proof of continuity of \( \overline{D}_K \) uses the assumption of \((\mathcal{C}, \sqsubseteq)\) being completely distributive.

Lemma 2. Let \((\mathcal{C}, \sqsubseteq, (s_i)_{i \in G})\) be a ses. Suppose that \((\mathcal{C}, \sqsubseteq)\) is completely distributive. Then, for any \( K \subseteq G \), \( \overline{D}_K \) is a space function.

Proof. Let \((\mathcal{C}, \sqsubseteq, (s_i)_{i \in G})\) be an ses and assume \((\mathcal{C}, \sqsubseteq)\) to be completely distributive. To show that \( \overline{D}_K \) is a space function we prove: (i) it satisfies (S.1) and (S.2) in Def. 6 and (ii) it is continuous.

(i) \( \overline{D}_K \) satisfies (S.1) and (S.2).

For (S.1), one can verify that \( \overline{D}_K(\text{true}) = \text{true} \).

To prove (S.2), it suffices to show that \( \overline{D}_K(c \sqcup d) \sqsubseteq \overline{D}_K(c) \sqcup \overline{D}_K(d) \). The other direction follows by monotonicity (Prop. 13). Consider the following derivation:
(ii) \( \overline{D}_K \) is continuous.

Let \( D \) be any directed set on \( C \), we will prove \( \overline{D}_K(\bigcup D) = \bigcup \{ \overline{D}_K(d) \mid d \in D \} \). We proceed with \( \overline{D}_K(\bigcup D) \subseteq \bigcup \{ \overline{D}_K(d) \mid d \in D \} \). The other direction follows by monotonicity.

By definition of \( \overline{D}_K \), \( \bigcup \{ \overline{D}_K(d) \mid d \in D \} = \bigcup \{ \bigcap \{ s_k(b_k) \mid (b_k)_{k \in K} \subseteq T_d \} \mid d \in D \} \).

Since \( (C, \sqsubseteq) \) is completely distributive (see Def. [1]), for the subset \( \{ \bigcup_{k \in K} s_k(a_k) \mid d \in D, (a_k)_{k \in K} \in T_d \} \)
of $C$, we have

$$\bigcup_{d \in D} \left\{ \bigcap_{(a_k)_{k \in K} \in T_d} \left\{ \bigcup_{k \in K} g_k(a_k) \right\} \right\} = \bigcap_{f \in F} \left\{ \bigcup_{d \in D} \left\{ \bigcup_{k \in K} g_k(f_k(d)) \right\} \right\}$$

where $F$ is the class of choice functions $f$ choosing for each $d \in D$ some index $f(d) \in T_d$. Recall that $f_k(d)$ is the $k$-th element of $K$-tuple $f(d)$. We can rewrite the right-hand side of the above equality using $\sqcup$ properties and the fact that $g_k$ preserves arbitrary joins (see Prop. 4). Then we obtain

$$\bigcup_{d \in D} \left\{ \bigcap_{(a_k)_{k \in K} \in T_d} \left\{ \bigcup_{k \in K} g_k(a_k) \right\} \right\} = \bigcap_{f \in F} \left\{ \bigcup_{k \in K} g_k \left( \bigcup_{d \in D} f_k(d) \right) \right\}.$$ 

We now show that for every $f \in F$, $(\bigcup_{d \in D} f_k(d))_{k \in K} \in T_{\sqcup D}$. Notice that for every $d \in D$, $\bigcup_{k \in K} f_k(d) \supseteq d$. We have

$$\bigcup_{k \in K} \left( \bigcup_{d \in D} f_k(d) \right) = \bigcup_{d \in D} \left( \bigcup_{k \in K} f_k(d) \right) \supseteq \bigcup_{d \in D} d = \bigcup_{d \in D} D.$$ 

Therefore $(\bigcup_{d \in D} f_k(d))_{k \in K} \in T_{\sqcup D}$ (see Def. 18). Then, for every $f \in F$, the element $\bigcup_{k \in K} g_k(\bigcup_{d \in D} f_k(d)) \in \{ \bigcup_{k \in K} g_k(a_k) \mid (a_k)_{k \in K} \in T_{\sqcup D} \}$.

this implies

$$\left\{ \bigcup_{k \in K} g_k \left( \bigcup_{d \in D} f_k(d) \right) \mid f \in F \right\} \subseteq \left\{ \bigcup_{k \in K} g_k(a_k) \mid (a_k)_{k \in K} \in T_{\sqcup D} \right\}.$$ 

Consequently,

$$\bigcap_{f \in F} \left\{ \bigcup_{k \in K} g_k \left( \bigcup_{d \in D} f_k(d) \right) \right\} \supseteq \bigcap_{f \in F} \left\{ \bigcup_{k \in K} g_k(a_k) \mid (a_k)_{k \in K} \in T_{\sqcup D} \right\} = \overline{D}_K \left( \bigcup_{d \in D} D \right).$$

Thus, we conclude $\overline{D}_K$ is continuous.

\[ \square \]

Finally we prove Th. 5. Its statement now is simplified due to Prop. 12.

**Theorem (5).** Let $(\overline{D}_I)_{I \subseteq G}$ be the distributed spaces of an scs $(C, \sqsubseteq, (s_i)_{i \in G})$. Suppose that $(C, \sqsubseteq)$ is a completely distributive lattice. Then $D_K = \overline{D}_K$.

**Proof.** Let $(\overline{D}_I)_{I \subseteq G}$ be the distributed spaces of an scs $(C, \sqsubseteq, (s_i)_{i \in G})$ and assume $(C, \sqsubseteq)$ to be completely distributive.

We divide the proof in two parts: I. $\overline{D}_K \subseteq \overline{D}_K$ and II. $\overline{D}_K \subseteq \overline{D}_K$.
Thus, from (I) and (II), we conclude Th. 5, i.e.,
\[ (6) \]

We now prove the compositional properties of distributed spaces introduced in Th. 6.

• Recall that from Def. 11, \( D_K \) is a space function and hence monotonic. Thus, for every \( (a_k)_{k \in K} \in T_c \), we have \( \bigcup_{k \in K} D_K(a_k) \subseteq \bigcup_{k \in K} D_K(a_k) \). Since \( D_K \) is a space function and monotonic, we know that \( \bigcup_{k \in K} D_K(a_k) = D_K \left( \bigcup_{k \in K} a_k \right) \). Thus, for every \( (a_k)_{k \in K} \in T_c \), we have \( D_K(c) \subseteq \bigcup_{k \in K} D_K(a_k) \), i.e., \( D_K(c) \) is a lower bound of \( S \). Then for every \( c \in C, D_K(c) \subseteq \bigcup S = D_K(c) \). Therefore \( D_K \subseteq D_K \) as wanted.

Thus, from (I) and (II), we conclude Th. 5, i.e., \( D_K = \overline{D_K} \).

6.2. Proof of Theorem 6

We now prove the compositional properties of distributed spaces introduced in Th. 6.

Theorem (6). Let \( (D_I)_{I \subseteq G} \) be the distributed spaces of an scs \( (C, \subseteq, (s_i)_{i \in G}) \). Suppose that \( (C, \subseteq) \) is completely distributive and let \( I, J, K \subseteq G \) be such that \( I = J \cup K \). Then

1. \( D_I(c) = \bigcap \{ D_J(a) \cup D_K(b) \mid a, b \in C \text{ and } a \sqcup b \supseteq c \} \).
2. \( D_I(c) = \bigcap \{ D_J(a) \cup D_K(a \rightarrow c) \mid a \in C \} \).
3. \( D_I(c) = \bigcap \{ D_J(a) \cup D_K(a \rightarrow c) \mid a \in C \text{ and } a \supseteq c \} \).

We present the proof of each item separately.

Proof of Theorem (6) (1). Let \( (D_I)_{I \subseteq G} \) be the distributed spaces of an scs \( (C, \subseteq, (s_i)_{i \in G}) \) where \( (C, \subseteq) \) is completely distributive. Let \( I, J, K \subseteq G \) be such that \( I = J \cup K \).

- \( D_I(c) \subseteq \bigcap \{ D_J(a) \cup D_K(b) \mid a, b \in C \text{ and } a \sqcup b \supseteq c \} \).

Let \( a, b, c \in C \) such that \( a \sqcup b \supseteq c \). Since \( (D_I)_{I \subseteq G} \) is a gcd (see Th. 2) and, both \( J \subseteq I \) and \( K \subseteq I \) hold, for every \( c \in C \), we have \( D_I(c) \subseteq D_J(c) \) and \( D_I(c) \subseteq D_K(c) \). Then \( D_I(c) \subseteq \bigcap D_J(a) \cup D_K(b) \). From the fact that \( D_I \) is a space function and hence monotonic we conclude \( D_I(c) \subseteq \bigcap D_J(a) \cup D_K(b) \). Therefore \( D_I(c) \subseteq \bigcap D_J(a) \cup D_K(b) \). Thus, \( D_I(c) \subseteq \bigcap D_J(a) \cup D_K(b) \) if \( a, b \in C \) and \( a \sqcup b \supseteq c \).

- \( D_I(c) \supseteq \bigcap \{ D_J(a) \cup D_K(b) \mid a, b \in C \text{ and } a \sqcup b \supseteq c \} \).

Let \( c \in C \) and \( S = \bigcup_{i \in I} s_i(c_i) \) if \( \{ (c_i)_{i \in I} \} \in T_c \). We first show the following claim:

Claim. For every \( \bigcup_{i \in I} s_i(c_i) \in S \), there are some \( a, b \in C \) such that \( (i) a \sqcup b \supseteq c \) and \( (ii) \bigcup_{i \in I} s_i(c_i) \supseteq \bigcup_{i \in I} s_i(c_i) \).

Let \( \bigcup_{i \in I} s_i(c_i) \subseteq S \). Since \( I = J \cup K \), we have \( \bigcup_{i \in I} s_i(c_i) = \bigcup_{i \in J} s_i(c_j) \cup \bigcup_{k \in K} s_k(c_k) \). Let \( a = \bigcup_{i \in J} c_i \) and \( b = \bigcup_{k \in K} c_k \). From Th. 5, we have

\[
D_J(a) = \bigcap \left\{ \bigcup_{j \in J} s_j(c_j) \mid (c_j)_{j \in J} \subseteq T_a \right\} \quad \text{and} \quad D_K(b) = \bigcap \left\{ \bigcup_{k \in K} s_k(c_k) \mid (c_k)_{k \in K} \subseteq T_b \right\} .
\]

32
Given that \((C, \sqsubseteq)\) is completely distributive and by associativity of \(\cap\), \(\mathbb{D}_J(a) \cup \mathbb{D}_K(b) = \bigcap R\) where \(R = \left\{ \bigcup_{j \in J} s_j(a_j) \cup \bigcup_{k \in K} s_k(b_k) \mid (a_j)_{j \in J} \in T'_a \text{ and } (b_k)_{k \in K} \in T'_b \right\}\). Clearly \(a \sqcup b \sqsubseteq c\) and \(\bigcup_{i \in I} s_i(c_i) \in R\). Then \(\bigcup_{i \in I} s_i(c_i) \supseteq \bigcap R = \mathbb{D}_J(a) \cup \mathbb{D}_K(b)\). This shows (i) and (ii).

From the above claim and Th. 5 we obtain \(\mathbb{D}_I(c) = \bigcap S \supseteq \bigcap \{\mathbb{D}_J(a) \cup \mathbb{D}_K(b) \mid a, b \in C\text{ and } a \sqcup b \sqsubseteq c\}\) as wanted. \(\square\)

**Proof of Theorem (6 (2)).** Let \((\mathbb{D}_I)_{I \subseteq G}\) be the distributed spaces of an ses \((C, \sqsubseteq, (s_i)_{i \in G})\) where \((C, \sqsubseteq)\) is completely distributive. Let \(I, J, K \subseteq G\) be such that \(I = J \cup K\).

Let \(a, c \in C\). Recall that \(a \to c\) represents the least element \(e \in C\) such that \(a \sqcup e \sqsubseteq c\). Take any \(b \in C\) such that \(a \sqcup b \sqsupseteq c\). Then \(b \sqsupseteq a \to c\) and since space functions are monotonic \(\mathbb{D}_J(a) \cup \mathbb{D}_K(b) \supseteq \mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c)\). From this it follows that \(\bigcap (S \cup \{\mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c)\} \cup \mathbb{D}_J(a) \cup \mathbb{D}_K(b))\) = \(\bigcap (S \cup \{\mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c)\})\) for any \(S \subseteq C\).

From Th. [6] (1) and the above argument, we have

\[
\mathbb{D}_I(c) = \bigcap \{\mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c) \mid a \in C\}. \tag{33}
\]

Thus \(\mathbb{D}_I(c) = \bigcap \{\mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c) \mid a \in C\}\). \(\square\)

**Proof of Theorem (6 (3)).** Let \((\mathbb{D}_I)_{I \subseteq G}\) be the distributed spaces of an ses \((C, \sqsubseteq, (s_i)_{i \in G})\) where \((C, \sqsubseteq)\) is completely distributive. Let \(I, J, K \subseteq G\) be such that \(I = J \cup K\).

Let \(c \in C\) and take any \(a' \not\sqsubseteq c\). It suffices to find \(a \in C\) such that \(a \sqsubseteq c\) and \(\mathbb{D}_J(a') \cup \mathbb{D}_K(a' \to c) \supseteq \mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c)\) since then \(\bigcap (S \cup \{\mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c)\} \cup \mathbb{D}_J(a') \cup \mathbb{D}_K(a' \to c))\) = \(\bigcap (S \cup \{\mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c)\})\) for any \(S \subseteq C\).

Given \(a' \not\sqsubseteq c\) either (a) \(a' \sqsupseteq c\) or (b) \(a' \text{ and } c\) are incomparable w.r.t. \(\sqsubseteq\), written \(a' \parallel c\).

\begin{itemize}
  \item Suppose (a) holds. Then take \(a = c\) thus \(a \to c = true\). By monotonicity we have \(\mathbb{D}_J(a') \cup \mathbb{D}_K(a' \to c) \supseteq \mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c)\) as wanted.
  \item Suppose (b) holds, i.e., \(a' \parallel c\). Notice that \(a' \to c \subseteq c\). By cases, assume \(a' \to c = c\). Then we can take \(a = true\), and thus \(a \to c = c\). By monotonicity we have \(\mathbb{D}_J(a') \cup \mathbb{D}_K(a' \to c) \supseteq \mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c)\) as wanted. Now suppose \(a' \to c \sqsubseteq c\) holds. We can build a poset \(P = \{\{a' \cup c\}, a', c, a' \to c, a' \cap (a' \to c)\}, \subseteq\) which is a non-distributive sub-lattice of \((C, \subseteq)\), isomorphic to a lattice known as \(N_5\) (see Fig. 7). But this contradicts \((C, \subseteq)\) to be distributive (see [5]).
\end{itemize}

From Th. [6] (2) and the above argument, we have

\[
\mathbb{D}_I(c) = \bigcap \{\mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c) \mid a \in C\}.
\]

Thus, \(\mathbb{D}_I(c) = \bigcap \{\mathbb{D}_J(a) \cup \mathbb{D}_K(a \to c) \mid a \subseteq c\}\). \(\square\)
7. Conclusions and Related Work

We have introduced an algebraic theory for reasoning about possibly infinite groups of agents. We have also shown that the theory can be applied to other domains such as geometry and mathematical morphology.

This paper is an extended version of our CONCUR’19 paper [8]. With respect to that work, we have provided significant advances in both the theoretical and practical implications of our notion of distributed spaces in cs. On the theoretical side, we characterized spatial functions as maps preserving arbitrary joins (Prop. [4]). This allowed us to delve into the interpretation of space functions of distributed spaces as meaningful operations in the field of mathematical morphology. We also pursued the study of conditions under which a piece of information derived by the combined local information of an infinite group of agents could be derived by some finite subgroup of those agents. We showed that this is the case when the information inferred by the group from a given supplied piece of information, is itself compact, and derivable from the combination of what each agent can infer from it in the local space (Th. [5]). We further showed, however, that compactness does not hold in general without those conditions (Th. [5]).

Furthermore, we showed a fundamental result that provides a way to compute, for completely distributive lattices, the greatest information that can infer some other given piece of information, and is below all possible combinations of local informations deriving that piece in the spaces of some given group of agents (Th. [5]). This result is presented in [8] for finite cs; in this work we extended it for completely distributive lattices. In [8] we had also stated some properties relating the information of a group of agents w.r.t the information of subgroups of those agents but only for finite cs. In this paper we generalized these for completely distributive lattices (Th. [6]).

Finally, we used the developed theory to investigate applications in mathematical morphology (MM). In this domain, two fundamental operations, dilation and erosion, provide ways to perform geometric transformations, in particular within the realm of image processing based on so-called structuring elements. We considered these MM operations, generalized with Minkowski addition over modules, and used our theory to derive some interesting distribution properties. We also gave the interpretation of maps in group spatial constraint systems as MM operations over structuring elements and showed that the maximum map under two given group distribution maps (seen as dilations) corresponds to the dilation over the intersection of their structuring elements. In so doing
we provided a proof that erosion and dilations for structuring elements that are modules form a Galois connection (Prop. 9). This allowed us to prove that the operation of erosion corresponds to our defined operation of projection of information into the spaces of the structural element (Def. 13). We also discussed an interpretation of dilations and erosions as epistemic as an agent’s perception of a given image (Remark 6).

**Related Work.** The closest related work is that of [5] (and its extended version [27]) which introduces spatial constraint systems (scs) for the semantics of a spatial ccp language. Their work is confined to a finite number of agents and to reasoning about agents individually rather than as groups. We added the continuity requirement to the space functions of [5] to be able to reason about possibly infinite groups. In [13, 28, 29, 30] scs are used to reason about beliefs, lies and other epistemic utterances but also restricted to a finite number of agents and individual, rather than group, behaviour of agents.

Our work is inspired by the epistemic concept of distributed knowledge [31]. Knowledge in distributed systems was discussed in [32], based on interpreting distributed systems using Hintikka’s notion of possible worlds. In this definition of distributed knowledge, the system designer ascribes knowledge to processors (agents) in each global state (a processor’s local state). In [1] the authors present a general framework to formalize the knowledge of a group of agents, in particular the notion of distributed knowledge. The authors consider distributed knowledge as knowledge that is distributed among the agents belonging to a given group, without any individual agent necessarily having this knowledge. In [3] the authors study knowledge and common knowledge in situations with infinitely many agents. The authors highlight the importance of reasoning about infinitely many agents in situations where the number of agents is not known in advance. Their work does not address distributed knowledge but points out potential technical difficulties in their future work.

Complete lattices have been used as a framework to define morphological operators specifically to study grey-level images [26]. In this context, dilations and erosions are defined as operators that preserve arbitrary suprema and infima, resp. This proposal is a generalization of what we studied in Section 3 where we present dilations and erosions by some structuring element. As a novelty, we proposed the scs $((\mathcal{P}(M), \subseteq, (\delta_S)_S \subseteq M)$ where $M$ is a module and $(\delta_S)_S \subseteq M$ are dilations defined on the cs $(\mathcal{P}(M), \subseteq)$. It allowed us to apply our theoretical results to prove MM properties, e.g., that dilations and erosions form a Galois connection (Prop. 9). Also, we provided the interpretation of distributed information for images. Namely, we showed that given two dilations $\delta_A$ and $\delta_B$, the greatest dilation below them is exactly $D\{A, B\}$ which in turn equals $\delta_{A \cap B}$ (Th. 8). As a future work, we plan to explore these results for grey-scale images.

In [19] the authors investigate the cardinality of the set $E(L)$ of all join-endomorphisms of a given lattice $L$. (A join-endomorphism is a self-map that preserves finite joins, hence it is a space function without the continuity requirement.) The authors also provide efficient algorithms to compute the meet of a given set of join-endomorphisms. In this paper, we briefly illustrated the use of Th. 5 to derive a polynomial complexity bound for computing this meet.

There are other group phenomena that are closely related to the group phenomena here studied. In particular, group polarization [33, 34], from social sciences, and group improvisation [35], from computer music. Group polarization refers to the natural tendency of a group to make more extreme decisions than their individuals. Group improvisation involves constraining musical pattern variation choices of a participant according to choices made by others in the group. We plan to study these phenomena in future work by building upon the present work.
References

[1] J. Y. Halpern, Y. Moses, Knowledge and common knowledge in a distributed environment, Journal of the ACM (JACM) 37 (1990) 549–587.

[2] W. Hildenbrand, On economies with many agents, Journal of Economic Theory 2 (1970) 161 – 188.

[3] J. Y. Halpern, R. A. Shore, Reasoning about common knowledge with infinitely many agents, Information and Computation 191 (2004) 1–40.

[4] V. A. Saraswat, M. Rinard, P. Panangaden, The semantic foundations of concurrent constraint programming, in: Proceedings of the 18th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL ’91, ACM, 1991, pp. 333–352. URL: http://doi.acm.org/10.1145/99583.99627

[5] S. Knight, C. Palamidessi, P. Panangaden, F. D. Valencia, Spatial and Epistemic Modalities in Constraint-Based Process Calculi, in: CONCUR 2012 - 23rd International Conference on Concurrency Theory, volume 7454 of Lecture Notes in Computer Science, Springer, 2012, pp. 317–332. URL: https://hal.archives-ouvertes.fr/hal-00761116

[6] R. Schneider, Convex Bodies: The Brunn–Minkowski Theory, Encyclopedia of Mathematics and its Applications, 2 ed., Cambridge University Press, 2013. doi: 10.1017/CBO9781139003858

[7] I. Bloch, H. Heijmans, C. Ronse, Mathematical morphology, in: M. Aiello, I. Pratt-Hartmann, J. Van Benthem (Eds.), Handbook of Spatial Logics, Springer Netherlands, 2007, pp. 857–944.

[8] M. Guzmán, S. Knight, S. Quintero, S. Ramírez, C. Rueda, F. Valencia, Reasoning about distributed knowledge of groups with infinitely many agents, in: W. J. Fokkink, R. van Glabbeek (Eds.), 30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands, volume 140 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 29:1–29:15. URL: https://doi.org/10.4230/LIPIcs.CONCUR.2019.29

[9] B. A. Davey, H. A. Priestley, Introduction to lattices and order, 2nd ed., Cambridge university press, 2002.

[10] S. Abramsky, A. Jung, Domain theory, in: S. Abramsky, et al. (Eds.), Handbook of Logic in Computer Science, volume 3, Oxford University Press, 1994, pp. 1–168.

[11] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, D. S. Scott, Continuous lattices and domains, Cambridge University Press, 2003.

[12] G. Birkhoff, Lattice Theory, Colloquium publications, American Mathematical Society, 1948. URL: https://books.google.com.co/books?id=o4bu3ex9BdkC

[13] S. Abramsky, D. M. Gabbay, T. S. E. Maibaum (Eds.), Handbook of Logic in Computer Science (Vol. 3): Semantic Structures, Oxford University Press, Inc., USA, 1995.

[14] F. S. Boer, A. Di Pierro, C. Palamidessi, Nondeterminism and infinite computations in constraint programming, Theoretical Computer Science (1995) 37–78.
[29] M. Guzmán, S. Perchy, C. Rueda, F. Valencia, Characterizing Right Inverses for Spatial Constraint Systems with Applications to Modal Logic, Theoretical Computer Science 744 (2018). URL: https://hal.inria.fr/hal-01675010

[30] S. Haar, S. Perchy, C. Rueda, F. Valencia, An Algebraic View of Space/Belief and Extrusion/Utterance for Concurrency/Epistemic Logic, in: 17th International Symposium on Principles and Practice of Declarative Programming (PPDP 2015), ACM SIGPLAN, 2015, pp. 161–172. URL: https://hal.inria.fr/hal-01256984

[31] R. Fagin, J. Y. Halpern, Y. Moses, M. Y. Vardi, Reasoning about knowledge, 4th ed., MIT press Cambridge, 1995.

[32] J. Y. Halpern, Using reasoning about knowledge to analyze distributed systems, Annual review of computer science 2 (1987) 37–68.

[33] J.-M. Esteban, D. Ray, On the measurement of polarization, Econometrica 62 (1994) 819–851.

[34] M. S. Alvim, S. Knight, F. D. Valencia, Toward a Formal Model for Group Polarization in Social Networks, in: M. S. Alvim, K. Chatzikokolakis, C. Olarte, F. Valencia (Eds.), The Art of Modelling Computational Systems: A Journey from Logic and Concurrency to Security and Privacy - Essays Dedicated to Catuscia Palamidessi on the Occasion of Her 60th Birthday, volume 11760 of Lecture Notes in Computer Science, Springer, 2019, pp. 419–441. URL: https://hal.archives-ouvertes.fr/hal-02410747

[35] C. Rueda, F. Valencia, On validity in modelization of musical problems by ccp, Soft Computing 8 (2004) 641–648. URL: https://doi.org/10.1007/s00500-004-0390-7
Index

$I$-group derivable, 16
$I$-group projection, 16
$I$-join projection, 10
$J$-tuples, 19
$i$-agent derivable, 10

Agent projection, 10
Arbitrary join preservation, 8
Aumann constraint system, 14

Bottom space, 13

Compact element, 5
Complete lattice, 5
Completely distributive lattice, 6, 19
Consistent, 7
Constraint frame, 7
Constraints, 6
Continuity, 14

D.1, first group distribution candidate axiom, 15
D.2, second group distribution candidate axiom, 15
D.3, third group distribution candidate axiom, 15
Dilation, 23
Directed set, 5
Distributed space, 13
Distribution candidate, 15
Distributive property, 5

Empty information, 7
Erosion, 23

G, group of agents, 7

Galois connection, 16
Group compactness, 17
Group non-compactness, 17
Group projection, 16

Heyting implication, 7
Inconsistent, 7
Increasing chain, 6

Join projection, 10

Mathematical morphology, 23
Maximum group distribution candidate, max gdc, 15
Minkowski addition, 22
Module, 22

Ring, 22

S.1, first space axiom, 8
S.2, second space axiom, 8
Scale function, 26
Self-map, 6
    Arbitrary join preserving, 6
    Continuous, 6
    Monotonic, 6
Space function, 8
    Arbitrary join preservation, 8
    Monotonic, 8
Space function order, 12

Top space, 13

Vector, 22

Whole information, 7
Index of Symbols

\( (P, \sqsubseteq) \), poset, 5
\( (\mathcal{S}(C), \sqsubseteq_s) \), set of space functions ordered by \( \sqsubseteq_s \), 12
\( (x_j)_{j \in J} \), \( J \)-tuple of elements \( x_j \in X \) for each \( j \in J \), 19
\( (C, \sqsubseteq, (s_i)_{i \in G}) \), spatial constraint system, scs, 9
\( R \)-module \( M \), module \( M \) over a ring \( R \), 22
\( C, (C, \sqsubseteq) \), Constraint system, cs, 6
\( \mathcal{D} = (\mathcal{D}_I)_{I \subseteq G} \), distributed spaces, 13
\( \mathcal{D}_I \), distributed space of the group \( I \), 13
\( \mathcal{D}_I(e) \), distributed knowledge of \( e \) in group \( I \), 15
\( K_i(e) \), agent \( i \) knows \( e \), 15
\( \mathcal{M}_3 \), non-distributive lattice, 19
\( \Pi_I(c) \), \( I \)-group projection of \( c \), 16
\( \mathcal{A} = (S, \mathcal{P}_1, \ldots, \mathcal{P}_n) \), Aumann structure, 14
\( \bigvee, \sqcup \), supremum, join, lub, 5
\( \bigvee_P, \sqcup_P \), join in poset \( P \), 5
\( \bigwedge, \sqcap \), infimum, meet, glb, 5
\( \bigwedge_P, \sqcap_P \), meet in poset \( P \), 5
\( \sqsubseteq \), entailment, 5, 6
\( \mathcal{d} = (d_I)_{I \subseteq G} \), group distribution candidate, gdc, 15
\( \delta_S \), dilation by \( S \), 23
\( \sqsubseteq_K \), 19, 29, 31
\( \varepsilon_S \), erosion by \( S \), 23
\( \text{false} \), 5, 6
\( \sqsubseteq_s \), space function order, 12
\( \lambda_\bot \), bottom space, 13
\( \lambda_\top \), top space, 13
\( \mathcal{M}_2 \) square lattice, 10
\( \mathbb{N}_5 \), non-modular lattice, 33
\( \text{max} \), maximum operator, 5
\( \oplus \), Minkowski addition, 22
\( \mathcal{P}_i \), information set, 14
\( \pi_I(c) \), \( I \)-join projection of a group \( I \) of \( c \), 10
\( \pi_i(c) \), \( i \)-agent projection of \( c \), 10
\( s_r \), scale by \( r \), 26
\( \mathcal{S}(C) \), set of space functions, 8
\( s = (s_i)_{i \in G} \), tuple of space functions, 9
\( s_i \), space function, 7
\( \rightarrow \), Heyting implication, 7
\( \text{true} \), 5, 6
\( T_c \), set of \( J \)-tuples s.t. the join of its elements derives \( c \), 29
\( X^J \), set of \( J \)-tuples \( (x_j)_{j \in J} \), 19