LATTICE THEORY IN MULTI-AGENT SYSTEMS

Hans Riess

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2022

Supervisor of Dissertation

Robert Ghrist, Andrea Mitchell University Professor of Mathematics & Electrical and Systems Engineering

Graduate Group Chairperson

Troy Olsson, Associate Professor of Electrical and Systems Engineering

Dissertation Committee

George Pappas, UPS Foundation Professor of Electrical and Systems Engineering
Alejandro Ribeiro, Professor of Electrical and Systems Engineering

for my dad who would

on napkins and on airplanes

teach me algebra

ACKNOWLEDGEMENT

As I listen to “Electric Sunrise” by Plini on repeat, I would like to first thank my advisor, Rob Ghrist, for your selfless stoicism, diligent guidance, and edgy-yet-cerebral musical taste.

Secondly, I would like to thank the funding agencies that made my time at Penn possible: the Defense Advanced Research Projects Agency (DARPA), the Basic Research Office (BRO), the Simons Foundation, the National Science Foundations (NSF). I would also like to thank John Leggett III for endowing the fellowship I received during my first two years at Penn. I am humbled that some of your extreme generosity was bestowed upon me.

I offer thanks to all of Rob’s students, past and present, including Iris Yoon, Huy Mai, Zoë Cooperband, Julian Gould, Darrick Lee, and Miguel Lopez. I especially want to thank Gregok Roerk for helping me appreciate lattice theory and Justin Curry for your mentorship in my undergraduate years Duke. I owe much to Jakob Hansen who introduced me to sheaf Laplacians. Thanks also to other students, especially Mikhail Hayhoe, Alp Aydinoglu, and Michael Sobrepera.

I would like to thank the several faculty members I have worked with at Penn. As I embark on my own postdoc, I only begin to fully appreciate all that you do. I would like to thank postdocs Paige North, Yiannis Kantaros, Alejandro Parada-Mayorga, as well as Dan Guaralnik. I would like to thank Alejandro Ribeiro for instilling upon me your unique perspectives on machine learning and data science. George Pappas, for teaching me Linear Systems and recommending a book.

I would like to thank some of the fellow researchers at other universities which whom I have crossed paths over the years. I owe gratitude to Sanjeevi Krishnan who introduced me to the Tarski Fixed Point Theorem, Vidit Nanda and Mike Lesnik for prolonging the life of the MacPherson Seminar at the Institute, Mike Munger for our continued collaboration on some of the economic implications of this work, Kelly Spendlove and Juan-Pablo Vigneaux for our stimulating mathematical discussion, Gunnar Carlsson for your advice, and Michael Zavlanos for new beginnings.

I would like to thank Erich Prince, for your friendship to me in a city where I once knew no one else. Joe Wolf, for tennis matches at a moment's notice. I would like to thank my parents, for always believing in me. Most of all, I would like to thank my fianceé, Rebecca. You have been incredibly patient with me, and I love you.

ABSTRACT

LATTICE THEORY IN MULTI-AGENT SYSTEMS

Hans Riess

Robert Ghrist

In this thesis, we argue that (order-) lattice-based multi-agent information systems constitute a broad class of networked multi-agent systems in which relational data is passed between nodes. Mathematically modeled as lattice-valued sheaves, we initiate a discrete Hodge theory with a Laplace operator, analogous to the graph Laplacian and the graph connection Laplacian, acting on assignments of data to the nodes of a Tarski sheaf. The Hodge-Tarski theorem (the main theorem) relates the fixed point theory of this operator, called the Tarski Laplacian in deference to the Tarski Fixed Point Theorem, to the global sections (consistent global states) of the sheaf. We present novel applications to signal processing and multi-agent semantics and supply a plethora of examples throughout.

TABLE OF CONTENTS

ACKNOWLEDGEMENT iv

ABSTRACT vi

CHAPTER 1 : Introduction 1

1.1 Multi-Agent Systems 1

1.2 Lattice Theory 3

1.3 Survey of Contributions 7

1.4 Related Work 9

CHAPTER 2 : Lattices 17

2.1 From Relations to Posets 17

2.2 Fixed Points 24

2.3 From Posets to Lattices 28

2.4 Additional Examples 35

CHAPTER 3 : Galois Connections 41

3.1 Defining Properties 41

3.2 Galois Connections from Relations 46

3.3 Integral Transforms 49

3.4 Category Theory 52

CHAPTER 4 : Networks & Signals 59

4.1 Networks 59

4.2 Graph Signals 61

4.3 Representations of (Hyper)networks 65

CHAPTER 5 : Sheaves 67

5.1 Network Sheaves & Cosheaves 67

5.2 Homology & Cohomology 71

5.3 Hodge Laplacians 73

5.4 Examples of Sheaf Laplacians 77

CHAPTER 6 : Lattice-Valued Sheaves 82

6.1 Tarski Sheaves 82

6.2 New Sheaves from Old 84

6.3 New Sup-Lattices from Old 86

6.4 Sections 89

6.5 Sheaf Operations 91

CHAPTER 7 : The Tarski Laplacian 94

7.1 Consensus 94

7.2 Parallel Transport 99

7.3 The Hodge-Tarski Theorem 103

7.4 Towards Operator Theory 107

7.5 Algorithms 110

CHAPTER 8 : Signals 115

8.1 Algebraic Signal Processing 116

8.2 Lattice Signal Processing 122

8.3 Convolutional Neural Networks 135

CHAPTER 9 : Semantics 141

9.1 Temporal Logic 142

9.2 Muli-Agent Logic 152

APPENDICES 169

BIBLIOGRAPHY 177

CHAPTER 1

Introduction

In this thesis we offer a novel approach to understanding multi-agent systems from a lattice-theoretic point of view, a stark contrast to modern approaches relying on spectral graph theory [36], dynamical systems [61], and control theory [34].

1.1. Multi-Agent Systems

A system is something so fundamental it is difficult to define. While we will even not attempt to define systems here, we will define multi-agent systems as a special class of systems consisting of many interacting components called *agents*. Agents are differentiated from ordinary components of systems such as the environment because agents are typically assumed to possess some degree of autonomy, intelligence, or communication capabilities, likely all three. Consider just a few examples of multi-agent systems.

1. *Wireless Communication.* Communication systems are naturally multi-agent systems. Agents send/receive *messages*. More elaborate multi-agent systems usually rely on a communication subsystem in order to gain information from other agents in the system.

2. *Swarm robotics.* Multiple (usually simple) robots (e.g. quadrotor UAVs) collaborate to perform tasks such as surveillance, payload transportation, and more [1]. Swarms, it is widely held, outperforms smaller groups of more sophisticated robots due to advantages of scalability and robustness.

3. *Sensor networks.* Multiple agents, equipped with one or more sensors, gather information about the environment or other agents inside or outside of the system. Applications include wearable devices, threat detection, industrial monitoring, and environmental conditions.

4. *Firms.* Firms compete with directly with other firms, exchanging information via

prices as well as through other methods such as collusion, acquisition, or mergers.

In multi-agent systems, heterogeneous streams of information, possibly encoded in various types of data structures, are collected, processed, and exchanged among the various agents in the system. Consider the following examples of information processing tasks performed by multi-agent systems.

1. *Consensus.* Consensus is a broad category of information processing that aims for every agent to eventually come to an agreement on a particular state or quantity of interest that depends on the state of all the agents. Some examples include flocking [157] and distributed formation control [54] in swarm robotic systems, synchronization, and rendezvous in space [40] and time [122]. A consensus algorithm, also called a consensus protocol, is a rule of interaction between agents specifying the exchange of information between a given agent and all of its neighbors in the network [124].

2. *Collaborative Filtering.* Collaborative filtering is a general technique of filtering information from multiple sources facilitated by "interactions" between agents. The most popular use-case of collaborative filtering is recommendation systems which take into account both the preferences of an individual agent as well as similar agents, neighbors, in a graph. To this end, graph signal processing [126] has shown to be an effective tool for rating prediction [84].

3. *Information Fusion.* Information fusion, also known as *data fusion*, is the task of integrating data from multiple sources in heterogeneous formats. In one view [91], data fusion is the study of the transformation of various sources of information into a representation that can be interpreted by a machine or human.

In each of the above information processing tasks, local information is aggregated in order to obtain global information. Sheaf theory [24, 43], it is our view, is an appropriate context to study such data relationships. Sheaves are data structures supplying mathematical rigor to the soft questions such as *is local data consistent?*, *do globally consistent assignments* *of data even exist?* In the agriculturally inspired nomenclature, a sheaf is a fixation of stalks (e.g. vector spaces, sets, lattices, etc.) to a base (e.g. graph, hypergraph, simplicial complex, manifold). A network sheaf (Definition 5.1) is a sheaf $\underline{\mathcal{F}}$ based over an undirected graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ specified by the data of objects (e.g. vectors paces) over nodes and edges called *stalks*, and maps preserving the structure of objects (e.g. linear transformations) called *restriction maps*. If i and j are nodes and ij is an edge, stalks are denoted $\underline{\mathcal{F}}(i)$, $\underline{\mathcal{F}}(j)$, and $\underline{\mathcal{F}}(ij)$, and restriction maps denoted $\underline{\mathcal{F}}(i \trianglelefteqslant ij) : \underline{\mathcal{F}}(i) \to \underline{\mathcal{F}}(ij)$.

Sheaves facilitate the identification of consistent assignments of data called *global sections* (Definition 5.3). We argue that global sections

$$\Gamma(G; \underline{\mathcal{F}}) = \Big\{ \underbrace{\mathbf{x} \in \prod_{i \in \mathcal{V}_G}}_{\textit{assignments of data}} \mid \underbrace{\underline{\mathcal{F}}(i \trianglelefteqslant ij)(x_i) = \underline{\mathcal{F}}(j \trianglelefteqslant ij)(x_j)}_{\textit{agreement along restriction maps}} \quad \forall ij \in \mathcal{E}_G \Big\}$$

are a natural extension or generalization of consensus.[1] In Chapter 9, we discover a notion of consensus in multi-agent epistemic logic which we could call *semantic consensus*.

1.2. Lattice Theory

Lattices are a class of ordered sets with two “merging” operations called *meet* and *join*. As ordered sets, finite lattices can be visualized with *Hasse diagrams*, directed acyclic graphs (DAGs). The nodes of a Hasse diagram are the elements of the ordered set. If $x \preccurlyeq y$, and there is no other element between x and y, we draw a directed edge from x to y. Orders are elementary, but lattices are less known. There may be a historical explanation.

Lattice theory [18, 45, 70, 140] is a multi-faceted field of mathematics, with a rich history [17]. However, as a field, lattice theory never fully matured, perhaps unjustly so [142]:

> Never in the history of mathematics has a mathematical theory been the object of such vociferous vituperations as lattice theory. Dedekind, Jonsson, Kurosh, Malcev, Ore, von Neumann, Tarski, and most prominently Garrett Birkhoff

[1] If every restriction map is the identity, then we recover consensus exactly.

> have contributed a new vision of mathematics, a vision that has been cursed by a conjunction of misunderstanding, resentment, and raw prejudice.

Lattices, as a mathematical concept, is a relaxation of a boolean algebra [21]. To describe these structures, it was Klein who originally coined a term, *verband*, which loosely translates from German as "association." However, Garret Birkhoff, popularized the English nomenclature [17], "lattice," due to the fact (with some speculation) that the Hasse diagrams of lattices often resemble lattices of the garden variety (pardon the pun!).

We recall this obscure history to highlight the fact that "lattice" has an unfortunate ambiguous meaning. To a physicist, a lattice is (likely) a discreet subset of $\mathbb{R}^d$ (Figure 1.1). To a network-scientist [163], a lattice is (likely) an undirected graph whose nodes uniformly follow a grid pattern and whose edges are drawn according to (unique!) nearest neighbors. We call these objects *integer lattice* or *lattice graphs* to differentiate them from an *order lattices.*

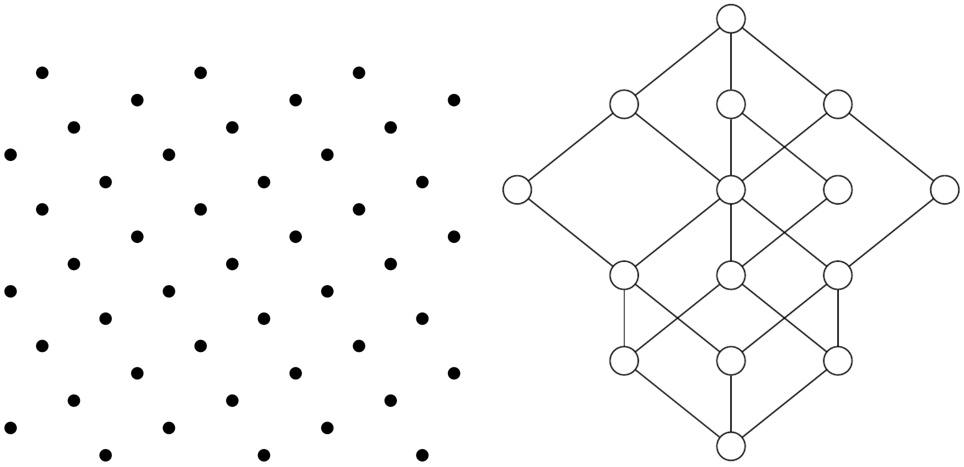


Figure 1.1: (Left) integer lattice; (right) order lattice.

Thus, lattice theory, as a subject, is certainly conflicted in its identity. Brought about as a weakening of the axioms boolean logic (e.g. no longer requiring negation and distributivity), on one hand, lattice theory belongs to logic. An argument could also be made that lattice theory belongs to combinatorics [141]. In our view, lattice theory, while establishing connections between the areas mentioned and more [142], belongs to algebra, the general study of symbol manipulation. For one, a lattice is a universal algebra [144], roughly, a set equipped with a collection of n-ary operations satisfying equational identities. Another compelling reason to highlight the algebraic side of lattice theory is that lattices of subgroups of a group, a collection of symmetries, are perhaps the first example studied in lattice theory. Permutations on n elements form a group. You can reverse permutations by sending j to i, as opposed to sending i to j. Furthermore, you can apply permutations repeatedly and it doesn't matter where you place parentheses. The group of permutations on 4 elements, a seemingly benign structure, has a rather intricate lattice of subgroups (Figure 1.2). Substructures share non-trivial relationships, even in such a simple example.

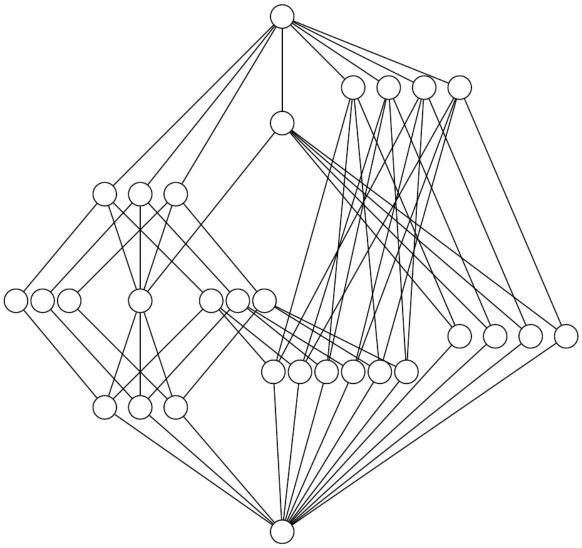


Figure 1.2: Hasse diagram of the lattice of subgroups of S_4.

Lattices function both as ordered sets and algebraic structures. Suppose $x, y \in \mathbf{L}$, an ordered set with $\preccurlyeq$, then $x \vee y$ (join) is the least upper bound (if it exists) and $x \wedge y$ (meet) is the greatest lower bound (again, it exists). On one hand, a lattice is an ordered set such that the least upper bounds and greatest lower bounds are guaranteed to exist (Definition 2.6). On the other hand, a lattice is, a set with binary operations $\vee$ and $\wedge$ satisfying axioms (Definition 2.5). Elementary examples of lattices include the following.

Examples 1.1 (Lattices).

1. Subsets of an arbitrary set with the operations union and intersection. Alternatively, subsets with an inclusion relation.

2. Subgroups (subspaces) of a group (vector space) with product (sum) and intersection (Figure 1.2). Alternatively, subgroups (subspaces) with an inclusion relation.

3. Partitions of an arbitrary set with coarsest common refinement under refinement. Alternatively, partitions under the refinement relation.

4. Truth values $\mathbf{B} = \{0, 1\}$ with `OR`/ `AND`. Alternatively, $\mathbf{B}$ with the order $0 < 1$.

Other quite general examples are embedded in other lattices such as powersets and partitions, including subpartitions (Example 2.14), and lattices characterizing formal concepts [165] and fuzzy logic [16]. Less known is a "lattice theory of information" in which the information content (relative entropy) of a random variable establishes a lattice order. Joins and meets have interesting interpretations (Example 2.15).

This reoccurring duality between order and algebraicity inspires a new outlook on multi-agent systems. Suppose a system consists of $\{1, 2, \ldots, n\}$ agents whose communication patterns are modeled by a graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ with $\mathcal{V}_G = \{1, 2, \ldots, n\}$. Data collected in a lattice by individual agents models relational information about the system. On the other hand, because of the duality, this data is amalgamated with data stored by neighboring agents via the binary operations, meet and join.

As a point of clarification, it is not our goal to develop a general theory of multi-agent systems, but to make a case that certain mathematical tools are fundamental in modeling multi-agent information systems. To this end, it is our hope that the theory of lattice-valued sheaves (Chapter 6) and sheaf Laplacians (Chapter 7) posited in this manuscript inspire novel information and control systems.

1.3. Survey of Contributions

The *Tarski Laplacian* is the key construction and contribution. The Tarski Laplacian was first introduced [63] in an attempt to establish a Hodge theory and thus define cohomology of cellular sheaves valued in lattices. The difficulty in defining cohomolgoy is that the construction of the cellular sheaf cochain complex initiated by Shepard [152] breaks apart. To make a long story short, meets and joins lack inverses. Nonetheless, some key computational and theoretical construction are still possible in *homological algebra* [69]. Cohomology of mathematical objects such as simplicial complexes, topological spaces, or groups is often helpful in classifying properties of a structure. We would expect cohomology of lattice-valued sheaves would have ties to the behavior of the information systems they model. In the theory of cellular sheaves [43], sheaf cohomology H^0 classifies the global sections. Global sections are identified as a lattice of fixed points via the Tarski Laplacian (Hodge-Tarski Theorem, Theorem 7.1).

Homology of chain complexes in semiexact categories (the category of lattices and Galois connections is semiexact) as well as exact sequences and the snake lemma were established Marco Grandis [69]. We suspect a proper theory of sheaf cohomology of $\mathcal{S}up$-valued sheaves would require either a modified notion of projective or injective resolutions of sheaves [24], and, thus, a notion of projective or injective (complete) lattices. With the program to develop a cohomology theory for lattice-valued sheaves supplanted by novel applications of the Tarski Laplacian, our efforts are contained in the following manuscript [63].

It has been suggested that sheaf theory simultaneously straddles both the axes of algebra/geometry and structure/obstruction [60]. Lattice-valued sheaf theory, we place in the second quadrant of this helpful philosophical model (Figure 1.3). While lattices are pure algebraic structures, in Chapter 7, we introduce geometric notations such as Laplace operators and even parallel transport.

From the standpoint of category theory, lattice-valued sheaves over (incidence) posets are a categorification [8] of sheaves valued in an abelian category. Abelian groups, for instance, are replaced with lattices of subgroups (categories) and homomorphisms are replaced with Galois connections (adjunctions). This fortuitous lifting of classical sheaf theory to lattice theory is not without a price. When a functor from a poset into the category of complete lattices and Galois connections factors through the functor sending an object to its lattice of (normal) subobjects, we cannot, in general, recover the homology of the abelian sheaf, from the lattice-valued sheaf [63, Section 4.3].

The Tarski Laplacian is a local operator that acts on assignments of data to a sheaf of lattices. The Tarski Laplacian is called a "Laplacian" for a number of reasons. Primarily, the Hodge-Tarski Theorem identifies the global sections of a lattice-valued sheaf with fixed points. We offer a deep connection to the graph connection Laplacian [153] based on a notion of parallel transport between local sections of a network sheaf recently introduced [20]. We hope the Tarski Laplacian can establish connections to information geometry [103] and representation theory [95].

Dynamical systems on network sheaves with coefficients in Hilbert spaces (e.g. $\mathbb{R}^d$ or $L_2(\mathbb{R}^d)$) were initiated by Hansen [75, 78, 72]. We introduce a dynamical system

$$\mathbf{x}[t+1] = \qquad L\mathbf{x}[t] \wedge \mathbf{x}[t]$$

called the *heat equation* with a time-varying version called the *gossip equation* (3). Beyond diffusion dynamics, saddle-point dynamics, as well as other local dynamics based on simple aggregation rules are of interest. In the narrow case of approximation of the constant sheaf (Definition 7.2), consensus algorithms are a byproduct of diffusion dynamics driven by the Tarski Laplacian.

We spend the better part of Chapter 8 comparing approaches to lattice signal processing, as well as providing our own. We identify the convolution defined by Püschel [132] as an algebraic signal model and describe the relationship between two algebraic signal models corresponding to signals on lattices. In the final chapter, after discussing a class of networked model checking problems, we present applications of the Tarski Laplacian to semantics, offering novel models of knowledge diffusion/consensus motivated by epistemic logic [52].

1.4. Related Work

We make no attempt at a comprehensive literature review, focusing on a few relevant topics, instead.

1.4.1. Quantum logic

In physics, Birkhoff and von Neumann proposed lattice theory as a logical model for quantum mechanics [19]. Their work is based on the following observation. Suppose H is a Hilbert space, then the set of closed subspaces of H forms an (orthocomplemented) lattice under the closure of subspace sums (join) and intersection (meet). These subspaces are in one-to-one correspondence with projections which von Neumann reasoned could be viewed as quantum observables [161]. However, their approach did not stand the test of time, due to Bell's Theorem being experimentally disproved [6]. Consequently, monoidal categories/string diagrams [37] and linear logic [66], the logic of quantum information theory, are more fashionable these days.

1.4.2. Supermodular games

Economists have applied lattice theory to a particular class of n-person games (multi-agent system). A real-valued function on a lattice is *supermodular* if

$$f(x \vee y) + f(x \wedge y) \geqslant f(x) + f(y) \quad \forall x, y \in \mathbf{L}.$$

Then, a *supermodular game* consists of players $i \in \{1, 2, \ldots, n\}$ each with a lattice of possible strategies $\mathbf{L}_i$ and a utility function

$$u_i(x_i, \mathbf{x}_{-i}) : \prod_{i=1}^{n} \mathbf{L}_i \to \mathbb{R},$$

$$x_i \in \mathbf{L}_i, \quad \mathbf{x}_{-i} \in \prod_{j \neq i} \mathbf{L}_i \quad i \in \{1, 2, \ldots, n\}$$

such that $u_i(-, \mathbf{x}_{-i})$ is supermodular and u_i satisfies an additional property of *increasing differences.* Then, every n-person supermodular game has a Nash equilibrium [159] (see also [160, Chapter 7]). It was later shown by Zhou that if each strategy lattice is complete, the set of Nash equilibrium forms a complete lattice [172]. While pertinent to problems in economics such as Bertrand competition [50], supermodular games have found practical use in wireless communication networks, with applications to energy-efficient power allocation [105], power control [4], and interference compensation [83]. Submodular functions on a lattice, real-valued functions satisfying

$$f(x \vee y) + f(x \wedge y) \leqslant \qquad f(x) + f(y) \quad \forall x, y \in \mathbf{L},$$

and algorithms to minimize them were introduced by Topkis [159]. Unsupervised learning problems such as principal component analysis (PCA) and generalized PCA, have been reformulated as constrained submodular maximization problems on the lattice of spaces of $\mathbb{R}^d$ [119], as well as meta-learning [3].

1.4.3. Discrete event systems

In discrete event systems [31], an area of control theory, Galois connections model the triggering of events. A discrete event system is a transition system (Chapter 9) with discrete states and transitions triggered by events. Linear systems [34] and their properties (e.g. controlability, observability), are well-understood. In general, discrete event systems are less tractable, however, in the past two decades, much headway has been made by representing discrete-event systems with square matrices with elements in the following lattice.

Let $\mathbb{R}_{\max}$ denote the extended reals $\mathbb{R} \cup \{-\infty, \infty\}$. $\mathbb{R}_{\max}$ is a (complete) lattice under the operations max and min which we denote with $\vee$ and $\wedge$. $\mathbb{R}_{\max} \setminus \{-\infty, \infty\}$ has the structure of a group under addition (Definition 2.8). Replacing the "times" with "plus" and "times" in an ordinary euclidean space with "max" or "min" one obtains a *max-plus vector space* $\mathbb{R}^m_{\max}$, an example of a *weighted lattice* [110]. As in ordinary linear algebra, matrix multiplication defines a transformation. If $A \in \mathbb{R}^{m \times m}_{\max}$ is a matrix and $\mathbf{x} \in \mathbb{R}^m_{\max}$ is a vector, then matrix multiplication is defined in two dual ways

$$\begin{aligned}(A \ovee \mathbf{x})_i &= \bigvee_{i=1}^{m} a_{ij} + x_j,\\ (A \owedge \mathbf{x})_i &= \bigwedge_{i=1}^{m} a_{ij} + x_j.\end{aligned}$$

For illustration, consider the following discrete event system. Suppose a set of $\{1, 2, \ldots, m\}$ events begin at time $x_i[t], i \in \{1, 2, \ldots, m\}$ after $t \in \mathbb{N}$ rounds. Suppose, an event j can only begin after another even j has terminated (e.g. ironing a shirt after drying). Then, we say i transitions from j. With considerations of system design, waiting times depend not only on the ongoing event i but also on the target event j. Let a_{ij} denote the duration of time between the start of i and the (subsequent) start of j. If j does not depend on process i, then set $a_{ij} = -\infty$. Then,

$$x_i[t+1] = \bigvee_{j=1}^{m} x_i[t] + a_{ij} \quad i = 1, 2, \ldots, m$$

is the new start time of event i. Let $A \in \mathbb{R}^m_{\max}$ be the matrix with entries a_{ij}, then the global dynamics are written in the following form

$$\mathbf{x}[t+1] = A \ovee \mathbf{x}[t].$$

Suppose in two interacting discrete event systems, events are collected from the same set $\{1, 2, \ldots, n\}$. If $\mathbf{x}[t], \mathbf{y}[t] \in \mathbb{R}^m_{\max}$ are vectors of event times after t rounds of System 1 and System 2, respectively, and $A, B \in \mathbb{R}^{m\times m}$ are matrices of transition times, then the *synchronization problem* asks whether $(\mathbf{x}[0], \mathbf{y}[0])$ can be chosen so that there exists a t_0 such that $\mathbf{x}[t] = \mathbf{y}[t]$ for all $t \geqslant t_0$. Clearly, this is equivalent to solving the system

$$A \varovee \mathbf{x} = \qquad\qquad B \varovee \mathbf{y}$$

for $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^m_{\max} \times \mathbb{R}^m_{\max}$.

Cunninghame-Green & Butkovic proposed a solution to the synchronization problem called the *alternating method* [41]. Suppose $A^\dagger \in \mathbb{R}^{m\times m}_{\max}$ with entries $[a^\dagger_{ij}] = -a_{ji}$. Then, the maps

$$\begin{aligned} \mathbf{x} \mapsto & \qquad\qquad A \varovee \mathbf{x}, \\ A^* \varowedge \mathbf{y} \mapsfrom & \qquad\qquad \mathbf{y} \end{aligned}$$

form a Galois connection (Example 3.1). Each iteration of the alternating method is equivalent to applying the Tarski Laplacian (Chapter 7)

$$L(\mathbf{x}, \mathbf{y}) = \qquad \Big(A^\dagger \varowedge (B \varovee \mathbf{y})\, , B^\dagger \varowedge (A \varovee \mathbf{x})\Big) \tag{1.1}$$

on a particular “sheaf” over the graph $\bullet - \bullet$ (Figure

Synchronization is equivalent to being a section of the sheaf. Therefore, by the Hodge-Tarski Theorem (Theorem 7.1), synchronization is also equivalent to the condition

$$\begin{aligned} A^\dagger \varowedge (B \varovee \mathbf{y}) \succcurlyeq \mathbf{x} \\ B^\dagger \varowedge (A \varovee \mathbf{x}) \succcurlyeq \mathbf{y} \end{aligned}$$

for all $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^m_{\max} \times \mathbb{R}^m_{\max}$.

1.4.4. Consensus

Lattice consensus algorithms in the networked multi-agent setting (Algorithm 1, Algorithm 2, Algorithm 3) are novel, but consensus on lattices, while obscure, broadly construed, is not. While we have discussed several notions of consensus in multi-agent systems, consensus has other connotations in both computer science as well as economics.

Reaching agreement among remote processes (agents) is a fundamental problem in distributed computing. In a consensus protocol, each process $i \in \{1, 2, \ldots, n\}$ receives or seeds an input register x_i, sends and receives messages to other processes, and, eventually, must determine an output register y_i coinciding with the output registers of all other processes. Unfortunately, it has been shown that, roughly, no asynchronous consensus protocol can tolerate (i.e. not affect the output) a single fault. For instance, if a process is unexpectedly removed from the system [55], this would be considered a *fault*.

Now, suppose input and output registers are collected in a lattice (Definition 2.5). Then, lattice agreement is defined to be a decision $\{y_i\}_{i=1}^{n}$ such that $\{y_i\}_{i=1}^{n}$ is a chain and the following inequality holds for all $i \in \{1, 2, \ldots, n\}$

$$\bigwedge_{i=1}^{n} x_i \preccurlyeq \qquad y_i \preccurlyeq \qquad \bigvee_{i=1}^{n} x_i.$$

Lattice agreement, centralized branch-and-bound algorithms, have been shown to satisfy fault-tolerance in a variety of settings [171, 170, 169], unlike lattice consensus.

In the theory of social choice [5], consensus is roughly equivalent to "choosing in groups" [118]. At the highest level of generality, if X is a finite set of alternatives (choices), a consensus rule is a map

$$\chi : \bigcup_{k>1} X^k \to \wp(X) \setminus \emptyset$$

sending a tuple $\boldsymbol{\pi}$ of arbitrary individual choices called a profile to a nonempty set of group choices $chi(\boldsymbol{\pi})$ [13]. Suppose we fix the number of voters (agents) to be n and replace X

with a lattice $\mathbf{L}$, then a *(latticial) consensus function*[2] [97] or *aggregation function* [85, 98] is a map

$$\chi : \mathbf{L}^n \to \mathbf{L}.$$

Lattices frequently arise in consensus settings in social choice. For instance, $\mathbf{L}$ could be the lattice of (transitive, reflexive) preference relations on X or a lattice of choice functions [114], maps $c : \wp(X) \to \wp(X)$ such that $c(A) \subseteq A$ for all $A \subseteq X$. In the nomenclature, a consensus function χ is a *meet-projection* if there is a subset $N \subseteq \{1, 2, \ldots, n\}$ such that $\chi(\boldsymbol{\pi}) = \bigwedge_{i \in N} \pi_i$ and, in particular, a *Pareto consensus function* if $N = \{1, 2, \ldots, n\}$.

A third interpretation of lattice consensus comes about in the general science of classification/taxonomy. In evolutionary biology, phylogenetic trees can be inferred from DNA sequences [121]. Having several candidates for the evolutionary tree of a particular organism, consensus is one approach to determining an aggregate phylogenetic tree [26]. In the same spirit, when an ensemble of hierarchical clustering algorithms produces various candidate dendrograms, consensus on the outputs selects a single representative tree [123].

Last, but not least, max/min consensus aims to compute the aggregate max or min of a signal on the nodes of a network. In some settings, maximum and minimum are approximated [156], or algorithms for reaching consensus on general functions on the inputs were proposed [39]. Max consensus was also framed as a max-plus-linear system [122]. Useful applications of min/max consensus and, arguably, meet/join consensus, include decentralized leader election [22] and minimum-time rendezvous [122].

1.4.5. Applied sheaf theory

Applied sheaf theory likely originated in the 1970s. Sheaves were argued to be an intuitive way of thinking about certain image segmentation problems [10]. In the 1990s, the following audacious claim was made by Goguen, who applied sheaf theory to electrical circuitry [67]:

[2] If such a function is monotone and sends the top (bottom) profile to the top (bottom) element of $\mathbf{L}$, then it has been called a *aggregation function* [23], although "aggregation function" is sometimes synonymous with a general consensus function.

> The sheaf condition appears to be satisfied by the behaviours of all naturally arising systems from computing science. This "Sheaf Hypothesis" is similar to the Church-Turing thesis, that all intuitively computable functions are computable in the precise sense of Turning machines.

Sheaf theory, as a field of applied mathematics, is not yet mature. Some successes include sheaves as a tool for integrating sensors [138], filter design [137], path optimzation and routing [38, 116]. The Greatest triumph of applied sheaf may be an ongoing program in the foundations of topological data analysis [59]. The theory of both persistent homology [89, 43, 108] and Reeb graphs [46] has much benefited from a sheaf-theoretic viewpoint. In most settings, the data category of the sheaf (i.e. what lives "upstairs") is the category of finite-dimensional vector spaces and the space is a graph. In this manuscript, we study sheaves over graphs valued in $\mathcal{S}up$, the data category of complete lattices (Chapter 6). We briefly review adjacent work on sheaves valued in nonabelian categories.

In quantum mechanics, sheaves have been used to model non-locality: mysterious interactions between fundamental particles, presumably occurring faster than the speed of light [2]. Contextuality, a version of non-locality, is formulated as a sheaf of events, a sheaf in the data category of sets. Cohomology is undefinable here, hence, obstructions to locality, computed as cohomology in degree one, cannot be explicitly calculated from the sheaf of events. One approach is to extend the sheaf of events to a sheaf of abelian groups factoring through the free functor sending a set to its free abelian group. one disadvantage of this approach is that sections may appear that weren't there before. Alternatively, Čech cohomology was recently defined for sheaves of semimodules [115].

In our opinion, the most significant development in applied sheaf theory is the introduction of sheaf Laplacians [77]. Thus far, sheaf Laplacians have inspired applications in opinion dynamics [78], distributed optimization [75], sheaf learning [76], and graph neural networks [20, 12].

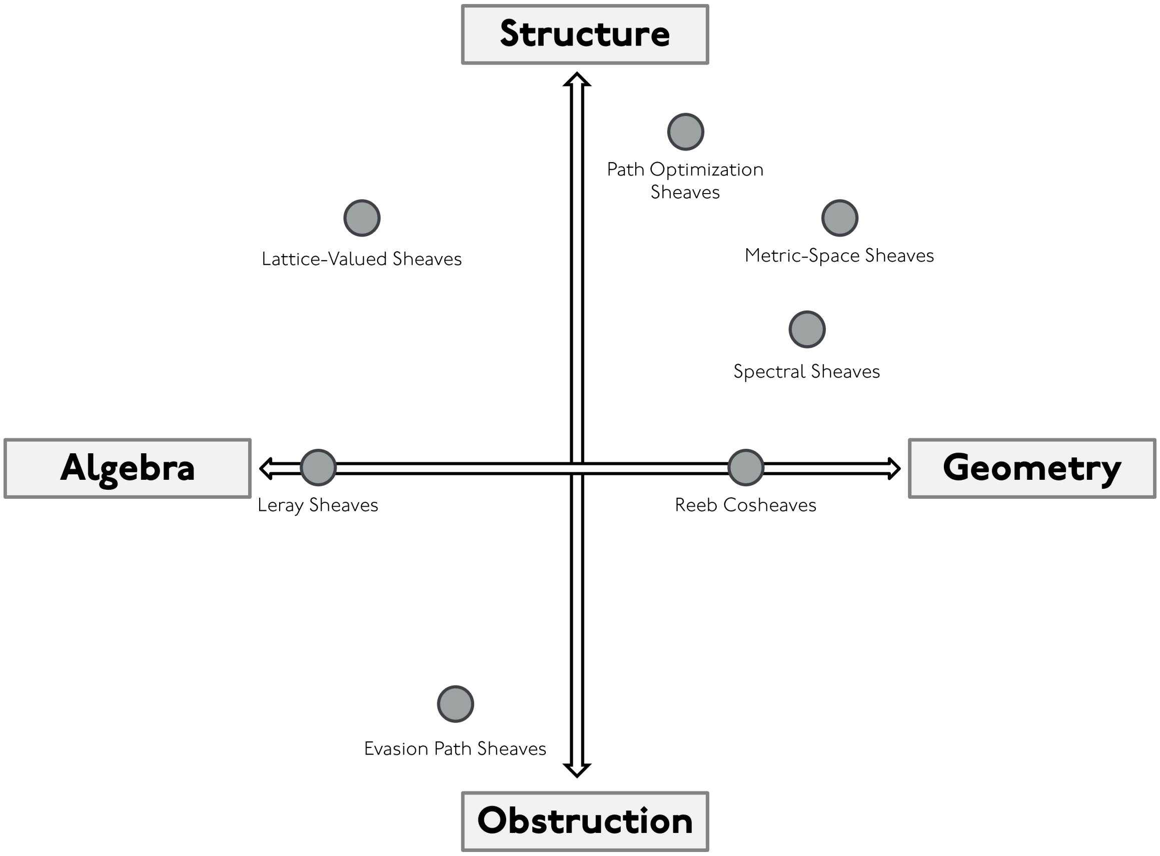


Figure 1.3: A philosophical model of applied sheaf theory: (from left to right) Leray sheaves [44], lattice-valued sheaves [63], evasion path sheaves [62], path optimization sheaves [116], Reeb cosheaves [46], spectral sheaves [77], and (pseudo)-metric space sheaves [139].

$$\begin{tikzcd}
A^{\dagger} \owedge \mathbf{z} & & \mathbf{z} \arrow[ll, mapsto] \arrow[rr, mapsto] & & B^{\dagger} \owedge \mathbf{z} \\
\mathbb{R}^{n}_{\max} \arrow[rr, bend left, "A"] & & \mathbb{R}^{n}_{\max} \arrow[ll, bend left, "A^{\dagger}"] \arrow[rr, bend right, "B^{\dagger}"'] & & \mathbb{R}^{n}_{\max} \arrow[ll, bend right, "B"'] \\
\bullet \arrow[rrrr, dash] & & & & \bullet \\
\mathbf{x} \arrow[r, mapsto] & A \ovee \mathbf{x} & & B \ovee \mathbf{y} & \mathbf{y} \arrow[l, mapsto]
\end{tikzcd}$$

Figure 1.4: A max-plus sheaf modeling synchronization of discrete-event systems [41].

CHAPTER 2

Lattices

2.1. From Relations to Posets

Given sets X and Y, a binary relation is simply a subset $\mathcal{R} \subseteq X \times Y$. Sometimes we write $x\mathcal{R}y$ whenever $(x, y) \in \mathcal{R}$. Suppose $\mathcal{R} \subseteq X \times Y$ and $\mathcal{S} \subseteq Y \times Z$. We can compose relations as follows:

$$\mathcal{S} \circ \mathcal{R} = \{(x, z) \mid \exists\ y \in Y \text{ such that } x\mathcal{R}y,\ y\mathcal{S}z\}.$$

We may also take unions $\mathcal{R} \cup \mathcal{R}'$ and intersections $\mathcal{R} \cap \mathcal{R}'$ of relations. Relations have duals. Suppose $\mathcal{R} \in X \times Y$ is a relation. Then $\mathcal{R}^\dagger$ is the relation $\{(y, x) \mid x\mathcal{R}y\}$.

Example 2.1 (Databases)**.** Suppose X is a set of instances and Y is a set of attributes. We set $x\mathcal{R}y$ if object x has attribute y. Most of the time, attributes are not binary labels. One way to accommodate this situation to convert these labels to integers. Then, a multi-valued relation is a map $\mathcal{R} : X \times Y \to \mathbb{N}$. Then, we define a filtration of relations

$$\mathcal{R}_{\leqslant m} = \{(x, y) \mid \mathcal{R}(x, y) \leqslant m\}. \tag{2.1}$$

This sequence of relations is a filtration in the sense that $\mathcal{R}_{\leqslant m} \subseteq \mathcal{R}_{\leqslant m'}$ whenever $m \leqslant m'$.

Motivated by the preceding example, it is convenient to have notation for the set of attributes that is related to a given object. Let $\mathcal{R} \subseteq X \times Y$ be a relation and $\sigma \subseteq X$. Then, the *intent* of σ, denoted $\sigma^\uparrow$, is the set $\{y \mid x\mathcal{R}y\ \forall\ x \in \sigma\}$. Conversely, the *extent* of a $\tau \subseteq Y$, denoted $\tau^\downarrow$, is the set $\{x \mid x\mathcal{R}y\ \forall\ y \in \tau\}$.

Whenever $X = Y$, we say $\mathcal{R}$ is an *endorelation* (or simply a relation from context). We denote the set of endorelations on X with $\operatorname{Rel} X$. Endorelations are the same as directed graphs. For an $x \in X$, the intent $x^\uparrow$ is sometimes called the *reachability set* or *children* of x. The extent $x^\downarrow$ is called the *predecessor set* or *parents* of x. Several properties of endorelations are of note.

Definition 2.1. Suppose $\mathcal{R} \in \operatorname{Rel} X$. Then, $\mathcal{R}$ is

1. *Transitive* if $x\mathcal{R}y$, $y\mathcal{R}z$ implies $x\mathcal{R}z$,
2. *Reflexive* if $x\mathcal{R}x$,
3. *Anti-symmetric* if $x\mathcal{R}y$ and $y\mathcal{R}x$ implies $x = y$,
4. *Symmetric* if $x\mathcal{R}y$ and $y\mathcal{R}x$,
5. *Serial* if for every $x \in X$, there exist $y \in X$ such that $x\mathcal{R}y$,
6. *Euclidean* if for every $x, y, z \in X$, $x\mathcal{R}y$ and $x\mathcal{R}z$ implies $y\mathcal{R}z$,
7. *Connex* if for every $x, y \in X$, $x\mathcal{R}y$ or $y\mathcal{R}x$.

A *equivalence relation* is a relation $\sim \in \operatorname{Rel} X$ satisfying transitivity, reflexivity and symmetry. It is well known that equivalence relations are in one-to-one correspondence between equivalence relations and partitions. A partition of X is a collection of subsets $\pi = \{X_i\}_{i\in I}$ such that $\bigcup \pi = X$ and $X_i \cap X_j = \emptyset$ for all $i \neq j$. Let $\Pi\,(X)$ denote the set of partitions of X. It is often convenient to represent a partition e.g. $\pi \in \Pi\,(\{1, 2, 3, 4\})$ as $\pi = \{12|3|4\}$.

Example 2.2 (Semantics)**.** This example serves as a prelude to Chapter 9. Suppose S is set of states of a system. For instance, S could be units of time, or, in a multi-agent system, each agent could have local states S_i, for instance, detection states $\{pos, neg\}$ in a sensor network, or message statuses $\{sent, received\}$. In the local paradigm, global states are a prodcut of local states $S = \prod_i S_i$. A *Kripke relation* is an endorelation $\mathcal{K} \in \operatorname{Rel}(S)$. A Kripke relation defines a "possible worlds model" upon which various multi-agent logics [52] are built. For instance, given a state s, the reachability set $s^{\uparrow} = \{t \mid s\mathcal{K}t\}$ constitutes set of states for which a given proposition must be true in order that the proposition be "especially" true at state s.

Definition 2.2. A *poset* ($\underline{\text{p}}$artially $\underline{\text{o}}$rdered $\underline{\text{set}}$) $(\mathrm{P}, \preccurlyeq)$ is a set X with a transitive, reflexive and anti-symmetric binary relation $\preccurlyeq$.

We often omit the order $\preccurlyeq$ in $(\mathrm{P}, \preccurlyeq)$ when it is clear from context. If we remove the anti-symmetry requirement, we say P is a preordered set or *proset*. We write $x \prec y$ if $x \preccurlyeq y$ and $x \neq y$. The ordering $\prec$ is called a *strict order*.

Before getting to far along, we describe a standard way to visualize posets which will require a definition. We say that an element y of P *covers* x written $x \lessdot y$ if there does not exist $z \in \mathrm{P}$ with $x \prec z \prec y$. We form a covering (endo)relation $\lessdot$ depicting the elements that cover each other. We can draw the covering relation as a directed graph called a *Hasse diagram*. For historical reasons, we do not draw arrows because arrows always point from the bottom to the top of the page. Paths in the hasse diagram correspond to chains. Transitivity and reflexivity is implicit. We do not draw self-loops or arrows that realize transitivity, but they are there (Figure 2.1.)

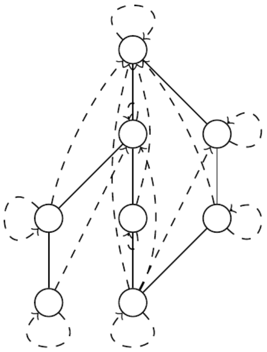


Figure 2.1: Covering relation (Hasse diagram) v.s. entire partial order.

2.1.1. Subsets of posets

Several types of subsets of P are of interest. An *upset* is a subset $U \subseteq \mathrm{P}$ with the property

$$x \in U,\ y \succcurlyeq x \ \Rightarrow\ y \in U.$$

Similarly, a *downset* D is a subset $D \subseteq \mathrm{P}$ with the property

$$x \in D,\ y \preccurlyeq x,\ \Rightarrow\ y \in D.$$

A *principal upset* is generated by a single $x \in \mathrm{P}$: $\uparrow x = \{y \in \mathrm{P} \mid y \succcurlyeq x\}$. A *principal downset* is generated by a single $x \in \mathrm{P}$: $\downarrow x = \{y \in \mathrm{P} \mid y \preccurlyeq x\}$.

An *interval* in a poset P is a subset $I \subseteq \mathrm{P}$ with the convexity property that for every $x, y \in I$, if $x \preccurlyeq z \preccurlyeq y$ in P, then $z \in \mathrm{P}$ also. An interval is *closed* if it is the form $[x, y] = \{z \mid x \preccurlyeq z \preccurlyeq y\}$. We denote the set of closed intervals in a poset $\mathbb{IP}$. A *chain* is a subset $C \subseteq \mathrm{P}$ such that C is totally ordered: for every $x, y \in C$, either $x \preccurlyeq y$, or $y \preccurlyeq x$ (or both). Similarly, an *antichain* is a subset $A \subseteq \mathrm{P}$ such that for every $x, y \in A$, neither $x \preccurlyeq y$ nor $y \preccurlyeq x$ (written $x \parallel y$). The *height* of P is the (path) length of the maximal chain. The *width* of P is the cardinality of the maximal antichain.

A subset $V \subseteq \mathrm{P}$ is *directed* if for every $x, y \in \mathrm{P}$, there is a z such that $z \succcurlyeq x, z \succcurlyeq y$. A subset $F \subseteq \mathrm{P}$ is *filtered* if for every $x, y \in \mathrm{P}$, there is a z such that $z \preccurlyeq x, y$.

2.1.2. Convergence

When computing with lattices, it is useful to have a notion when chains "converge" in both a finite and trans-finite sense.

Definition 2.3. $(\mathrm{P}, \preccurlyeq)$ satisfies the descending chain condition (DCC) if every every chain $C \subseteq P$ has a minimum element. Similarly, P satisfies the ascending chain condition (ACC) if every chain C has a maximum element.

A fortuitous sufficient condition for satisfying DCC (dualy, ACC) is a grading. A poset P is *graded* if there is a map

$$r : \mathrm{P} \to \mathbb{N} \tag{2.2}$$

such that $r(x) < r(y)$ whenever $x \preccurlyeq y$ (strictly monotone) and $r(y) = r(x) + 1$ whenever $x \lessdot y$. A grading on a poset is a particularly restrictive topological sorting of the Hasse diagram. In general, a *topological sorting* [38]assigns a natural number to every node of a directed graph such that the number assigned to the source of an arrow is less than the number assigned to the target.

Examples 2.1 (Grading)**.** Suppose V is a vector space. The lattice of finite-dimensional subspaces is a lattice graded by subspace dimension. Similarly, if X is a set, the lattice of finite subsets is a lattice graded by cardinality. The lattice of (finite) partitions of X is graded by the number of parts.

Proposition 2.1. Suppose P is graded with grading $r : \mathrm{P} \to \mathbb{N}$. Then, P satisfies DCC.

Proof. Consider a chain $C \subseteq \mathrm{P}$. Then, $r(C)$ is a decreasing sequence of natural numbers bounded below by 0. Hence, $r(C)$ converges i.e. has a minimum element. □

Some posets, for instance, the unit interval $[0, 1] \subseteq \mathbb{R}$, do not satisfy the ascending/descending chain condition. For instance, the sequence $\{a\}_{n\in\mathbb{N}} = 1 - 1/n$ is an ascending chain in $[0, 1]$, yet a_{n+1} is strictly greater than a_n for every $n \geqslant 0$. However, the least upper bound of $\{a_n\}$ provides an answer to the question: *what does a_n converge to?* In general, least upper bounds and greatest lower bounds equip posets with notions of convergence as well as algebraic properties.

2.1.3. Completeness

Suppose $S \subseteq \mathrm{P}$ is an arbitrary subset. If they exists, we may compute least upper bounds and greatest lower bounds which we hereafter call *joins* and *meets* respectively:

$$\begin{aligned}\bigvee S = &\qquad \min\{y \mid y \succcurlyeq x \ \forall \ x \in S\};\\ \bigwedge S = &\qquad \max\{y \mid y \preccurlyeq x \ \forall \ x \in S\}.\end{aligned}$$

We say that P is a *complete lattice* if for every subset $S \subseteq \mathrm{P}$, $\bigvee S$ exists. P is *directed complete* if $\bigsqcup V$ exists for every directed subset V.[3] In fact, it is enough that a poset contains either arbitrary meets or arbitrary joins in order to be a complete lattice, explaining the alternative nomenclature *suplattices.*

Proposition 2.2. Suppose $(\mathrm{P}, \preccurlyeq)$ is a poset and $S \subseteq \mathrm{P}$ an arbitrary subset. Then, $\bigvee S$ exists if and only if $\bigwedge S$ exists.

Proof. We write

$$\begin{aligned} \bigvee S = & \qquad \bigwedge \{y \in \mathrm{P} \mid y \succcurlyeq x, \ \forall\, x \in S\} \\ = & \qquad \bigwedge \bigcap_{x \in S} \uparrow x. \end{aligned}$$

□

Warning 2.1. Going forward, we make the following abuse of notation. If P and Q are partial orders, we do not always distinguish between the order relation on P and Q, writing $\preccurlyeq$ for both of them. We also do not distinguish between joins/meets in P and joins/meets in Q. Both we write both with the symbols $\bigvee$ and $\bigwedge$.

2.1.4. Maps between posets

A map f between posets $f : \mathrm{P} \to \mathrm{Q}$ is *order-preserving* or *monotone* if $x \preccurlyeq y$ implies $f(x) \preccurlyeq f(y)$. The set of order-preserving maps between P and Q is a poset with $f \preccurlyeq G$ if $f(x) \preccurlyeq g(x)$ for all $x \in \mathrm{P}$. f is an *order embedding* if $f(x) \preccurlyeq f(y)$ if and only if $x \preccurlyeq y$; an *order isomorphism* if f is a surjective order embedding.

The following example is relevant in explainable AI (artificial intelligence).

Example 2.3 (Binary Classification)**.** Suppose $f_\theta : \mathbb{R}^d \to [0, 1]$ is a statistical model with parameter θ that inputs financial data about an applicant (e.g. income, negative debt, credit score) and outputs the likelihood that an application will not default on a load. In

[3] Of course, there is a dual notion of filtered complete, but, for reasons of tradition, directed complete posets are studied. Also for historical reasons, joins of directed sets are written with $\bigsqcup$.

a supervised training, $f_\theta(\mathbf{x}_i) = 1$ if i made loan payments and or $f_\theta(\mathbf{x}_i) = 0$ if i defaulted. In this scenario, one would desire for f_θ to be monotone to account for the fact that an applicant with a higher income, lower debt, and higher credit score will always be more likely to pay a loan back. This means that $f_\theta(\mathbf{x}) \leqslant f_\theta(\mathbf{x})$ whenever $\mathbf{x} \preccurlyeq \mathbf{y}$.

Posets, and all relations, have a duality baked into their definitions. Given $(\mathrm{P}, \preccurlyeq)$, let $(\mathrm{P}^{\mathrm{op}}, \sqsubseteq)$ be the poset called the *opposite poset* with the same base set P and order $x \sqsubseteq y$ if and only if $x \succcurlyeq y$. Often in order-theoretic arguments, we invoke the *Duality Prinicple* which loosely says *if a statement is true for* P*, then the dual statement obtained by dualizing all the order-theoretic definitions in the statment is true for* P^{op}. As an example, a map f is *order-reversing* or *antitone* if $x \preccurlyeq y$ implies $f(y) \preccurlyeq f(x)$. Alternatively, by the Duality Principle, we could say an antitone map between P and Q is a monotone map $f : \mathrm{P} \to \mathrm{Q}^{\mathrm{op}}$ or, equivalently, a monotone map $f : \mathrm{P}^{\mathrm{op}} \to \mathrm{Q}$.

A map $f : \mathrm{P} \to \mathrm{Q}$ is join-preserving if for every subset $S \subseteq \mathrm{P}$

$$f(\bigvee S) = \qquad\qquad\qquad\qquad \bigvee f(S).$$

A map $f : \mathrm{P} \to \mathrm{Q}$ is meet-preserving if for every subset $S \subseteq \mathrm{P}$

$$f(\bigwedge S) = \bigwedge f(S).$$

If P and Q are directed complete partial orders, then we say $f : \mathrm{P} \to \mathrm{Q}$ is *continuous* if for every directed subset $V \subseteq \mathrm{P}$, $f(\bigsqcup V) = \bigsqcup f(V)$.

Example 2.4 (Computer Vision)**.** Let $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty, +\infty\}$ denote the extended real numbers. Suppose $f : \Omega \to \mathbb{R}_{\max}$ is a grayscale image where Ω is the image domain (e.g. $\mathbb{Z}^2$). In image filtering, it is popular to define filters called erosions and dilations [79]. If is a second image called the *kernel*, define the *dilation* of f by

$$(f \oplus g)(x) = \bigvee_{y \in \Omega} f(x-y) + g(y).$$

Define the *erosion* of f by

$$(f \ominus g)(x) = \bigvee_{y \in \Omega} f(x+y) - g(y).$$

In the lattice of images

$$\mathbb{R}_{\max}^{\Omega} = \{f : \Omega \to \mathbb{R}_{\max}\},$$

the operators

$$\begin{aligned} \delta_g &= f \oplus g \\ \varepsilon_g &= f \ominus g \\ \delta_g, \varepsilon_g &: \mathbb{R}_{\max}^{\Omega} \to \mathbb{R}_{\max}^{\Omega} \end{aligned}$$

satisfy

$$\begin{aligned} \delta_g(\bigvee_i x_i) &= \bigvee_i \delta_g(x_i) \\ \varepsilon_g(\bigwedge_i x_i) &= \bigwedge_i \varepsilon_g(x_i) \end{aligned}$$

for any $g \in \mathbf{L}$. Hence, dilations, as the name suggests, are join-preserving, and erosions are meet-preserving. Other filters called *openings* and *closing* are defined by composing an erosions and dilations and are useful in denoising, contrast enhancement, reconstruction and edge detection [109].

2.2. Fixed Points

Frequently, we study maps $f : \mathrm{P} \to \mathrm{P}$. Such a map is *inflationary* if $f(x) \succcurlyeq x$ for all $x \in \mathrm{P}$ and *deflationary* if $f(x) \preccurlyeq x$ for all $x \in \mathrm{P}$. We write $\mathrm{Pre}(f)$ and $\mathrm{Post}(f)$ for the subsets of deflationary and inflationary points, respectively. Let $\mathrm{Fix}(f)$ denote the set of fixed points $\mathrm{Pre}(f) \cap \mathrm{Post}(f)$ (anti-symmetry axiom).

2.2.1. (Co)closure Operators

A map $f : \mathrm{P} \to \mathrm{P}$ is *idempotent* if $f^2 = f$. Suppose P is a poset and $f : \mathrm{P} \to \mathrm{P}$ is an order-preserving map. There are various ways to characterize $\mathrm{Fix}(f)$ depending on the properties of P and f.

Definition 2.4 (Closure Operators)**.** Suppose $f : \mathrm{P} \to \mathrm{P}$ is a map and P is a poset. If f is order-preserving, inflationary, and idempotent, we say f is a *closure operator*. Conversely, if $g : \mathrm{P} \to \mathrm{P}$ is order-preserving, deflationary, and idempotent we say g is a *coclosure operator*. The fixed points of a (co)closure operator are called *(co)closed*.

The theory of (co)closure operators [140] allows us to explicitly compute joins and meets of the fixed point lattice.

Theorem 2.1 ((Co)closure)**.** Suppose $\mathbf{L}$ is a complete lattice and $f : \mathbf{L} \to \mathbf{L}$ is a closure operator on a complete lattice ($\mathbf{L}$. Then, $\mathrm{Fix}(f)$ is a complete lattice with the following meets and joins

$$\begin{aligned} \bigsqcap_{i \in I} x_i &= \bigwedge_{i \in I} x_i \\ \bigsqcup_{i \in I} x_i &= f\left(\bigvee_{i \in I} x_i\right). \end{aligned}$$

Dually, suppose g is a coclosure operator on $\mathbf{L}$. Then, $\mathrm{Fix}(f)$ is a complete lattice with the following meets and joins

$$\begin{aligned} \bigsqcap_{i \in I} x_i &= f\left(\bigwedge_{i \in I} x_i\right) \\ \bigsqcup_{i \in I} x_i &= \bigvee_{i \in I} x_i. \end{aligned}$$

Proof. See [140, Theorem 3.8]. □

It is not surprising that closure operators come about naturally in topology.

Example 2.5 (Closure Spaces)**.** A *closure space* is a pair (X, c) where X is an (arbitrary) set and c is a closure operator $c : \wp(X) \to \wp(X)$. Closed subsets in X are precisely subsets A with $c(A) = A$. A closure space is *topological* if

$$\begin{aligned} c(A \cup B) = & \quad c(A) \cup c(B), \\ c(\emptyset) = & \quad \emptyset. \end{aligned}$$

Recent work extends persistent homology to new settings (e.g. directed graphs) using closure spaces [27].

While (co)closure operators appear frequently throughout various branches of mathematics, the requirement of idempotency and inflation are often too stringent in practical situations. The following fixed point theorem is integral to our study of operators on lattice-valued sheaves.

Theorem 2.2 (Tarski Fixed Point Theorem)**.** Suppose $\mathbf{L}$ is a complete lattice and $f : \mathbf{L} \to \mathbf{L}$ is an order-preserving map. Then, $\mathrm{Fix}(f)$ is a complete lattice.

We will need a few lemmas.

Lemma 2.1. Suppose $f : \mathbf{L} \to \mathbf{L}$ is order-preserving and $\mathbf{L}$ is complete. Suppose $\{x_i\}_{i \in I}$ is a subset of $\mathbf{L}$ indexed by an arbitrary set I. Then,

$$f\left(\bigwedge_{i \in I} x_i\right) \preccurlyeq \quad \bigwedge_{i \in I} f(x_i).$$

Similarly,

$$f\left(\bigvee_{i \in I} x_i\right) \succcurlyeq \quad \bigvee_{i \in I} f(x_i).$$

Proof of Lemma 2.1. $f\left(\bigwedge_{i \in I} x_i\right)$ is *a* lower bound of each $f(x_i)$ because f is order-preserving and $\bigwedge_i x_i \preccurlyeq x_i$. Moreover, $\bigwedge_{i\in I} f(x_i)$ is *the* greatest lower bound of the subset $\{f(x_i)\}_{i\in I}$, implying $\bigwedge_{i\in I} f(x_i)$ precedes all other lower bounds. Hence,

$$f\left(\bigwedge_{i\in I} x_i\right) \preccurlyeq \qquad\qquad \bigwedge_{i\in I} f(x_i).$$

The second statement follows from the duality principle. □

Lemma 2.2. Suppose $f : \mathrm{P} \to \mathrm{P}$ is order-preserving. Then, $f^{\mathrm{op}} : \mathrm{P}^{\mathrm{op}} \to \mathrm{P}^{\mathrm{op}}$ defined to be the same map on sets, but under the opposite order is order-preserving as well. If f is join-preserving, then f^{op} is meet preserving and vice versa. Furthermore, $\mathrm{Pre}(f^{\mathrm{op}}) = \mathrm{Post}(f)$ and $\mathrm{Post}(f^{\mathrm{op}}) = \mathrm{Pre}(f)$.

Proof of Lemma 2.2. $x \preccurlyeq y$ in P if and only if $x \succcurlyeq y$ in P^{op}. f is order-preserving implies $f(x) \preccurlyeq f(y)$ in P, or $f(x) \succcurlyeq f(y)$ in P^{op}. Hence, f^{op} is order-preserving. Meets in P^{op} are equivalent to joins in P. □

Proof of Theorem 2.2. The classical argument [158] that $\mathrm{Fix}(f)$ is complete has three parts. First, we show $\mathrm{Pre}\, f \subseteq \mathbf{L}$ is an invariant subset under f. Suppose $x \in \mathrm{Pre}(f)$. Then,

$$f(f(x)) \preccurlyeq f(x) \preccurlyeq x$$

which implies $f(x) \in \mathrm{Pre}(f)$ by monotonicity of f and transitivity. Next, we show that $\mathrm{Pre}(f)$ is complete. Let $\{x_i\}_{i\in I} \subseteq \mathrm{Pre}(f)$. Then, Lemma 2.1 implies that $\mathrm{Pre}(f)$ is $\bigwedge$-complete:

$$f\left(\bigwedge_{i\in I} f(x_i)\right) \preccurlyeq \bigwedge_{i\in I} f(x_i)$$

implies $\bigwedge_{i\in I} f(x_i) \in \mathrm{Pre}(f)$. By Proposition 2.2, $\mathrm{Pre}(f)$ is also $\bigvee$-complete. Finally, define

$${f_{|\,\mathrm{Pre}(f)}}^{\mathrm{op}} : \mathrm{Pre}(f)^{\mathrm{op}} \to \mathrm{Pre}(f)^{\mathrm{op}}.$$

Invariance and Lemma 2.2 imply ${f_{|\,\mathrm{Pre}(f)}}^{\mathrm{op}}$ is well-defined and order-preserving on $\mathrm{Pre}(f)^{\mathrm{op}}$ and that $\mathrm{Pre}({f_{|\,\mathrm{Pre}(f)}}^{\mathrm{op}}) = \mathrm{Post}\left(f_{|\,\mathrm{Pre}(f)}\right)$. Now, the argument in the second part shows $\mathrm{Post}\left(f_{|\,\mathrm{Pre}(f)}\right)$ is complete. Furthermore,

$$\mathrm{Fix}(f) = \qquad\qquad\qquad\qquad \mathrm{Post}\left(f_{|\,\mathrm{Pre}(f)}\right)$$

and we are done. □

The proof of Theorem 2.2 implies the following corollaries.

Corollary 2.1. The subsets $\mathrm{Pre}(f)$ and $\mathrm{Post}(f)$ are complete.

Proof. Let $f \wedge \mathrm{id}$ be the map defined $f(x) \wedge x$ on $x \in \mathbf{L}$. Then, $\mathrm{Fix}(f \wedge \mathrm{id}) = \mathrm{Post}(f)$ because $f(x) \succcurlyeq x$ if and only if $f(x) \wedge x = x$. Similarly, $\mathrm{Fix}(f \vee \mathrm{id}) = \mathrm{Pre}(f)$. □

In particular application domains, it is only necessary to show there is a least fixed point. Hence, the Tarski Fixed Point Theorem is sometimes presented as the following corollary.

Corollary 2.2. Suppose $\mathbf{L}$ is a complete lattice and $f : \mathbf{L} \to \mathbf{L}$ is order-preserving. Then, f has a least fixed point x^*.

Proof. $x_* = \bigwedge \mathrm{Fix}(f)$. □

2.3. From Posets to Lattices

Thus far, we have seen that complete lattices are posets with arbitrary meets and joints. We now present an alternative algebraic view of lattices in which we only require binary or nullary (in the case of bounded latttices) meets and joins.

Definition 2.5 (Algebraic Definition)**.** A *lattice* $\mathbf{L}$ is a set L with a pair of binary operations

$$\vee, \wedge : \mathbf{L} \times \mathbf{L} \to \mathbf{L}$$

called *join* and *meet* such that for every $x, y, z \in \mathbf{L}$

1. $x \vee (y \vee z) = (x \vee y) \vee z$; $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ (Associativity),
2. $x \vee y = y \vee z$; $x \wedge y = y \wedge z$ (Commutativity),
3. $x \vee x = x$; $x \wedge x = x$ (Idempotence),
4. $x \wedge (x \vee y) = x$; $x \vee (x \wedge y) = x$ (Absorption).

A *bounded lattice* is a lattice with elements $0, 1 \in \mathbf{L}$ with the axiom

5. $x \wedge 1 = x$; $x \vee 0 = x$

A *semilattice* $\mathbf{S}$ is a set S with a binary operation

$$\wedge : \mathbf{S} \times \mathbf{S} \to \mathbf{S}$$

such that for all $x, y \in \mathbf{S}$

1. $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ (Associativity),
2. $x \wedge y = y \wedge z$ (Commutativity),
3. $x \wedge x = x$ (Idempotence).

A *bounded semilattice* is a semilattice with an element $1 \in \mathbf{S}$ with the axiom

4. $x \wedge 1 = x$.

Example 2.6 (Nesting Poset)**.** Suppose $\mathbb{M}$ is a compact smooth 2-manifold (e.g. a closed and bounded surface in $\mathbb{R}^3$) diffeomorphic to the 2-sphere $\mathbb{S}^2$. Suppose $f : \mathbb{M} \to \mathbb{R}$ is a Morse function [112] factoring through an embedding and a projection onto the z-axis

$$i : \mathbb{M} \overset{i}{\hookrightarrow} \mathbb{R}^3 \xrightarrow{\pi} \mathbb{R}$$

By a version of the Implicit Function Theorem [112, Theorem 2.3], if $t \in \mathbb{R}$ is a regular value, then $f^{-1}(t)$ is a compact manifold of dimension 1. By the well-known classification of such manifolds, it follows there is a diffeomorphism

$$f^{-1}(t) \simeq \bigsqcup_{i=1}^{m} \mathbb{S}^1,$$

and, furthermore, f^{-1} is embedded in $\mathbb{R}^2$. Hence, f^{-1} is a disjoint union of Jordan curve [81], say, $\mathbb{Y} = \gamma_1, \gamma_2, \gamma_m\}$. The complement $\mathbb{R}^2 \setminus \mathbb{Y}$ consists of exactly $m+1$ connected components, with 1 unbounded component and m bounded components. It follows that the set of (path) connected components $\pi_0(\mathbb{M} \setminus \mathbb{Y})$ have the structure of a join-semilattice under the following order [32] equivalent to the circle containment order [145].

We say two components in $\pi_0(\mathbb{M} \setminus \mathbb{Y})$ are adjacent if they they share a boundary. Then, for any pair of adjacent components $c, c' \in \pi_0(\mathbb{M} \setminus \mathbb{Y})$, then $\alpha_i \preccurlyeq \alpha_j$ if and only if the for the corresponding (unshared) boundaries $\gamma_i and \gamma_j$, we have an inclusion of interiors $\mathrm{int}(\gamma_i) \subseteq \mathrm{int}(\gamma_j)$ (existence, by the Jordan Curve Thoerem [81]) or if $\mathrm{int}(\gamma_j)$ is undbounded.

On the other hand, lattices are ordered sets with binary and empty meets and joins.

Definition 2.6 (Combinatorial)**.** A *lattice* is a partially ordered set $\mathbf{L}$ such that for every $x, y \in \mathbf{L}$

$$\begin{aligned} x \vee y = && \min\left\{z \mid z \succcurlyeq x,\ z \succcurlyeq y\right\}, \\ x \wedge y = && \max\left\{z \mid z \preccurlyeq x,\ z \preccurlyeq y\right\}. \end{aligned}$$

A lattice is *bounded* if additionally there are elements $0, 1 \in \mathbf{L}$ such that $0 \preccurlyeq x \preccurlyeq 1$ for all $x \in \mathbf{L}$.

The following proposition allows you to move freely between the order-theoretic and algebraic definitions of lattices.

Proposition 2.3. The algebraic (Definition 2.5) and combinatorial (Definition 2.6) definitions of lattices are equivalent.

Proof. See [140, Theorem 3.17]. □

Consequently, a lattice (bounded) $\mathbf{L}$ has the structure of both a (bounded) *meet-semilattice* $(\mathbf{L}, \wedge)$ and a (bounded) *join-semilattice* $(\mathbf{L}, \vee)$.

There are a few specializations of lattices that we take note. A *distributive lattice* is a lattice with the property that for all $x, y, z \in \mathbf{L}$

$$\begin{aligned} x \wedge (y \vee z) = & \qquad (x \wedge y) \vee (x \wedge z), \\ x \vee (y \wedge z) = & \qquad (x \vee y) \wedge (x \vee z). \end{aligned}$$

A bounded lattice is *complemented* if for every $x \in \mathbf{L}$ there exists an element $y \in \mathbf{L}$ such that $x \vee y = 1$ and $x \wedge y = 0$. Powersets (Example 2.12) are an example of a complemented distributive lattice, also know as a *Boolean lattice.*

A *sublattice* $\mathbf{K} \subseteq \mathbf{L}$ is a subset $K \subseteq \mathbf{L}$ closed under the $\vee$ and $\wedge$ operation in $\mathbf{L}$. A *quasisublattice* of $\mathbf{L}$ is a lattice $\mathbf{K}$ with an order embedding $i : \mathbf{K} \hookrightarrow \mathbf{L}$. Clearly, a sublattice is a quasi-sublattice, but not the other way around. If $\mathbf{K}$ and $\mathbf{L}$ are complete, then we say $\mathbf{K}$ is a *complete quasisublattice.*

Example 2.7 (Lattice of Subgroups)**.** Suppose $(G, \star, e)$ is a group [**?**]. The (normal, arbitrary) subgroups of G form a lattice $\operatorname{Sub} G$ with

$$\begin{aligned} M \vee N = & \qquad \{m \cdot n \mid m \in M,\ n \in N\}, \\ M \wedge N = & \qquad M \cap N. \end{aligned}$$

The lattice $\operatorname{Sub} G$ order embeds into the powerset $\wp(G)$. However, $\operatorname{Sub} G$ is not a sublattice because the join of M and N in $\operatorname{Sub} G$ differs from the join in $\wp(G)$, $M \cup N$.

A lattice *homomorphism* is a map $f : \mathbf{K} \to \mathbf{L}$ such that for all $x, y \in \mathbf{K}$

$$\begin{aligned} f(x \vee y) = && f(x) \vee f(x), \\ f(x \wedge y) = && f(x) \wedge f(y). \end{aligned}$$

A *bounded lattice homomorphism* is a lattice homomorphism with

$$\begin{aligned} f(0) = && 0, \\ f(1) = && 1. \end{aligned}$$

A homomorphism f is an isomoprhimsm if there is another homomorphism $g : \mathbf{L} \to \mathbf{K}$ with $g \circ f = \mathrm{id}$ and $f \circ g = \mathrm{id}$. The reader may check that an order isomoprhism between (bounded) lattices is an isomorphism of (bounded) lattices.

Example 2.8 (Multidimensional Persistence)**.** This example was adapted from recent work on persistence modules over lattice [113]. Given a lattice $\mathbf{L}$ (e.g $\mathbb{N} \times \mathbb{N}$ representing a bifiltration), a *persistence module* is an assignment $F : \mathbf{L} \to \mathcal{V}ec$ of lattice elements to finite-dimensional $\Bbbk$-vector spaces such that $x \preccurlyeq y$ in $\mathbf{L}$ induces a linear transformation $F(x) \xrightarrow{F(x \preccurlyeq y)} F(y)$ (formally, a functor into the category of vector spaces). A "change in basis" of a persistence modules is a lattice homomorpism $f : \mathbf{K} \to \mathbf{L}$. This induces a lattice homomorphism $\hat{f} : \mathbb{I}\mathbf{K} \to \mathbb{I}\mathbf{L}$. Certain interesting functions describing births and deaths of i-dimensional topological features, e.g. in a metric space or filtration of simplicial complexes [59], are written as maps $f_i : \mathbb{I}\mathbf{L} \to \mathbb{Z}$ which map an interval $[a, b] \mapsto \dim(Z_i F(a) \cap B_i F(b))$ where $Z_i F(a)$ and $B_i F(a)$ are the cycles and boundaries of homology [81].

2.3.1. Lattices with multiplicaiton

Quite often, lattices are equipped with an algebraic structure other than binary meets and joins. For instance, the real numbers $\mathbb{R}$ is an (unbounded) lattice that also has the structure of a field. We briefly recall a few definitions.

Definition 2.7 (Monoid)**.** A *monoid* is a tuple $(M, \star, e)$ such that M is an arbitrary set, $\cdot : M \times M \to M$ is an associative binary operation, and $e \in M$ is an element such that $m \star e = e \cdot m = m$ for all $m \in M$. A monoid is a *group* if for every $m \in M$, there exist a unique element $m^{-1} \in M$ such that $m^{-1} \star m = m \star m^{-1} = e$. A monoid is *commutative* if $m \star n = n \star m$ for all $m, n \in M$. A monoid is *idempotent* if $m \star m = m$ for all $m \in M$.

Examples 2.2 (Monoids)**.**

1. The natural numbers with addition $(\mathbb{N}, +, 0)$ is a monoid, but not a group. On the other hand, $(\mathbb{Z}, +, 0)$ is a group.

2. Suppose X is a set, with possible additional structure. Then, the set of maps $f : X \to X$ is a monoid denoted $\mathrm{End}(X)$. The identity e is the idenity map $\mathrm{id} : X \to X$. Multiplication is merely composition of endomorphisms.

3. A bounded semilattice is a monid with $(\mathbf{S}, \wedge, 1)$.

Proposition 2.4. Suppose $(M, \star, e)$ is a commutative idempotent monoid. Then, M is a semilattice.

Proof. Define a relation $m \preccurlyeq n$ if and only if $m \cdot n = m$. First check $(M, \preccurlyeq)$ is a partial order with top element e. Then, check $m \wedge n$ is in fact the element $m \cdot n$. □

Residuated lattices are lattices equipped with a compatible monoid structure.

Definition 2.8 (Residuated Lattice)**.** A *residuated lattice* $(\mathbf{L}, \wedge, \vee, 0, 1, \star, e, [-,-])$ is a bounded lattice $(\mathbf{M}, \wedge, \vee, 0, 1)$ such that $(\mathbf{L}, \cdot, e)$ is a commutative monoid with a second (non-commutative) binary operation $[-,-]$ on $\mathbf{M}$ with the property that

$$x \star y \preccurlyeq z \quad \text{if and only if} \qquad x \preccurlyeq [y, z] \tag{2.3}$$

for all $x, y, z \in \mathbf{L}$. A residuated lattice is *complete* if $\mathbf{L}$ is a complete lattice.

Example 2.9 (Fuzzy Logic)**.** Consider three residuated lattice structures on the unit interval with the lattice structure $([0,1], \min, \max, 0, 1)$, sometimes called *t-norms.* Let $r, s, t \in [0,1]$.

1. Suppose $s \star t = s \cdot t$ is ordinary multiplication. Then, $[s,t]$ is calculated

$$[s,t] = \begin{cases} t/s & s > t \\ 1 & s \leqslant t \end{cases}$$

from

$$r \cdot s \leqslant t \quad \text{if and only if} \qquad\qquad r \leqslant [s,t].$$

2. Define $s \star t = \min(s+t-1, 0)$ and check

$$[s,t] = \begin{cases} 1-s+t & s > t \\ 1 & s \leqslant t \end{cases}.$$

3. Define $s \star t = \min(s,t)$ and check

$$[s,t] = \begin{cases} t & s > t \\ 1 & s \leqslant t \end{cases}.$$

Given a residuated lattice structure on $[0,1]$ and a set X, a *fuzzy set* is a map $A : X \to [0,1]$. Viewing fuzzy sets as propositions, the logical operations of conjunction ($\wedge$), disjunction ($\vee$) and implication ($\to$) are defined pointwise according to the residuated lattice structure. For instance, if A and B are propositions, $[A,B]$ is a fuzzy set with $([A,B])(s) = A(s) \to B(s)$.

Example 2.10 (Heyting Algebra)**.** Every bounded lattice is a monoid under the meet operation. However, this monoid structure does not *ipso facto* satisfy (2.3) which now reads

$$x \wedge y \preccurlyeq z \quad \text{if and only if} \qquad x \preccurlyeq [y, z]$$

for all $x, y, z \in \mathbf{H}$, a *Heyting algebra.* It follows that

$$[y, z] = \qquad \bigvee \{x \in \mathbf{H} \mid x \wedge y \preccurlyeq z\}$$

leading to the definition of a *pseudocomplement* $\neg x = [x, 0]$. Examples of Heyting algebras are plentiful and run deep in foundation of mathematics [107].

Example 2.11 (Max-Plus Lattice)**.** Max-plus algebra has found plentiful applications in locomotion [106], control [109], scheduling [31], and machine learning [111]. All of max-plus algebra is based on the idea of "doing linear algebra" over a residuated lattice, instead of a field. Consider the extended reals $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty, \infty\}$ which forms a complete lattice under $\leqslant$ with the additional requirement that $-\infty \leqslant t \leqslant \infty$ for all $t \in \mathbb{R}$. Then, $(\mathbb{R}_{\max}, \wedge, \vee, -\infty, \infty, +, 0, [-, -])$ has the structure of a residuated lattice with $[s, t] = \max(t - s, 0)$. A *max-plus* or *tropical* vector space, is then the product lattice $\mathbb{R}^d_{\max}$. In some setting, this product structure is called a *weighted lattice* [110] because you can define "scalar multiplication" component-wise.

2.4. Additional Examples

Now that most of the background material on lattices is established, we dive into some more examples.

Example 2.12 (Subsets)**.** Powersets are fundamental examples of lattices to the degree that the meet and join symbols $\wedge/\vee$ are derivates of the symbols $\cap/\cup$ which normally represent intersection and union. Given a set S, the *powerset* $\wp(S)$ is a lattice under intersection and union. By definition, a union of set of subsets sets is the smallest set containing every subset. Similarly, an intersection of a set of subsets is the largest set contained in each subset. If we let $\mathbf{B} = \{0M1\}$, then powersets are equivalent to the lattice of maps $S \to \mathbf{B}$, with the inherited lattice structure

$$A \wedge B(x) = \qquad \texttt{AND}\left(A(x), B(x)\right),$$
$$A \vee B(x) = \qquad \texttt{OR}\left(A(x), B(x)\right),$$

$A, B \in \mathbf{B}^S$. Similarly, a *multiset* is a map $S \to \mathbb{N}$, often denoted $\mathbb{N}\left[S\right]$. The lattice structure on multisets is the following

$$A \wedge B(x) = \qquad \min\left(A(x), B(x), \right)$$
$$A \vee B(x) = \qquad \max\left(A(x), B(x), \right)$$

$A, B \in \mathbb{N}\left[S\right]$.

An interesting use-case of powerset lattices is modeling preferences. If X is a set of alternatives, for instance, restaurants), then $\operatorname{Rel} X := \wp(X \times X)$ is the set of all possible preference relations on X. Preference relations can be combined with the meet and join axioms, in this case, intersection and union. More interesting lattice structures are obtained by restricting $\operatorname{Rel} X$ to transitive-reflexive relations, as it is often argued that preferences are transitive.

Tangential to the theory of preference relations, is the theory of choice functions. A *choice function* is a map $c : \wp(X) \to \wp(X)$ such that $c(A) \subseteq A$ for all $A \in \wp(X)$. Intuitively, a choice function narrows down a subset of choices. It is established that the set of choice functions forms a lattice with pointwise union and intersection; interesting subclasses of choice functions form lattices as well [114].

Example 2.13 (Subspaces)**.** Let V be a $\Bbbk$-vector space. Let $\operatorname{Sub} V$ consist of the set of all subspaces of V. If we write, $\operatorname{Grass}(V, d)$, for the *Grassmanian*, the d-dimensional subspaces, then we can write

$$\operatorname{Sub} V = \qquad \coprod_{d \geqslant 0} \operatorname{Gr}(V, d).$$

$\mathrm{Sub}\, V$ is a lattice under the subspace ordering $K \leqslant W$ if K is a subspace of W. The lattice operations are the following: $W \vee W' = \{w + w' \mid w \in W,\ w' \in W'\}$ often denoted $W + W'$; $W \wedge W' = W \cap W'$ which is trivially a subspace of V. Principal Component Analysis (PCA) and its variants can be thought of as a projection of datapoints onto an optimal subspace W of some feature space V [166].

Suppose $T : V \to V$ is a linear transformation. We say a subspace $W \subseteq V$ is *invariant* if $T(W) \subseteq W$. For instance, 1-dimensional invariant subspaces are spanned by vectors $\mathbf{x}$ satisfying $T(\mathbf{x}) = \lambda \mathbf{x}$ for some scalar λ—these invariant subpaces are eigenspaces. It follows that the invariant subspaces of T denoted $\mathrm{Inv}\, T$ form a lattice, a sublattice of $\mathrm{Sub}\, V$. Conditions are known for invariant subspaces to be isomorphic [25].

Example 2.14 (Partitions)**.** The set of partitions $\Pi\,(S)$ of a set S forms a lattice. Suppose π and π' are partitions. Then, the relation $\pi \preccurlyeq \pi'$ means that π *is finer than* π*'* or π *is a refinement of* π'. Put another way, for every $A \in \pi$, there is an $A' \in \pi'$ with $A \subseteq A'$. The join $\pi \vee \pi'$ is the finest common coarsening of π and π' while $\pi \wedge \pi'$ is the coarsest common refinement. Suppose $\pi = \{12|34|5\}$ and $\pi' = \{123|45\}$, then $\pi \wedge \pi' = \{12|3|4|5\}$ and $\pi \vee \pi' = \{12345\}$. Partitions are ubiquitous in engineering, from teams of autonomous agents [88] to hierarchial clustering algorithms [30]. A *subpartition* is a partition of some subset S' of the ground set S. By similar reasoning, subpartitions form a lattice denoted $\Pi_*\,(S)$. Maps from the time domain to the lattice of subpartitions have been used to model dynamic graph with changing node sets [92].

Example 2.15 (Information [151])**.** The following example is due to Recall, a random variable is a measurable map $X : \Omega \to A$ to some set A which, for practical reasons, we call the *alphabet*. Depending on the context, we can think of A as a set of symbols or as a set of words. The *entropy*

$$H(X) = -\sum_{w \in A} \mathbf{Pr}\left[X^{-1}\{w\}\right] \log\left(\mathbf{Pr}\left[X^{-1}\{w\}\right]\right).$$

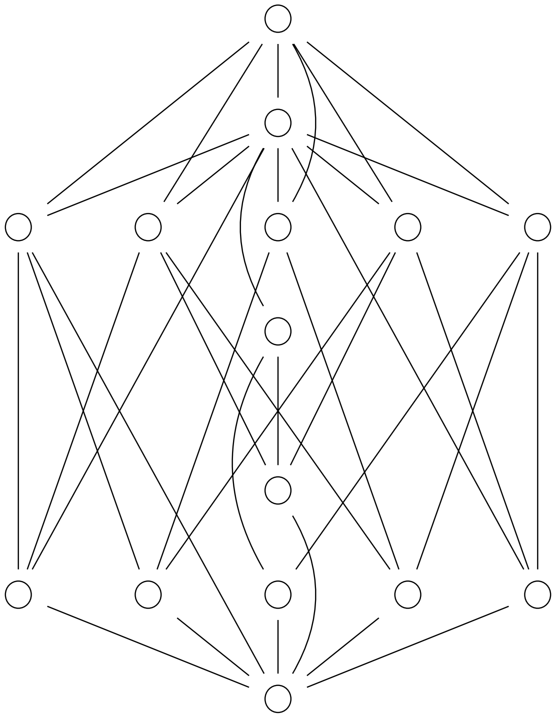


Figure 2.2: Lattice of partitions $\Pi(X)$ on $X = \{1, 2, 3, 4\}$

Entropy rougly represents the amount of information (bits) required to encode the outcome of X. A variation of entropy, relative entropy, denoted $H(Y|X)$, is the amount of information to describe the outcome of another random variable Y given that the value X generated is already known. Omitting some technical details [102], random variables over a fixed probability distribution form a poset with $X \succcurlyeq Y$ if and only if $H(Y|X) = 0$.

It is know the set of A-valued random variables under the information order forms a lattice isomoprhic to $\Pi(A)$ [102]. The *information lattice* is a sublattice $\mathbf{I}$ of $\Pi(S)$ subordinate to the σ-algebra defining the underlying probability distribution. The join of random variables $X \vee Y$ is the *joint information content* between then, exactly the joint random variable (X, Y). The meet $X \wedge Y$, on the other hand, is the *common information* between the two variables.

2.4.1. Decomposition

A recurring theme in algebra is to decompose structures into their smallest possible components. This leads to the following definition.

Definition 2.9. Let $\mathbf{L}$ be a lattice. An element $0 \neq z \in \mathbf{L}$ is join-irreducible if $z = x \vee y$ implies $x = z$ or $y = z$. Similarly, an element $1 \neq z \in \mathbf{L}$ is meet-irreducible if $z = x \wedge y$ implies $z = x$ or $z = y$. We denote the set of join-irreducible elements $\mathcal{J}(\mathbf{L})$.

Clearly, $(\mathcal{J}(\mathbf{L}), \preccurlyeq)$ is a poset with $x \preccurlyeq y$ if and only if $x \wedge y = x$. Less clearly, downsets form distributive lattices.

Theorem 2.3 (Birkhoff Duality)**.** Suppose $\mathbf{L}$ is a finite distributive lattice and P is a finite poset. Then,

$$\mathcal{D}\left(\mathcal{J}(\mathbf{L})\right) \cong \mathbf{L}$$

and

$$\mathcal{J}\left(\mathcal{D}(\mathrm{P})\right) \cong \mathrm{P}.$$

Proof. See [140, Theorem 10.1]. □

Theorem 2.3 allows us to move back and forth between these, albeit restrictive, classes of posets and lattices with ease. Because $\mathcal{D}(\mathrm{Q})$ is a lattice of subsets (i.e. sublattice of the powerset $\wp(\mathrm{Q})$), Theorem 2.3 can also be interpreted as *every finite distributive lattice can be represented as a lattice of subsets.* In general, not restricted to finite posets and distributed lattices, the story is more complicated as illustrated by the following example.

Example 2.16 (Natural Numbers)**.** Consider the natural numbers $\mathbb{N}$ with the order $m \preccurlyeq n$ if and only if $n|m$, i.e. there is an $a \in \mathbb{N}$ with $n = a \cdot m$. $(\mathbb{N}, \preccurlyeq)$ is a lattice with $m \wedge n = \mathrm{lcm}(m, n)$ and $m \vee n = \gcd(m, n)$. First, notice that $\mathbb{N}$ is distributive by the classic distributive properties of gcd and lcm. Hence, $(\mathbb{N}, \preccurlyeq)$ is an (infinite) distributive lattice. The join-irreducibles are the prime number $\mathcal{J}(\mathbb{N})$ with the partial order inherited by $(\mathbb{N}, \preccurlyeq)$. It follows that $\mathcal{D}(\mathcal{J}(\mathbb{N})) = \wp(\mathfrak{p})$. However, by the Fundamental Theorem of Arithmetic, $(\mathbb{N}, \preccurlyeq) \cong \mathbb{N}\left[\mathfrak{p}\right]$—every natural number has a unique prime factorization. Hence, $\mathcal{D}\left(\mathcal{J}(\mathbf{L})\right) \ncong \mathbf{L}$. Because $\mathbb{N}$ is not finite, we could not apply Birkhoff Duality.

CHAPTER 3

Galois Connections

Suppose P and Q are posets. A Galois connection is a pair of compatible order-preserving maps, one from P to Q, another Q to P, such that arbitrary joins and arbitrary meets are preserved in P and Q respectively. On one hand, Galois connections are share a resemblance to linear adjoints (Definition 3.1), but on the other hand, being order approximations of an inverse (Proposition 3.2), share a resemblance to Moore-Penrose pseudoinverses.

3.1. Defining Properties

Definition 3.1. Let $(\mathrm{P}, \preccurlyeq)$ and $(\mathrm{Q}, \preccurlyeq)$ be posets. A *Galois connection* is a pair $(f_\bullet, f^\bullet)$ of order-preserving maps $f_\bullet : \mathrm{P} \to \mathrm{Q}$ and $f^\bullet : \mathrm{Q} \to \mathrm{P}$ such that for all $x \in \mathrm{P}$, $y \in \mathrm{Q}$,

$$f_\bullet(x) \preccurlyeq y \Leftrightarrow \qquad x \preccurlyeq f^\bullet(y). \tag{3.1}$$

We call $f_\bullet$ a *lower-adjoint* and $f^\bullet$ an *upper-adjoint*.

Remark 3.1. The definition of a Galois connection is reminiscent of the definition of the adjoint of a linear transformation. If $T : V \to W$ is a map of inner-product spaces $(V, \langle\ ,\ \rangle_V)$, $(W, \langle\ ,\ \rangle_W)$, then $T^* : W \to V$ is the map defined by the following property

$$\langle T(\mathbf{x}), \mathbf{y}\rangle_W = \qquad \langle \mathbf{x}, T^*(\mathbf{y})\rangle_V. \tag{3.2}$$

Given fixed complete lattices $\mathbf{K}$ and $\mathbf{L}$, we denote the set of Galois connections between these lattices $\mathcal{L}tc(\mathbf{K}, \mathbf{L})$ (notation adapted from [69]). The set $\mathcal{L}tc(\mathbf{K}, \mathbf{L})$ is a poset with the order $f \leqslant g$ if and only if $f_\bullet(x) \preccurlyeq g_\bullet(x)$ and $g^\bullet(y) \preccurlyeq f^\bullet(y)$ for all $x, y \in \mathbf{L}$ endowing it with the structure of a complete lattice. Galois connections have an equivalent definition highlighting the property that they are the “best approximation” of an order inverse.

Proposition 3.1. Suppose P and Q are posets. The following are equivalent.

1. $(f_\bullet, f^\bullet) : \mathrm{P} \to \mathrm{Q}$ is a Galois connection.

2. $f_\bullet : \mathrm{P} \to \mathrm{Q}$ and $f^\bullet : \mathrm{Q} \to \mathrm{P}$ are order-preserving, and $f^\bullet f_\bullet \succcurlyeq \mathrm{id}_\mathrm{P}$ and $f_\bullet f^\bullet \preccurlyeq \mathrm{id}_\mathrm{Q}$.

Proof. Suppose $f_\bullet(x) \preccurlyeq y$ if and only if $x \preccurlyeq f^\bullet(y)$ for all $x \in \mathrm{P},\ y \in \mathrm{Q}$. Substitute $f_\bullet(x)$ for y. Then, $x \preccurlyeq f^\bullet f_\bullet(x)$. Alternatively, if we substitute $f^\bullet(y)$ for x, then we obtain $f_\bullet f^\bullet(y) \preccurlyeq y$. Conversely, suppose $f^\bullet \circ f_\bullet \succcurlyeq \mathrm{id}_\mathrm{P}$. If $f_\bullet(x) \preccurlyeq y$, then, because $f^\bullet$ is order-preserving, we have $f^\bullet(f_\bullet(x)) \preccurlyeq f^\bullet(y)$. Because $f^\bullet(f_\bullet(x)) \succcurlyeq x$, it follows by transitivity $f^\bullet(y) \succcurlyeq x$. Similarly, if $x \preccurlyeq f^\bullet(y)$, then $f_\bullet(y^\bullet) \preccurlyeq y$ implies $f_\bullet(x) \preccurlyeq y$. □

Composing lower and upper adjoints stabilizes.

Proposition 3.2. Suppose $(f_\bullet, f^\bullet)$ is a Galois connection as above. Then,

$$\begin{aligned} f_\bullet f^\bullet f_\bullet = &\quad f_\bullet; \\ f^\bullet f_\bullet f^\bullet = &\quad f^\bullet. \end{aligned}$$

Proof. For the first, it suffices to show that $f_\bullet f^\bullet f_\bullet(x) \succcurlyeq f_\bullet(x)$ for all $x \in \mathrm{P}$ because $f_\bullet f^\bullet f_\bullet \preccurlyeq f_\bullet$ follows from Proposition 3.1. Also, by Propositon 3.1, $f^\bullet f_\bullet(x) \succcurlyeq x$. $f_\bullet$ is order-preserving implies $f_\bullet f^\bullet f_\bullet(x) \succcurlyeq f_\bullet(x)$. The second is dual to the first. □

One consequence of the Proposition 3.2 is that $f^\bullet f_\bullet$ and $f_\bullet f^\bullet$ are idempotent.

Corollary 3.1. Suppose $(f_\bullet, f^\bullet)$ is a Galois connection as above. Then,

$$\begin{aligned} f^\bullet f_\bullet f^\bullet f_\bullet = &\quad f^\bullet f_\bullet \\ f_\bullet f^\bullet f_\bullet f^\bullet = &\quad f_\bullet f^\bullet \end{aligned}$$

Another consequence of Proposition 3.2 characterizes Galois connections in which $f_\bullet$ is an order-embedding.

Proposition 3.3. Suppose $(f_\bullet, f^\bullet) : \mathrm{P} \to \mathrm{Q}$ is a Galois connection and $f_\bullet$ is an order-embedding. Then, $f^\bullet f_\bullet = \mathrm{id}_{\mathrm{P}}$.

We need a lemma.

Lemma 3.1. Suppose $(f_\bullet, f^\bullet)$ is a Galois connection. Then, $f_\bullet$ is an order-embedding if and only if $f^\bullet f_\bullet$ is an order-embedding.

Proof of Lemma 3.1. Suppose $f_\bullet$ is an order-embedding. Then, by Proposition 3.2, $f_\bullet f^\bullet f_\bullet(x) \preccurlyeq f_\bullet f^\bullet f_\bullet(y)$ if and only if $x \preccurlyeq y$. This implies $x \preccurlyeq y$ if and only if

$$\begin{aligned} f^\bullet f_\bullet(x) \preccurlyeq & \quad f^\bullet f_\bullet f^\bullet f_\bullet(y) \\ \preccurlyeq & \quad f^\bullet f_\bullet(y) \end{aligned}$$

by Corollary 3.1. □

Proof of Proposition 3.3. By Proposition 3.1, it suffices to show $f^\bullet f_\bullet(x) \preccurlyeq x$ for all $x \in \mathrm{P}$. By Lemma 3.1, $f^\bullet f_\bullet(x) \preccurlyeq x$ if and only if

$$f^\bullet f_\bullet \left(f^\bullet f_\bullet(x)\right) \preccurlyeq \quad f^\bullet f_\bullet(x).$$

By Corollary 3.1, $f^\bullet f_\bullet(x) \preccurlyeq x$ holds for all $x \in \mathrm{X}$. □

A Galois connection gives rise to a closure and coclosure operator.

Corollary 3.2. $f^\bullet f_\bullet : \mathrm{P} \to \mathrm{P}$ is a closure operator. $f_\bullet f^\bullet : \mathrm{Q} \to \mathrm{Q}$ is a co-closure operator.

Proof. Proposition 3.1 and Corollary 3.1 □

We denote $\mathrm{Fix}(f^\bullet f_\bullet)$ by $\mathbf{Cl}(\mathbf{K})$ and $\mathrm{Fix}(f_\bullet f^\bullet)$ by $\mathbf{Cl}(\mathbf{L})$.

Definition 3.2. Given a connection $(f_\bullet, f^\bullet) : \mathbf{K} \to \mathbf{L}$, define a poset

$$\mathrm{Gal}(f) = \{(x, y) \mid f_\bullet(x) = y,\ x = f^\bullet(y)\} \subseteq \mathbf{K} \times \mathbf{L}$$

called the *Galois lattice* of f.

$\mathrm{Gal}(f)$ is in fact a complete lattice.

Proposition 3.4. Suppose $f : \mathbf{K} \to \mathbf{L}$ is a Galois connection between complete lattices $\mathbf{K}$ and $\mathbf{L}$. Then, the following are isomorphic complete lattices: $\mathbf{Cl}(\mathbf{K})$, $\mathbf{Cl}(\mathbf{L})$, $\mathrm{Gal}(f)$.

Proof. For the lattice structure, Theorem 2.1. The isomorphism follows from Proposition 3.2. □

If $f = (f_\bullet, f^\bullet)$ is a Galois connection, $f_\bullet$ is join-preserving and $j^\bullet$ is meet-preserving. Moreover, if $f_\bullet$ is a join-preserving map, $f^\bullet$ is uniquely defined. Conversely, if $f^\bullet$ is a join-preserving map, then $f_\bullet$ is uniquely defined.

Theorem 3.1 (Adjoint Functor Theorem[4])**.** Suppose P and Q are posets and $(f_\bullet, f^\bullet) : \mathrm{P} \to \mathrm{Q}$ is a Galois connection. Then, $f_\bullet$ preserves joins and $f^\bullet$ preserves preserves meets (whenever they exist). Conversely, if $f : \mathrm{P} \to \mathrm{Q}$ is a join-preserving, there there exists a map $f^* : \mathrm{Q} \to \mathrm{P}$ given by

$$f^*(y) = \bigvee \{x \mid f(x) \preccurlyeq y\} \tag{3.3}$$

such that (f, f^*) is a Galois connection.

It is sometimes convenient to rewrite the formula for f^* as

$$f^*(y) = \bigvee f^{-1}(\downarrow y)$$

[4] This result is a special case of the Adjoint Functor Theorem due to Freyd [57].

Proof of Theorem 3.1. In the first part, suppose $f_\bullet$ is lower adjoint. We show $f_\bullet$ preserves joins. Suppose $\{x_i\}_i \subseteq \mathrm{P}$ is an arbitrary subset of P. Then,

$$f_\bullet\left(\bigvee_i x_i\right) \preccurlyeq y \quad \Leftrightarrow \qquad\qquad \bigvee_i x_i \preccurlyeq f^\bullet(y)$$

Set

$$y = \bigvee_i f(x_i).$$

Then,

$$f_\bullet\left(\bigvee_i x_i\right) \preccurlyeq \bigvee_i f(x_i) \quad \Leftrightarrow \qquad\qquad \bigvee_i x_i \preccurlyeq f^\bullet(\bigvee_i f(x_i))$$

By Proposition 2.1, $\bigvee_i x_i \preccurlyeq f^\bullet(\bigvee_i f(x_i))$ holds tautologically. Hence, $f_\bullet(\bigvee_i x_i) \preccurlyeq \bigvee_i f(x_i)$ holds. By Proposition 2.1 again,

$$f_\bullet(\bigvee_i f(x_i)) \succcurlyeq \bigvee_i f(x_i)$$

, leading to the equality desired.

In the second part, we show (3.3) is the upper adjoint of f. By Proposition 3.1, it suffices to show $f^*f(x) \succcurlyeq x$ and $ff^*(y) \preccurlyeq y$. We have

$$f^*f(x) = \bigvee f^{-1}\left(\downarrow f(x)\right).$$

But $\{x\} \subseteq f^{-1}f(x) \subseteq f^{-1}\left(\downarrow f(x)\right)$ implies

$$f^*f(x) = \bigvee f^{-1}\left(\downarrow f(x)\right) \succcurlyeq \bigvee f^{-1}f(x) \succcurlyeq \bigvee\{x\} = x.$$

For the other inequality, we have

$$ff^*(y) = f\left(\bigvee f^{-1}(\downarrow y)\right) = \bigvee ff^{-1}(\downarrow y)$$

since, by the first part, f preserves joins. Notice, $ff^{-1}(\downarrow y) \subseteq \downarrow y$. Hence,

$$ff^*(y) = \bigvee ff^{-1}(\downarrow y) \preccurlyeq \bigvee \downarrow y = y.$$

□

The dual result is the following.

Corollary 3.3. If $f : \mathrm{Q} \to \mathrm{P}$ is meet-preserving, there exists a join-preserving map $f_* : \mathrm{P} \to \mathrm{Q}$ such that (f_*, f) is a Galois connection, defined

$$f_*(x) = \qquad \bigwedge\{y \mid f(y) \succcurlyeq x\}.$$

3.2. Galois Connections from Relations

Some authors define Galois connections to be adjoint order-reversing maps, while others—as do we—call adjoint pairs of order-preserving maps Galois connections. We define a *contravariant Galois connection* as a pair $(f^{\uparrow}, g^{\downarrow})$ of order-reversing maps

$$f^{\uparrow} : \mathbf{K} \longleftrightarrow \mathbf{L} : f^{\downarrow}$$

such that

$$f^{\downarrow}f^{\uparrow}(x) \succcurlyeq x : \qquad f^{\uparrow}f^{\downarrow}(y) \succcurlyeq y.$$

or, equivalently,

$$f^{\uparrow}(x) \succcurlyeq y \Leftrightarrow \qquad f^{\downarrow}(y) \succcurlyeq x$$

Whenever there is any ambiguity, we call an order-preserving Galois connection *covariant*. There is an isomorphism

$$\mathcal{L}tc\,(\mathbf{K}^{\mathrm{op}}, \mathbf{L}) \cong \mathcal{L}tc\,(\mathbf{K}, \mathbf{L}^{\mathrm{op}})^{\mathrm{op}},$$

and we may view $(f^{\uparrow}, f^{\downarrow})$ as an ordinary (order-preserving) Galois connection in two ways.

Caveat lector. Although there is a one-to-one correspondence between the *sets* of join-reversing maps and meet-reversing maps, the correspondence is not an isomorphism of lattices.

There are canonical ways to construct Galois connections of each type (covariant, contravariant) from a binary relation $\mathcal{R} \subset X \times Y$. The covariant *Galois correspondence* is originally due to Schmidt [146], while the contravariant Galois correspondence is due to Ore [125].

In Chapter 9, we encounter a Galois connection constructed in the above manner.

Theorem 3.2. Suppose $\mathcal{R} \in \mathrm{Rel}(X, Y)$ is a binary relation. Then, $\mathcal{R}$ induces a covariant Galois connection

$$\begin{tikzcd}
\wp(X) \arrow[r, bend left, "\mathcal{R}_\exists"] \arrow[r, phantom, "\perp"] & \wp(Y) \arrow[l, bend left, "\mathcal{R}_\forall"]
\end{tikzcd}$$

with

$$\mathcal{R}_\exists(U) = \{y \in Y \mid (\exists x) \quad x \in U \ \wedge \ x \ \mathcal{R} \ y\}, \tag{3.4}$$

$$\mathcal{R}_\forall(V) = \{x \in X \mid (\forall y) \quad x\mathcal{R}y \Rightarrow y \in V\}, \tag{3.5}$$

$U \subseteq X, V \subseteq Y$.

Proof. First, we show that $\mathcal{R}_\exists$ is join-preserving so we can apply Proposition 3.1 in order to compute $\mathcal{R}_\forall$. Let $\{U_i\}_{i \in I}$ be a family of subsets of X. Then,

$$\begin{aligned}
\mathcal{R}_\exists\left(\bigcup_{i \in I} U_i\right) &= \\
&= \{y \in Y \mid (\exists x) x \in \bigcup_{i \in I} U_i \;\wedge\; x\mathcal{R}y\} \\
&= \{y \in Y \mid (\exists x \; \exists i \in I) \quad x \in U_i \;\wedge\; x\mathcal{R}y\} \\
&= \bigcup_{i \in I} \{y \in Y \mid (\exists x) \quad x \in U_i \;\wedge\; x\mathcal{R}y\} \\
&= \bigcup_{i \in I} \mathcal{R}_\exists(U_i).
\end{aligned}$$

Now, we can apply the second part of Proposition 3.1 to compute $\mathcal{R}_\forall$. Let $V \subseteq Y$. Then,

$$\begin{aligned}
\mathcal{R}_\forall(V) &= \bigcup \{U \subseteq X \mid \mathcal{R}_\exists(U) \subseteq V\} \\
&= \bigcup \{U \subseteq X \mid \{y \in Y \mid (\exists x) \quad x \in U \;\wedge\; x \;\mathcal{R}\; y\} \subseteq V\} \\
&= \{x \in X \mid (\forall y) \quad x \;\mathcal{R}\; y \Rightarrow y \in V\}
\end{aligned}$$

as desired. □

The following construction has lead to the field of formal concept analysis (FCA) [165].

Theorem 3.3. Suppose $\mathcal{R} \in \mathrm{Rel}(X, Y)$ is a binary relation. Then, $\mathcal{R}$ induces a contravariant Galois connection

$$\begin{tikzcd}
\wp(X) \arrow[r, bend left, "\mathcal{R}^\uparrow"] \arrow[r, phantom, "\perp"] & \wp(Y)^{\mathrm{op}} \arrow[l, bend left, "\mathcal{R}^\downarrow"]
\end{tikzcd}$$

with

$$\mathcal{R}^\uparrow(U) = \{y \in Y \mid (\forall x)\,(x \in U \Rightarrow x \;\mathcal{R}\; y)\} \tag{3.6}$$

$$\mathcal{R}^\downarrow(V) = \{x \in X \mid (\forall y)\,(y \in V \Rightarrow x \;\mathcal{R}\; y)\}, \tag{3.7}$$

$U \subseteq X, V \subseteq Y$.

Proof. We show $\mathcal{R}^{\uparrow}(U) \supseteq V$ if and only if $\mathcal{R}^{\downarrow}(V) \supseteq U$ for all $U \subseteq X, V \subseteq Y$ since in $\wp(X)^{\mathrm{op}}$, $V \preccurlyeq V'$ if and only if $V \supseteq U'$. Observe,

$$\mathcal{R}^{\ \downarrow} = \mathcal{R}^{\dagger\uparrow}.$$

Hence, by simply replacing the role of X and Y, we are done. □

3.3. Integral Transforms

Recall the residual operation $[-.-] : \mathbb{L} \times \mathbb{L} \to \mathbb{L}$ of a residuated lattice has the defining property

$$x \star y \preccurlyeq z \quad \Leftrightarrow \qquad\qquad\qquad x \preccurlyeq [y, z].$$

Another way of stating this property is there is a Galois connection

$$(- \star y, [y, -]) : \mathbb{L} \to \mathbb{L}.$$

We have the following lemma as consequence.

Lemma 3.2. Suppose $(\mathbb{L}, \wedge, \vee, 0, 1, \star, e, [-,-])$ is a complete residuated lattice. Then,

1. $(\bigvee_i x_i) \star y = \bigvee_i (x_i \star y)$,

2. $[x, \bigwedge_i y_i] = \bigwedge_i [x, y_i]$.

Proof. Apply Theorem 3.1. □

Suppose $\mathbb{L}$ is a residuated lattice, and suppose X is a set, possibly with additional structure.

A *kernel* is a map

$$H : X \times X \to \mathbb{L}$$

Integration over kernels is readily defined as follows.[5] Suppose f is a map $f : X \to \mathbb{L}$ and $H : X \times X \to \mathbb{L}$ is a kernel. Then, the *integral transform* of f is the map $\hat{f} : X \to \mathbb{L}$ defined

$$\hat{f}(x) = \bigvee_{y \in X} H(x, y) \star f(y). \tag{3.8}$$

Theorem 3.4. Suppose $\mathbb{L}$ is a residuated lattice and $H : X \times X \to \mathbb{L}$ is a kernel. Then, the following forms a Galois connection

$$\begin{tikzcd}
\mathbb{L}^X \arrow[r, bend left, "(\hat{-})"] \arrow[r, phantom, "\perp"] & \mathbb{L}^X \arrow[l, bend left, "(\check{-})"]
\end{tikzcd}$$

$$\begin{aligned}
(f : X \to \mathbb{L}) \mapsto &\qquad \hat{f}(x) = \bigvee_{y \in X} H(x, y) \star f(y) \\
\check{g}(y) = \bigwedge_{x \in X} [H(y, x), g(y)] \leftarrow\!\shortmid &\qquad (g : X \to \mathbb{L})
\end{aligned}$$

Proof. Our strategy is to show $(\hat{-})$ is join-preserving and apply Theorem 3.1 to calculate $(\check{-})$. Suppose $f = \bigvee_{i \in I} f_i$ where $f_i \in \mathbb{L}^X$ are $\mathbb{L}$-valued functions on $\mathbb{L}$ with the order $f_i \preccurlyeq g_i$

[5] We provide a definition slightly more general than the one already introduced by Maragos [109], but less general than an analogous categorical one [71].

if and only if $f_i(x) \preccurlyeq g_i(x)$ for all $x \in X$. Then,

$$\begin{aligned} \hat{f}(x) = &\quad \bigvee_{y \in X} H(x,y) \star \left(\bigvee f_i(y)\right) \\ = &\quad \bigvee_{y \in X} \bigvee_{i \in I} H(x,y) \star f_i(y) \\ = &\quad \bigvee_{i \in I} \hat{f}_i(x) \end{aligned}$$

by Lemma 3.2.

To calculate the right adjoint, by Theorem 3.1,

$$\check{g} = \quad \bigvee \{f \mid \hat{f} \preccurlyeq g\},$$

which implies f is the greatest function whose integral transform is less than g

$$\check{g}(y) = \quad \bigvee_{f \in \mathbb{L}^X} \{f(y) \mid \hat{f}(y) \preccurlyeq g(y) \quad \forall x\}.$$

Hence, for all $y \in X$, f satisfies

$$\bigvee_{x \in X} H(y,x) * f(x) \preccurlyeq \quad g(y).$$

Applying Lemma 3.2,

$$H(y,x) * f(x) \preccurlyeq g(y) \quad \Leftrightarrow \quad f(x) \preccurlyeq [H(y,x), g(y)].$$

Therefore, tightening the bound,

$$\check{g}(y) = \quad \bigwedge_{x \in X} [H(y,x), g(y)].$$

□

Example 3.1 (Max-Plus Connection)**.** Suppose $\mathbb{L}$ is the residuated lattice $(\mathbb{R}_{\max}, \min, \max, -\infty, \infty, 0, +)$ with $[x, y] = y - x$ if $x, y \in \mathbb{R}_{\max} \setminus \{-\infty, \infty\}$. Then, Theorem 3.4 implies

$$(A \varovee \mathbf{x})_i = \qquad \bigvee_{i=1}^{n} a_{ij} + x_j$$

has an upper adjoint defined by $A^\dagger \varowedge \mathbf{x}$ where $A^\dagger$ is the matrix with $a^\dagger_{ij} = -a_{ji}$. Hence,

$$(A^\dagger \varowedge \mathbf{x})_j = \qquad \bigwedge_{i=1}^{n} a_{ji} - x_j.$$

Consequently, by Proposition 3.1,

$$\begin{aligned} A^\dagger \varowedge (A \varovee \mathbf{x}) \geqslant & \qquad \mathbf{x}, \\ A \varovee \left(A^\dagger \varowedge \mathbf{x}\right) \leqslant & \qquad \mathbf{x}, \end{aligned}$$

and

$$A \varovee \mathbf{x} \leqslant \mathbf{y} \quad \text{if and only if} \qquad \mathbf{x} \leqslant A^\dagger \varowedge \mathbf{y}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n_{\max}$.

3.4. Category Theory

We do no assume prior knowledge of category theory. However, as is discussed elsewhere [56], there are irrefutable parallels between order theory and category theory. Reoccurring constructions pertaining to sheaves and cosheaves are further illuminated if we view them as categorical limits and colimits [133] which, here, we present in a more friendly form- as equalizers and coequalizers.

A category $\mathcal{C}$ consists of a collection of *objects* $\operatorname{Ob}\mathcal{C}$ and a collection of *morphisms* also called *arrows* $\operatorname{Mor}\mathcal{C}$ such that for any two objects $x, y \in \operatorname{Ob}\mathcal{C}$, there is a collection of morphisms

$\mathrm{Hom}_{\mathcal{C}}(x, y) \subseteq \mathrm{Mor}\,\mathcal{C}$ between them with a map

$$\circ_{x,y,z} : \mathrm{Hom}_{\mathcal{C}}(x, y) \times \mathrm{Hom}_{\mathcal{C}}(y, z) \to \mathrm{Hom}_{\mathcal{C}}(x, z)$$

satisfying[6]

- $(h \circ g) \circ f = h \circ (g \circ f)$ (Associativity).
- For every $x \in \mathrm{Ob}\, C$, there is a $\mathrm{id}_x \in \mathrm{Hom}_{\mathcal{C}}(x, x)$ with $f \circ \mathrm{id} = f$ and $\mathrm{id} \circ f = f$ for every composable $f \in \mathrm{Mor}\,\mathcal{C}$ (Identity).

Colloquially, *if you tell me the objects and arrows, I can tell you the category.*

Examples 3.1.

- Suppose $(\mathrm{P}, \preccurlyeq)$ is a preorder. Then, we may define a category $\mathcal{P}$ with $\mathrm{Ob}\,\mathcal{P} = \mathrm{P}$ and $\mathrm{Hom}_{\mathcal{P}}(x, y) = *$ (a one-point set) whenever $x \preccurlyeq y$ in P and $\mathcal{P}(x, y) = \emptyset$ otherwise.
- Suppose $(M, \star, e)$ is a *monoid.* Then, M is a category $\mathcal{M}$ with $\mathrm{Ob}\,\mathcal{M} = *$, a single object, and $\mathrm{Hom}_{\mathcal{M}}(*, *) = M$. Composition of arrows is defined according to the binary operation $\star$.

3.4.1. Functors

Categories are useful often because we can compare them, a perspective that has lead to connections between different branches of mathematics. Comparing two categories amounts to demonstrating a mapping between them called a functor. A *functor* F between categories $\mathcal{C}$ and $\mathcal{D}$ consists of a maping $F : \mathrm{Ob}\,\mathcal{C} \to \mathrm{Ob}\,\mathcal{D}$ and a map $F_{x,y} : \mathrm{Hom}_{\mathcal{C}}(x, y) \to \mathrm{Hom}_{\mathcal{D}}(Fx, Fy)$ such that for every $f \in \mathrm{Hom}_{\mathcal{C}}(x, y)$ and $g \in \mathrm{Hom}_{\mathcal{C}}(y, z)$,

$$\begin{aligned} F_{x,z}(g \circ f) = && F_{y,z}(g) \circ F_{x,y}(f), \\ F(\mathrm{id}_x) = && \mathrm{id}_{Fx}. \end{aligned}$$

[6] Usually, the subscripts in $\circ_{x,y,z}$ are omitted.

We have already seen functors. For instance, a functor between poset categories $F : \mathcal{P} \to \mathcal{Q}$ is precisely an order-preserving map; a functor between monoid categories $F : \mathcal{M} \to \mathcal{N}$ is precisely a *monoid homomorphism.* Functors also form a category in their own right. If $\mathcal{D}$ and $\mathcal{D}$ are categories, then the $\mathcal{D}^{\mathcal{C}}$ is a category whose objects are functors $F : \mathcal{C} \to \mathcal{D}$ whose morphisms are maps between functors called natural transformations. If $F, G : \mathcal{C} \to \mathcal{D}$, then a *natural transformation* $\eta : F \Rightarrow G$ is, roughly, a set of morphisms $\{\eta_x : Fx \to Gx\}_{x \in \mathrm{Ob}\,\mathcal{C}}$ such that the following diagram commutes for every $f \in \mathrm{Hom}_{\mathcal{C}}(x, y)$

$$\begin{tikzcd}
Fx \arrow[r, "Ff"] \arrow[d, "\eta_x"'] & Fy \arrow[d, "\eta_y"'] \\
Gx \arrow[r, "Gf"'] & Gy
\end{tikzcd}$$

As expected, the the morphisms $\mathrm{Hom}_{\mathcal{D}^{\mathcal{C}}}(F, G)$ are, roughly, the set of natural transformations between F and G.

3.4.2. (Co)equalizers

We make use of exactly one technical construction from category theory called a *(co)equalizer.* Suppose $\mathcal{C}$ is a data category and $f, g : A \to B$ are parallel morphisms from between objects $A, B \in \mathrm{Ob}\,\mathcal{C}$ (e.g. linear transformations, join-preserving maps). By "a diagram commutes," we mean that if you compose a sequence of arrows in the diagram to form a map from a source object to a target object, you obtain the same map as if you had composed any other sequence of arrows to go from the same source object to the same target object.

Definition 3.3 (Equalizer)**.** The *equalizer* of a parallel pair $f, g : A \to B$ is an object E and a map $i : E \to A$ such that for any map $j : D \to A$ there is a unique map $j' : D \to E$ such that the following diagram commutes

$$\begin{tikzcd}
E \arrow[r, "i"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
& D \arrow[u, "j"] \arrow[ul, dashed, "j'"] &
\end{tikzcd} .$$

Definition 3.4 (Coequalizer)**.** The *coequalizer* of a parallel pair $f, g : A \to B$ is an object Q and a map $p : B \to Q$ such that for any map $q : B \to P$ there is a unique map $q' : Q \to P$ such that the following diagram commutes

$$\begin{tikzcd}
& P & \\
A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[u, "q"] \arrow[r, "p"] & Q \arrow[ul, dashed, "q'"']
\end{tikzcd}$$

3.4.3. (Co)products

Everyone is familiar with products of sets (cartesian products) or products of vector spaces (direct sum). These constructions often generalize in order categories.

Definition 3.5 (Product)**.** Suppose $A, B \in \mathrm{Ob}\,\mathcal{C}$. The *product* of A and B is an object denoted $A \prod B$ with maps $p : A \prod B \to A$ and $q : A \prod B \to B$ called *projection maps* such that for an object D and maps $p' : D \to A$ and $q' : D \to B$ there is a unique map $i : D \to A \prod B$ such that the diagram commutes

$$\begin{tikzcd}
A & & B \\
& A \prod B \arrow[ul, "p"'] \arrow[ur, "q"] & \\
& D \arrow[u, dashed, "i"] \arrow[uul, bend left, "p'"] \arrow[uur, bend right, "q'"'] &
\end{tikzcd} .$$

Definition 3.6 (Coproduct)**.** Suppose $A, B \in \mathrm{Ob}\,\mathcal{C}$. The *coproduct* of A and B is an object denoted $A \coprod B$ with maps $i : A \to A \coprod B$ and $j : B \to A \coprod B$ called *injection maps* such that for an object D and maps $i' : A \to D$ and $j' : B \to D$ there is a unique map

$p : A \coprod B \to D$ such that the diagram commutes

$$\begin{tikzcd}
 & D & & \\
 & A \coprod B \arrow[u, dashed, "p"] & & . \\
A \arrow[ur, "i"'] \arrow[uur, bend left, "i'"] & & B \arrow[ul, "j"] \arrow[uul, bend right, "j'"'] &
\end{tikzcd}$$

Example 3.2 (Meets & Joins)**.** Recall that a poset P is a category with an arrow $x \to y$ between two elements (objects) if and only if $x \preccurlyeq y$. The product of two elements (if it exists) is the element $x \wedge y$ defined by the universal property

$$\begin{tikzcd}
x & & y & \\
 & x \wedge y \arrow[ul, "\preccurlyeq"] \arrow[ur, "\preccurlyeq"'] & & . \\
 & z \arrow[u, dashed, "\preccurlyeq"] \arrow[uul, bend left, "\preccurlyeq"] \arrow[uur, bend right, "\preccurlyeq"'] & &
\end{tikzcd}$$

Similarly, the coproduct of x and y is the element $x \vee y$ defined by the universal property

$$\begin{tikzcd}
 & z & & \\
 & x \vee y \arrow[u, dashed, "\preccurlyeq"] & & . \\
x \arrow[ur, "\preccurlyeq"'] \arrow[uur, bend left, "\preccurlyeq"] & & y \arrow[ul, "\preccurlyeq"] \arrow[uul, bend right, "\preccurlyeq"'] &
\end{tikzcd}$$

Products and coproducts of arbitrary arity, if they exist, are similarly defined. For the case of meets, we prove the following universal property from first principles.

Lemma 3.3. Let $\mathbf{L}$ be a complete lattice, I an arbirary indexing set, $\{x_i\}_{i\in I} \subseteq \mathbf{L}$, and $y \in \mathbf{L}$. Then,

$$y \preccurlyeq \qquad \bigwedge_{i\in I} x_i$$

if and only if

$$y \preccurlyeq \qquad\qquad\qquad\qquad x_i$$

for all $i \in I$.

Proof. Suppose y is a lower bound of the subset $\{x_i\}_{i\in I}$. By definition, $\bigwedge_{i\in I} x_i$ is the greatest lower bound, hence, $\bigwedge_{i\in I} x_i \succcurlyeq y$. Conversely, if $y \preccurlyeq \bigwedge_{i\in I} x_i$, then y precedes the greatest lower bound of $\{x_i\}_{i\in I}$ implying y is also a lower bound by transitivity. $\square$

We encourage the reader, as we develop the theory of lattice-valued sheaves and sheaf Laplacians, to imagine how one could extend results from posets, meets, joins and Galois connections to categories, products, coproducts, and adjunctions, respectively.

3.4.4. Adjunctions

A Galois connection is a specialization of a more general construction in category theory called an adjunction.

Definition 3.7. Suppose $\mathcal{C}$ and $\mathcal{D}$ are categories and $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to \mathcal{C}$ are functors. Then, L is a left adjoint (equivalently, R is a right adjoint) written $L \dashv R$ if and only if there is a bijection

$$\mathrm{Hom}_{\mathcal{D}}(Fx, y) \leftrightarrow \mathrm{Hom}_{\mathcal{D}}(x, Gy)$$

for all $x \in \mathrm{Ob}\,\mathcal{C}$, $y \in \mathrm{Ob}\,\mathcal{D}$.

We have already seen that if $(f_\bullet, f^\bullet) : \mathbf{K} \to \mathbf{L}$ are Galois connections, the $f_\bullet$ and $f^\bullet$ are functors. It remains to see

$$\mathrm{Hom}_{\mathbf{L}}(f_\bullet(x), y) \leftrightarrow \mathrm{Hom}_{\mathbf{K}}(x, f^\bullet(y)). \tag{3.9}$$

Since $\mathbf{K}$ and $\mathbf{L}$ are posets, hom-sets $\mathrm{Hom}(x, y)$ is nonempty if and only if $x \preccurlyeq y$. It follows that (3.9) is equivalent to Definition 3.1.

While we explictly showed some key properties of Galois connections hold, we, could have derived from more far more general properties of adjunctions. For instance, it is widely known that left adjoints preserve colimits, including coproducts, and right adjoints preserve limits, including products (Theorem 3.1), or that adjoints satisfy the triangle identities (Proposition 3.2).

3.4.5. Relevant examples

The following are an exhaustive list of categories employed in this manuscript.

Definition 3.8 (Relevant Categories)**.**

1. Sets and functions: $\mathcal{S}et$.
2. Vector spaces over a field $\Bbbk$ and linear transformations: $\mathcal{V}ec_{\Bbbk}$
3. Real or complex Hilbert spaces spaces and linear transformations: $\mathcal{H}il_{\mathbb{R}}$ or $\mathcal{H}il_{\mathbb{C}}$.
4. Posets and order-preserving maps: $\mathcal{P}os$.
5. Monoids and monoid homomorphisms: $\mathcal{M}on$.
6. Complete lattices and join-preserving maps: $\mathcal{S}up$
7. Complete lattices and meet-meet preserving maps: $\mathcal{I}nf$.
8. Complete lattices and Galois connections: $\mathcal{L}tc$. The objects are complete lattices and the morphisms are pairs $(f_\bullet, f^\bullet)$ of lower/upper adjoints. Composition is by

$$\begin{aligned} \circ : \quad & \mathrm{Hom}_{\mathcal{L}tc}(\mathbf{L}, \mathbf{M}) \times \mathrm{Hom}_{\mathcal{L}tc}(\mathbf{K}, \mathbf{L}) \to \mathrm{Hom}_{\mathcal{L}tc}(\mathbf{L}, \mathbf{M}) \\ (g_\bullet, g^\bullet) \circ (f_\bullet, f^\bullet) = \quad & (g_\bullet f_\bullet, f^\bullet g^\bullet) \end{aligned}$$

CHAPTER 4

Networks & Signals

Networks are ubiquitous, even when we do not realize it. Some examples of networks include social networks, neuronal networks, electrical networks, sensor networks, coauthorship networks, communication networks, transportion networks, and correlation networks. Often the structure of a network determines its function function [35], while other times it is the other way around [163]. Dynamical systems supported on networks, *networked dynamical systems*, exhibit behavior that is more complex than than dynamical systems on their own [162]. *Signal processing* is one view of networked dynamical systems, and it is the view we adopt.

4.1. Networks

We model networks with graphs. Recall, a *multiset*, denoted $\mathbb{N}[S]$ is a set S with a map $S \to \mathbb{N}$ encoding the multiplicity of each element of S. A *directed (multi)graph* is a tuple $G = (\mathcal{V}_G, \mathcal{E}_G)$ with $\mathcal{V}_G$ a set of *nodes* and $\mathcal{E}_G \in \mathbb{N}[\mathcal{V}_G \times \mathcal{V}_G]$ is a (multi-)set of *edges*.[7] An *undirected (multi)graph* is consists of the data $G = (\mathcal{V}_G, \mathcal{E}_G)$ with $\mathcal{E}_G \in \mathbb{N}[\mathcal{V}_G \times \mathcal{V}_G / \sim]$ where we identify the $(i, j) \in \mathcal{V}_G \times \mathcal{V}_G$ with (j, i) by $(i, j) \sim (j, i)$. A graph is *simple* if $\mathcal{E}_G$ is a set and $\mathcal{E}_G$ has an empty intersection with the diagonal $\Delta(\mathcal{V}_V) = \{(i, j) \in \mathcal{V}_V \times \mathcal{V}_V \mid i = j\}$. *We hereafter assume all graphs are simple undirected graphs, unless otherwise noted. We denote nodes with lowercase letters i, j, i', j' etc. and denote edges with pairs of lowercase letters $ij, i'j'$ etc.*

In a graph, a *path* γ from $i \in \mathcal{V}_G$ to $j \in \mathcal{V}_G$ is a sequence of nodes $\{i = i_0, i_1, \ldots, i_\ell = j\}$ such that $(i_k, i_{k+1}) \in \mathcal{E}_G$ for all $0 \leqslant k \leqslant \ell$; we write the legnth of γ as $|\gamma| = \ell$. The opposite path $\gamma^{-1} = (j = i_\ell, i_{\ell-1}, \ldots, i_0 = i)$ is a path from j to i. A path is a *loop* if it has the same source and target nodes. The *diameter* of a graph is the length of the longest

[7] Traditionally, a directed multigraph is called a *quiver* and, equivalently, presented as the data of $Q = (\mathcal{V}_Q, \mathcal{E}_Q, h, t)$. *Head* and *tail* are maps $h, t : \mathcal{E}_Q \to \mathcal{V}_Q$ sending an edge (arrow) to its head and tail, respectively.

non-intersecting path $\operatorname{diam}(G)$. Paths form a category as follows.

Definition 4.1 (Free Category)**.** Suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a category. The *free category* of G is a category $\operatorname{Free}(G)$ with objects $\mathcal{V}_G$ and morphisms between $i, j \in \mathcal{V}_G$ given by paths $\gamma : i \to j$. Composition of paths is concatenation written from left to right $\gamma_1 \cdot \gamma_2$ and the identity is the trivial path denoted ε_i for each node $i \in \mathcal{V}_G$.

Suppose $i \in \mathcal{V}_G$. The *neighbors* of i are the set $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}_G\}$. The cardinality of $\mathcal{N}_i$ is called the *degree* of i, denoted d_i. Given an edge $(i, j) \in \mathcal{E}_G$, the *boundary* is the set $\partial(i, j) = \{i, j\}$. Given a node $i \in \mathcal{V}_G$, the *coboundary* is the (multi)set $\partial i = \{e \in \mathcal{E}_G \mid i \in \partial i\}$. For an edge $ij \in \mathcal{E}_G$, the *edge neighbors* consist of $\mathcal{N}_i = \delta i \cup \delta j - ij$. We say i is *incident* to $e \in \mathcal{E}_G$ if $i \in \partial e$. G is *d-regular* if $d_i = d$ for every $i \in \mathcal{V}_G$. G is complete *complete* if $(i, j) \in \mathcal{E}_G$ for all $i, j \in \mathcal{E}_G$, $i \neq j$. A graph is *connected* if for every $i, j \in \mathcal{V}_G$, there is a path from i to j.

If G has n nodes, then square $n \times n$ matrices encode graph structure with their sparsity patterns. A weighted graph is a graph G with weight function $\omega : \mathcal{E}_G \to \mathbb{R}_+$. A weight function is frequently encoded as a $n \times n$ matrix $[A]_{ij} = a_{ij}$ called the *adjacency matrix* of the weighted graph

$$[A]_{ij} = \begin{cases} \omega(i, j), & \text{if } ij \in \mathcal{E}_G \\ 0, & \text{otherwise} \end{cases}.$$

Hence, a weighted graph is given as the data $G = (\mathcal{V}_G, \mathcal{E}_G, A)$. The *degree matrix* $[D]_{ij} = d_{ij}$ of G is the $n \times n$

$$[D]_{ij} = \begin{cases} \sum_{j=1}^{n} a_{ij}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}.$$

Remark 4.1 (Unweighted Graphs)**.** If a G is not weighted, then we set

$$\omega(i,j) = \begin{cases} 1 & ij \in \mathcal{E}_G \\ 0 & \text{otherwise} \end{cases}$$

Then, d_{ii} is the actual degree d_i of $i \in \mathcal{V}_G$.

4.2. Graph Signals

Graph matrices are highly useful for answering questions about signals on graph, such as, is a signal smooth are there Graph signal processing [126] is concerned with $\mathbb{R}$-valued signals on graph which are functions $\mathbf{x} : \mathcal{V}_G \to \mathbb{R}$. Graph *shift operators* are matrices S that encode the graph topology via their sparsity structure and act on graph signals. Here are a few shift operators.

- The *adjacency matrix* A

$$(A\mathbf{x})_i = \sum_{j \in \mathcal{N}_i} a_{ij} x_j \qquad ;$$

- The *Laplacian matrix* $L = D - A$

$$\begin{aligned} (L\mathbf{x})_i &= \sum_{j \in \mathcal{N}_i} a_{ij}(x_i - x_j) && (4.1) \\ &= d_{ii}x_i - \sum_{j \in \mathcal{N}_i} a_{ij} x_j; && (4.2) \end{aligned}$$

- The *random walk Laplacian matrix* $P = D^{-1}A$

$$(P\mathbf{x})_i = \sum_{j \in \mathcal{N}_i} \frac{1}{d_{jj}} a_{ij} x_j \qquad ; \qquad (4.3)$$

- Matrix normalizations of the above.

In all cases, applying the graph shift operator to a graph signal results in a diffused signal. Components of the signal are integrated with the components of neighboring nodes. Iterating the shift operator repeatedly with initial condition $\mathbf{x}[0]$ yields various difference equations. For instance,

$$\begin{aligned} \mathbf{x}[t+1] = &\quad \varepsilon S\mathbf{x}[t]; \\ \mathbf{x}[0] \in &\quad \mathbb{R}^n, \end{aligned} \tag{4.4}$$

or,

$$\begin{aligned} \mathbf{x}[t+1] - \mathbf{x}[t] = &\quad -\varepsilon S\mathbf{x}[t]; \\ \mathbf{x}[0] \in &\quad \mathbb{R}^n \end{aligned} \tag{4.5}$$

for step-size $\varepsilon > 0$.

Spectral properties of the graph shift operator reflect on the nature of the diffusion process. Intricate descriptions of the diffusion based on the eigenvalues and eigenvectors of S exist [36], but coarse results are based on the following facts, for the specific case $S = L$, the graph Laplacian.

Fact. The time-independent solution of equation (4.5) are Ker L whose dimension is equal to the number of connected components of G.

Proof. Solutions of (4.5) must satisfy

$$0 = \quad \mathbf{x}[t+1] - \mathbf{x}[t] = \quad -L \cdot \mathbf{x}[t]$$

implying $\mathbf{x}[t]$ is in the kernel of $\mathbf{L}$. □

Fact. Suppose $\varepsilon < 2d_{\max}$, the maximum degree of G. Then, the discrete-time linear time-invariant system (4.5) is globally exponentially stable.

Proof. Let $P = 1 - \varepsilon L$. The graph Laplacian is easily seen to be positive semi-definite. Hence, every eigenvalue is non-negative and has no complex part. By Gershgorin's Theorem [82], the eigenvalues of L lie in the union of closed disks

$$\bigcup_{i=1}^{n} \{z \mid |z - [L]_{ii}| \leqslant \sum_{i \neq j} |[L]_{ij}|\} \subseteq \mathbb{C}.$$

Since the eigenvalues are real non-negative, we can conclude that the eigenvalues of L are between $0 \leqslant \lambda \leqslant 2d_{\max}$. If λ_i is the i^{th} eigenvalue of L, then $\mu_i = 1 - \varepsilon\lambda_i$ is the i^{th} eigenvalue of P. Hence, if $\varepsilon < 1/2d_{\max}$, then

$$\begin{aligned} \mu_i = & & 1 - 1/2d_{\max} \cdot \lambda_i \\ < & & 1 - 1/2d_{\max} \cdot 2d_{\max} \\ = & & 1 \end{aligned}$$

Since the eignenvalues of P have magnitude less than one, we conclude from elementary linear systems theory [34] the system 4.5 is globally exponentially stable. □

Together, these imply the graph Laplacian diffusion process stabilizes at *locally constant signals*, signals constant on connected components of the graph.

Graph signal processing [126] harnesses graph diffusion in order to filter graph signals at different scales. (Linear) *graph filters* are linear transformations represented by matrices

$$H = \quad \sum_{k=0}^{\infty} h_k S^k$$

Filters act on signals by way of $\mathbf{z} = H\mathbf{x}$. Symmetric graph shift operators admit an orthonormal decomposition $S = V\Lambda V^{\top}$ where Λ is a diagonal matrix of eigenvalues and V is a column matrix of eigenvectors. Thus, maybe the best way to see what filters "do" is via the *graph Fourier transform*: $\hat{\mathbf{x}} = V^{\top}\mathbf{x}$. The upshot is that *filters* are component-wise determined by polynomials $\sum_{k=0}^{\infty} h_k \lambda_i^k$ in the Fourier domain for each eigenvalue λ_i of S.

4.2.1. Higher-order networks

Although this work is focused on ordinary networks, we can't help but mention higher-order networks because the sheaf Laplacian presented in Chapter 7 is defined on these objects as well, as even the unimaginative reader may ascertain.

A *simplicial complex* $\mathbb{X}$ is a collection of subsets of a node set $\mathcal{V}_G$ such that if $\sigma \in \mathbb{X}$ and $\tau \subseteq \sigma$, then $\tau \in \mathbb{X}$. As a consequence, individual nodes $i \in \mathcal{V}_G$ are elements of $\mathbb{X}$. Simply put, a simplicial complex is a *set closed under taking subsets.* An a single subset is called *simplex* (plural: *simplicies*). Simplices are graded by dimension $\dim \sigma = |\sigma| + 1$. Let $\mathbb{X}_k$ denote the set of $\sigma \in \mathbb{X}$ such that $|\sigma| = k+1$. These subsets are called *k-simplices.* Simplices, as do graph, have boundaries and coboundaries consisting of $\partial(\sigma) = \{\tau \in \mathbb{X}_{\dim \sigma - 1} \mid \tau \subsetneq \sigma\}$ and $\delta(\sigma) = \{\beta \in \mathbb{X}_{\dim \sigma + 1} \mid \beta \supsetneq \sigma\}$.

Simplicial complexes, like graphs, are purely combinatorial objects, but they can be realized as subsets of Euclidean space, much as graphs can be embedded. The *standard* n-simplex is the space $\Delta[k] = \big\{(t_0, \ldots, t_k) \in \mathbb{R}^{k+1} \mid \sum_i t_i = 1,\ t_i \geqslant 0\ \forall i\big\}$. The *geometric realizaton* of $\mathbb{X}$ is the topological space $|\mathbb{X}|$ formed by gluing together standard simplices. In this way, we can interpret a simplicial complex as a collection of points, line segments, filled-in triangles, filled-in tetrehedra, and so forth. It is natural to consider face relations here instead of simple incidence relations. We write $\sigma \trianglelefteqslant \tau$ if $\sigma \subseteq \tau$ and $|\tau| = |\sigma| + 1$. The face relation poset $(\mathrm{Face}(\mathbb{X}), \trianglelefteqslant)$ is the transitive reflexive closure of the face relation. A simplicial complex $\mathbb{X}$ is *orientated* if every k-simplex is endowed with an ordering $\tau = [i_0 i_1 \ldots i_k]$. If $\sigma \trianglelefteqslant \tau$, then, neccesarily, σ consists of the data of a cyclic permutation of $\{i_0, i_1, \ldots, i_k\}$. We say the *sign* of σ is the parity of this permuation.

Hypergraphs generalize simplicial complexes, are far more practical, yet less understood. Hypergraphs model higher-order networks where relationships between agents are not required to be pair-wise (e.g. group chats), and a relationship between a number of agents does not necessarily mean that each subset of agents also share a relationship on their own.

A *hypergraph* H is a tuple $(\mathcal{V}_H, \mathcal{H}_H)$ where $\mathcal{H}_H \subseteq \wp(\mathcal{V}_G)$. As such, a hypergraph inherits the partial order on $\wp(\mathcal{V}_G)$ bestowing a hypergraph with the structure of a poset $(\mathcal{H}_H, \subseteq)$. A simplicial complex, then, is synonymous with a down-closed hypergraph.

Finally, (regular) cell complexes are another generalization of simplicial complexes of a more topological flavor. Let $\mathbb{D}^k = \{x \in \mathbb{R}^{k+1} \mid \|x\| \leqslant 1\}$ be the unit disk (ball) of dimension k. A *regular cell complex* is constructed from sequence of topological spaces

$$\emptyset = \mathbb{X}_{-1} \subseteq \mathbb{X}_0 \subseteq \mathbb{X}_1 \subseteq \mathbb{X}_2 \subseteq \cdots$$

such that $\mathbb{X}_k$ is obtained from X_{k-1} by "gluing"copies of $\mathbb{D}^k$ to $\mathbb{X}_k$. For more facts about simplicial complexes and cell complexes consult a textbook on algebraic toplology [81]. Parity of faces as well as face relations port from simplicial complexes to this slightly more general setting.

4.3. Representations of (Hyper)networks

We discuss various order-theoretic and other combinatorial representation of graphs, hypergraphs and simplicial complexes

1. *Incidence poset.* Suppose G is a graph. Incidence forms a relation on the nodes G denoted with the symbol $\trianglelefteqslant$ according to the rule: $i \trianglelefteqslant e$ if $i \in \partial e$. For each edge ij, ther eare exactly two incidence relations $i \trianglelefteqslant ij \trianglerighteqslant j$. The transitive-reflexive closure of $\trianglelefteqslant$ completes the incidence relation to a partial order on the union $\mathcal{V}_G \cup \mathcal{E}_G$ which is denoted, in abuse of notation, P_G. We call this poset the *incidence poset* of G denoted P_G

2. *Face relation poset.* More generally, the simplices (cells) of a simplicial complex $\mathbb{X}$ (regular cell complex) form a relation with $\sigma \trianglelefteqslant \tau$ if and only if $\sigma \in \partial\tau$. The transitive reflexive closure of this relation is a poset denoted $\mathrm{Face}(\mathbb{X})$.

Hypergraphs are notoriously unwieldy due to their lack of scalability and uniformity. Hence, it is popular to *represent* hypergraphs with other structures, such as (undirected) graphs.

3. *Clique expansion.* The clique expansion of H is the graph G_c with $\mathcal{V}_{G_c} = \mathcal{V}_H$ and $ij \in \mathcal{E}_{G_c}$ if and only if there exists a hyperedges contianing both i and j.

4. *Line graph.* The line graph of H is the graph G_ℓ with $\mathcal{V}_{G_\ell} = \mathcal{H}_H$ and $ee' \in \mathcal{E}_{G_\ell}$ if and only if $e \cap e' \neq \emptyset$.

5. *Star espansion.* The star expansion is the bipartite graph G_s with $\mathcal{V}_{G_s} = \mathcal{V}_H \cup \mathcal{H}_H$ and $(i, e) \in \mathcal{E}_{G_s}$ if and only if $i \in \mathcal{V}_H$, $e \in \mathcal{H}_H$ and $i \in e$.

A *hypergraph signal* is simply a map $\mathbf{x} : \mathcal{V}_H \to \mathbb{R}$. One benefit of these graph representations is that they unleash a variety of ways to process hypergraph signals via graph signal processing.

We turn our attention to various order-theoretic approachs to representing hypergraphs. The lattice of down-closed sets $\mathcal{D}(\mathcal{H}_H)$ is a finite distributive lattice. If a hypergraph is a simplicial complex, then every hyperedge is down-closed, hence included in $\mathcal{D}(\mathcal{H}_H)$. The data of the star expansion is equivalent to the data of a binary *membership relation* $\mathcal{I} \subseteq \mathcal{V}_H \times \mathcal{H}_H$ with $i\mathcal{I}e$ if and only if node $i \in e$. The Galois lattice $\mathrm{Gal}(\mathcal{I})$, then, offers another lattice representaiton of a hypergraph. Of note, the Galois lattice $\mathrm{Gal}(\mathcal{I})$ is isomorphic to the lattice of subsets σ of nodes such that the intersection of the set of hyperedges containing every node in σ is again σ. Alternatively $\mathrm{Gal}(\mathcal{I})$ is isomorphic to the lattice of subsets τ of hyperedges such that the set of hyperedges containing every node in the joint intersection of nodes belonging to τ is again τ.

CHAPTER 5

Sheaves

Sheaf theory is the mathematical study of the "local to global." This is why it is not surprising that sheaf theory has been applied in settings where local data (behavior, information) extends to global data [62, 116]. In the general setting, sheaves are indexed by a poset of open sets. Recall, a $\mathbb{X}$ be a topological space. Then, the open subsets of $\mathbb{X}$, denoted $\mathrm{Open}(\mathbb{X})$, is a poset with inclusion $(\mathrm{Open}(\mathbb{X}), \subseteq)$. Then, a sheaf valued in a data category $\mathcal{D}$ over a topological space $\mathbb{X}$ is a contravariant functor

$$\underline{\mathcal{F}} : \mathrm{Open}(\mathbb{X})^{\mathrm{op}} \to \mathcal{D}$$

In this chapter, we discuss a combinatorial version of sheaf theory called *network sheaf theory.* Instead of data being stored over open sets, it is stored over nodes and edges of an undirected graph. More details on general sheaf theory are found in Appendix A.

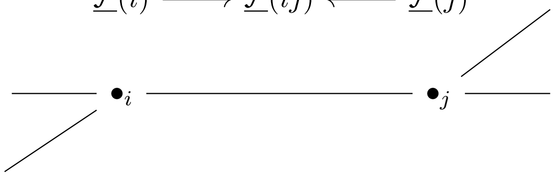
5.1. Network Sheaves & Cosheaves

Sheaves are a powerful data structure for interpreting multi-agent systems with near-arbitrary flexibility as far as the type of information/data each agent retrieves and exchanges with her its neighbors. Sheaves encode assignments of data to a geometric or topological structure. Examples of structures that one can build sheaves over are many but some common ones are graphs, simplicial complexes, cell complexes, posets, topological spaces, sites and varieties. While some of these "sheaf theories" are subtle, *network sheaf theory* is not.

Definition 5.1 (Network Sheaves & Cosheaves)**.** Suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a graph, P_G its incidence poset, and $\mathcal{D}$ a data category. A *network sheaf* is a functor $\underline{\mathcal{F}} : \mathrm{P}_G \to \mathcal{D}$. A *network cosheaf* is a functor $\overline{\mathcal{F}} : {\mathrm{P}_G}^{\mathrm{op}} \to \mathcal{D}$.

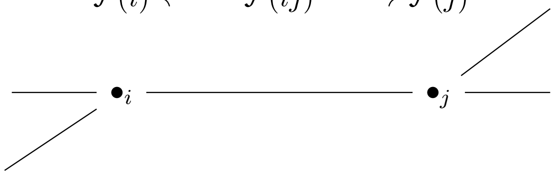
A sheaf $\underline{\mathcal{F}}$ assigns a $\underline{\mathcal{F}}(i)$ in $\mathcal{D}$ called a *stalk* to every $i \in \mathcal{V}_G$, a $\underline{\mathcal{F}}(ij)$ to every edge $ij \in \mathcal{E}_G$, and *restriction maps*

$$\underline{\mathcal{F}}(i) \xrightarrow{\underline{\mathcal{F}}(i \trianglelefteq ij)} \underline{\mathcal{F}}(ij) \xleftarrow{\underline{\mathcal{F}}(j \trianglelefteq ij)} \underline{\mathcal{F}}(j)$$



A cosheaf $\overline{\mathcal{F}}$ assigns a $\overline{\mathcal{F}}(i)$ in $\mathcal{D}$ called a *stalk* to every node $i \in \mathcal{V}_G$, a $\overline{\mathcal{F}}(ij)$ to every $ij \in \mathcal{E}_G$, and *corestriction maps*

$$\overline{\mathcal{F}}(i) \xleftarrow{\overline{\mathcal{F}}(ij \trianglerighteq i)} \overline{\mathcal{F}}(ij) \xrightarrow{\overline{\mathcal{F}}(ij \trianglerighteq j)} \overline{\mathcal{F}}(j)$$



A *local (co)section* is a partiular element $x_i \in \underline{\mathcal{F}}$ or $\overline{\mathcal{F}}(i)$.

If a (co)sheaf is a class (data structure), than a *(co)chain* in an instantiation of a (co)sheaf.

Definition 5.2 (Cochains). Suppose $\underline{\mathcal{F}}$ is a sheaf over G valued in a data category with products. The *cohains* of $\underline{\mathcal{F}}$ are the product

$$C^\bullet(\underline{\mathcal{F}}; G) = \prod_{\sigma \in \mathcal{V}_G \cup \mathcal{E}_G} \underline{\mathcal{F}}(\sigma)$$

We call the projection of $C^\bullet(G; \underline{\mathcal{F}})$ onto nodes the *0-cochains*

$$C^0(\underline{\mathcal{F}}; G) = \prod_{i \in \mathcal{V}_G} \underline{\mathcal{F}}(i)$$

and the projection onto edges the *1-cochains*

$$C^1(\underline{\mathcal{F}}; G) = \prod_{ij \in \mathcal{E}_G} \underline{\mathcal{F}}(ij).$$

Informally, (co)chains are sometimes called assignments to (co)sheaves [139]. A (co)chain is like a rough vector field [99] on a manifold in that tangent vectors in a neighborhood need not vary smoothly or continuously. . One of the successes of sheaf theory is the formalization of local-to-global phenomena. Historically, sheaf theory [24] has been motivated by the problem of extending (continuous, smooth, holomorphic, rational) functions defined on open sets in a (topological, manifold, complex manifold, variety) space to larger open sets. In the specialized case of network sheaves, the *local-to-global* problem asks *if a* 0*-cochain extend to a* 1*-cochain?* In the affirmative, a 0-cochian is said to be consistent.

Definition 5.3 (Sections)**.** Suppose $\underline{\mathcal{F}}$ is a sheaf and $\overline{\mathcal{F}}$ is a cosheaf over G. A *section* of $\underline{\mathcal{F}}$ is an assignment $\mathbf{x} \in C^{\bullet}(\underline{\mathcal{F}}; G)$

$$\underline{\mathcal{F}}_{i \trianglelefteq ij}(x_i) = \qquad x_{ij} = \qquad \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j) \tag{5.1}$$

agreeing over every edge ij. The collection of sections of $\underline{\mathcal{F}}$ are denoted $\Gamma\left(G; \underline{\mathcal{F}}\right)$.

The following example illustrates, even in a very simple example, that restriction maps as well as the topology of the graph determine whether or not consistent assignments exist.

Example 5.1 (Constant Sheaf)**.** Let G be an arbitrary graph. Let V be a vector space. The (linear) *constnat sheaf* denoted $\underline{V}$ is a sheaf valued in $\mathcal{V}ec$ with the following data

$$\begin{aligned} \underline{V}(i) &= \qquad V \\ \underline{V}(ij) &= \qquad V \\ \underline{V}_{i \trianglelefteq ij} &= \qquad \mathrm{id}_V. \end{aligned}$$

Of course the constnat sheaf makes sense in other data categories. It is the network sheaf with a particular object over every stalk and identity restriction maps.

Example 5.2 (A Twisted Sheaf)**.** Let $\underline{\mathcal{F}}$ be the sheaf over a 3-clique (Figure 5.1). In order for a 0-cochain to extend to a 1-cochain,

$$x_i = \qquad\qquad\qquad\qquad -x_j$$

for every edge ij. However, if agreement is reached over two edges, then, agreement is impossible over the remaining edge because the assignemnts to its endpoints are the same. Hence, the only global section is the zero vector $\mathbf{x} = \mathbf{0}$.

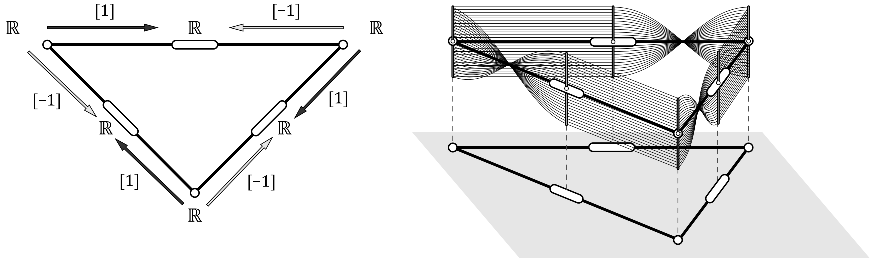


Figure 5.1: A twisted sheaf.

How do you calculate sections of a sheaf? An orientation on G is a pair of maps

$$(-)_{+,-} : \mathcal{E}_G \to \mathcal{V}_G,$$

sending an edge ij to its positive and negative endpoints $ij_+ \cup ij_- = \{i, j\}$. Given a choice of orientation, sections are equivalent to the following equalizer

$$\begin{tikzcd} H^0(G; \underline{\mathcal{F}}) \arrow[r, dashed] & C^0(G; \underline{\mathcal{F}}) \arrow[r, shift left, "\delta_-"] \arrow[r, shift right, "\delta_+"'] & C^1(G; \underline{\mathcal{F}}) \end{tikzcd} \tag{5.2}$$

where δ_-, δ_+ are defined component-wise

$$(\delta_- \mathbf{x})_{ij} = \qquad\qquad \underline{\mathcal{F}}(ij_- \trianglelefteq ij),$$

$$(\delta_+ \mathbf{x})_{ij} = \qquad\qquad \underline{\mathcal{F}}(ij_+ \trianglelefteq ij).$$

The maps $\delta_{-,+}$ are called *coboundary maps.* The equalizer condition specifies that $H^0(G; \underline{\mathcal{F}})$

is the largest subobject of $C^0(G; \underline{\mathcal{F}})$ such that

$$(\delta_- \mathbf{x})_{ij} = \qquad\qquad (\delta_+ \mathbf{x})_{ij}$$

for every $\mathbf{x} \in H^0(G; \underline{\mathcal{F}})$. In summary,

Proposition 5.1. $\Gamma\,(G; \underline{\mathcal{F}}) \cong H^0(G; \underline{\mathcal{F}})$.

An *additive category* is a data category in which maps have the structure of an abelian group; maps can be added, subtracted, and there is a zero map between any two objects.

Example 5.3 (Additive Category)**.** Suppose data category $\mathcal{D}$ is an additive category. Then, the equalizer (5.2) can be rewritten as the equalizer

$$\begin{tikzcd} H^0(G; \underline{\mathcal{F}}) \arrow[r, dashed] & C^0(G; \underline{\mathcal{F}}) \arrow[r, shift left, "\delta_+ - \delta_-"] \arrow[r, shift right, "0"'] & C^1(G; \underline{\mathcal{F}}) \end{tikzcd} \tag{5.3}$$

Then, $H^0(G; \underline{\mathcal{F}}) \cong \mathrm{Ker}\,(\delta_+ - \delta_-)$.

The consequence of the above example is that calculating cohomology, at least in the category of vector spaces, reduces to calculating the kernel of linear transformation. Of course, many information systems exhibit vector-valued data, but others exhibit other data types. Returning to the equalizer (5.2), if the data category is not additive, there is no clear path forward. We cannot write the equalizers as a kernel becuase each restriction map in a $\mathcal{D}$-valued network sheaf has no additive inverse. Hence, standard practices of linear and homological algebra are of no avail. We have already encountered categories that fail to be additive: the category $\mathcal{S}up$ of complete lattices and join-preserving maps, as well as the category $\mathcal{I}nf$ of complete lattices and meet-preserving maps.

5.2. Homology & Cohomology

In the past 20 years, applied algebraic topology [64] has flourished into a mature research community. Key to this development is *persistent homology* and software computing it [127]. Suppose $\mathbb{X}$ is a simplicial complex. Then, the *homology of* $\mathbb{X}$ *in degree* k, denoted $H_k(\mathbb{X})$, is a vector space whose basis elements correspond to independent connected components ($k = 0$), cycles ($k = 1$), voids ($k = 2$), and higher dimensional versions of these ($k > 2$). The difficulty with using homology directly to understand data is that it is not robust to perturbations. This leads to the notion of persistent homology introduced in the early 2000s [49]. A filtration of simplicial complexes (spaces) is a collection of subcomplexes (subspaces) $\{\mathbb{X}\}_{t\in \mathrm{P}}$ indexed by some poset P such that

$$\mathbb{X}_s \subseteq \mathbb{X}_t \text{ whenever} \qquad\qquad s \preccurlyeq t$$

in P. Homology is functorial in the sense that continuous function between space $f : \mathbb{X} \to \mathbb{Y}$ induces a linear transformation between vector spaces $f : H_k(\mathbb{X}) \to H_k(\mathbb{X})$ [81]. Hence, the inclusion maps of a filtration induce a series of maps $H_k(\mathbb{X}_s) \to H_k(\mathbb{X}_t)$ whenever $s \preccurlyeq t$. Filtrations indexed by a chain C have point-wise decomposition into subobjects called a barcode [128] which has been highly successful in summarizing the topology of data [59].

We have not said anything about how homology or persistent homology is actually computed. This is because network sheaves have cohomology, a dualization of homology. Cohomology is defined in a general algebraic setting, but is equivalent to homology in field coeficients by the usual isomoprhism between a vector space and its dual.

Definition 5.4 (Cochain Complex)**.** Suppose $\mathcal{D}$ is a category with a zero map. A *(bounded) cochain complex* is a sequence of objects $C^\bullet = \{C^k\}_{k\geqslant 0}$ with maps between them called *coboundary maps*

$$C^0 \xrightarrow{\delta^0} C^1 \xrightarrow{\delta^1} C^2 \xrightarrow{\delta^2} C^3 \xrightarrow{\delta^3} \cdots$$

such that $\delta^{k+1}\delta^k = 0$ for every $k \geqslant 0$.

If $\mathcal{D}$ has kernels, (normal) images, and quotients (See [69]), the *cohomology* of $C^\bullet$ is

$$H^k(C^\bullet) = \qquad\qquad \text{Ker } (\delta^k)/\text{Im } (\delta^{k-1}).$$

5.3. Hodge Laplacians

Motivated by certain problems in multi-agent systems [124], decentralized algorithms for computing homology of simplicial complexes have been proposed [117]. In principle, in order to compute $H_k(\mathbb{X})$, a k-simplex (e.g. a node) can pass messages with neighboring k-simplices and eventually converge to a homology vector. These algorithms, while promising, have not been widely adopted. The concept, more broadly, of distributed computation of cohomology is pertinent to the study of sheaves, objects more conducive to distributive algorithms than unstructured simplicial complexes.

The following definition goes at least as far back as Eckmann [48]. Suppose $C^\bullet$ is a complex in the category of real (or complex) Hilbert spaces. Then, the *combinatorial Hodge Laplacian* is the linear transformation

$$\begin{aligned} \Delta_\bullet : & \quad C^\bullet \to C^\bullet \\ \Delta_\bullet = & \quad \delta^* \circ \delta + \delta \circ \delta^* \end{aligned} \tag{5.4}$$

The Hodge Laplacian integrates cochains in C^k with cochains in C^{k-1} and C^{k+1}, suggesting the following dynamical system, as a natural generalization of the *heat equation*

$$\dot{\mathbf{x}} = \quad -\Delta_k \mathbf{x}, \tag{5.5}$$

with initial condition $\mathbf{x}(0) \in C^k$.

Moreover, the interpretation of the stable points of the *heat equation* is justified by the following theorem.

Theorem 5.1 (Hodge Theorem)**.** Suppose $(C^\bullet, \delta^\bullet)$ is a cochain complex of inner-product spaces. Then,

$$H^k(C^\bullet) \cong \quad \text{Ker}\ (\Delta^k).$$

We provide a proof for the convenience of the reader.

Proof of the Hodge Theorem. For ease of notation, let $Z_k = \text{Ker }(\delta^k)$ and $B_k = \text{Im }(\delta^{k-1})$, called the *cocycles* and *coboundaries*, respectively. The Hodge Laplacian decomposes into a sum of symmetric, and hence positive semi-definite, linear transformations

$$\begin{aligned} \Delta_k &= \Delta_k^+ + \Delta_k^- \\ &= {\delta^k}^* \delta^k + \delta^{k-1} {\delta^{k-1}}^* \end{aligned}$$

called the *upper-* and *lower- Hodge Laplacians.* Recall, by definition of quotient vector spaces, in the category of finite-dimensional inner-product spaces

$$H^k(C^\bullet) \cong Z_k \cap B_k^\perp.$$

We have $x \in \text{Ker }(\Delta_k^+)$ if and only if $\langle x, \Delta_k^+ x\rangle_{C^k} = 0$ for all $x \in C^k$. Similarly, $x \in \text{Ker }(\Delta_k^-)$ if and only if $\langle x, \Delta_k^- x\rangle_{C^k} = 0$. Hence, it suffices to show

$$\begin{aligned} \text{Ker }(\Delta_k^+ + \Delta_k^-) &= \\ \text{Ker }(\Delta_k^+) \cap \text{Ker }(\Delta_k^-) &= Z_k \cap B_k^\perp. \end{aligned}$$

Thus, $x \in \text{Ker }(\Delta_k)$ if and only if

$$\begin{aligned} \langle \delta^k x, \delta^k x\rangle_{C^k} &= 0 \\ \langle {\delta^{k-1}}^* x, {\delta^{k-1}}^* x\rangle_{C^k} &= 0 \end{aligned}$$

which holds if and only if $x \in \text{Ker }(\delta^k) \cap \text{Ker }({\delta^{k-1}}^*)$.

We claim $\text{Ker }({\delta^{k-1}}^*) = B_k^\perp$. First, $x \in \text{Ker }({\delta^{k-1}}^*)$ if and only if

$$\langle {\delta^{k-1}}^* x, y\rangle_{C^{k-1}} = 0.$$

for every $y \in C^{k-1}$. Then,

$$\langle x, \delta^{k-1} y\rangle_{C^k} = \langle {\delta^{k-1}}^* x, y\rangle_{C^{k-1}} = 0$$

showing that x is orthogonal to every coboundary if and only ${\delta^{k-1}}^* x = 0$. □

By basic theory of ordinary differential equations, the solution to the heat equation (5.5) is written in closed form

$$\mathbf{x}(t) = e^{-\Delta_k t}\mathbf{x}(0).$$

Because Δ_k is a positive semidefinite linear operator, the Hodge Theorem implies $x(t)$ converges to a projection of $\mathbf{x}(0)$ onto $H^k(C^\bullet)$ as $t \to \infty$. Thus, a näive algorithm for computing $H^k(C^\bullet)$ consists of the following steps.

1. Select a $\mathbf{x}(0) \in C^k$.
2. Compute the trajectory of the heat equation (5.5) with initial condition $\mathbf{x}(0)$.
3. Add $\mathbf{x}(\infty)$ to a growing basis of $H^k(C^\bullet)$.

A strategy to compute cohomology, such as the one above, in prinicple could be parallelized because because the Hodge Laplacian is a local operator. However, it would be our guess that practical difficulties lie in numerical stability. On the other hand, calculating the entire cohomology subspace is not always practical or necessary. Applications often call for calculating a single nearest or optimal cohomology class.

5.3.1. Sheaf Laplacians

We are primarily interested in dynamical systems on sheaves, because, after all, the final destination for this line or research is designing, learning, and controlling multi-agent systems. For sheaves valued in the data category of real or complex Hilbert spaces, the fundamental ingredients for a Hodge theory are present due to the following fact. For the following, suppose $\mathbb{X}$ is a simplicial complex or regular cell complex. Then, a *cellular sheaf* valued in a category $\mathcal{D}$ is a functor $\underline{\mathcal{F}} : \mathrm{Face}(\mathbb{X}) \to \mathcal{D}$. As a special case, if $\mathbb{X}$ is a graph, then a cellular sheaf over $\mathbb{X}$ is exactly the same data as a network sheaf.

Proposition 5.2. Suppose $\underline{\mathcal{F}} : \mathrm{Face}(\mathbb{X}) \to \mathcal{H}il$ is a sheaf of real or complex Hilbert spaces over a simplicial complex. Suppose each simplex $\sigma \in \mathbb{X}_k$ is encoded as an ordered tuple. Then, the following is a cochain complex

$$C^0(\mathbb{X};\underline{\mathcal{F}}) \xrightarrow{\delta^0} C^1(\mathbb{X};\underline{\mathcal{F}}) \xrightarrow{\delta^1} \cdots \xrightarrow{\delta^{k-1}} C^k(\mathbb{X};\underline{\mathcal{F}}) \xrightarrow{\delta^k} \cdots$$

with

$$\begin{aligned} C^k(\mathbb{X};\underline{\mathcal{F}}) &= \bigoplus_{\sigma \in \mathbb{X}_k} \underline{\mathcal{F}}(\sigma), \\ \left(\delta^k \mathbf{x}\right)_\tau &= \sum_{\sigma \trianglelefteq \tau} \mathrm{sgn}(\sigma) \cdot \underline{\mathcal{F}}(\sigma \trianglelefteq \tau)(x_\sigma). \end{aligned}$$

Proof. See [43]. □

Consequently, the *sheaf cohomoloy* $H^k(\mathbb{X};\underline{\mathcal{F}})$ is nothing but the cohomology of the complex $C^\bullet(\mathbb{X};\underline{\mathcal{F}})$, and the *linear sheaf Laplacian*

$$L_k : C^k(\mathbb{X};\underline{\mathcal{F}}) \to C^k(\mathbb{X};\underline{\mathcal{F}})$$

is nothing but the Hodge Laplacian of $C^\bullet$. Restricting, now, to the case of linear sheaves over graphs, network sheaves, the linear sheaf Laplacian ($k = 0$) has has the following form

$$(L\mathbf{x})_i = \sum_{j \in \mathcal{N}_i} \underline{\mathcal{F}}^*_{i \trianglelefteq ij} \left(\underline{\mathcal{F}}_{i \trianglelefteq ij}(x_i) - \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j)\right) \tag{5.6}$$

5.4. Examples of Sheaf Laplacians

The examples to follow show that, indeed, network sheaf Laplacians generalize both graph Laplacians and graph connection Laplacians, each of which, in turn, approximate smooth Laplacians on manifolds. Both the graph Laplacian and the graph connection Laplacian serve as inspiration for Laplacians of lattice-valued sheaves (Chapter 7).

5.4.1. Graph Laplacians

We now revisit the graph Laplacian with a "sheafy" outlook. Suppose $\mathbb{M} \subseteq \mathbb{R}^N$ is a k-dimensional smooth (Riemann) manifold. Recall, a smooth manifold consists of the data of a space $\mathbb{M}$ and a homeomorphism

$$\varphi_U : U \to \mathbb{R}^k$$

for every open set U called a *coordinate chart* such that

$$\varphi_V \varphi_U^{-1} : \varphi_U(U \cap V) \to \varphi_V(U \cap V)$$

is a differentiable for open sets $U \cap V \neq \emptyset$. A real-valued function $f : U \to \mathbb{R}$ on an open U is said to be *smooth* if $f \circ \varphi_U^{-1} : \varphi_U(U) \to \mathbb{R}$ is differentiable. Real-valued functions on open sets define sheaf[8] $\mathcal{O}_{\mathbb{M}}$ defined by the following data

$$\begin{aligned} \mathcal{O}_{\mathbb{M}} : \quad & \mathrm{Open}(\mathbb{M})^{\mathrm{op}} \to \mathcal{V}ec \\ \mathcal{O}_{\mathbb{M}}(U) = \quad & \{f : U \to \mathbb{R}\} \\ \mathcal{O}_{\mathbb{M}}(V \subseteq U)(f) = \quad & f_{|V}. \end{aligned}$$

If G is a weighted complete graph constructed from $\mathbb{M}$ by sampling points $\{x_1, x_2, \ldots, x_n\}$ uniformly from $\mathbb{M}$ and weighting edges inversely proportional to the distances of their boundary in $\mathbb{R}^N$, then the constant sheaf $\underline{\mathbb{R}}$ over G approximates the sheaf of smooth functions

[8] See [24], for example

on $\mathbb{M}$. This weighted graph is directly weighted to the following network sheaf

$$\begin{aligned} \mathcal{F}(i) &= \mathbb{R} \\ \mathcal{F}(ij) &= \mathbb{R} \\ \mathcal{F}_{i \trianglelefteq ij}(x_i) &= e^{-\frac{\|x_i - x_j\|^2}{4t}} \cdot x_i \end{aligned}$$

where $t > 0$. The sheaf Laplacian of $\underline{\mathcal{F}}$ is the calculated

$$\begin{aligned} (L\mathbf{x})_i &= \sum_{j \in \mathcal{N}_i} \left(e^{-\frac{\|x_i - x_j\|^2}{2t}} \right)^* \left(e^{-\frac{\|x_i - x_j\|^2}{2t}} x_i - e^{-\frac{\|x_i - x_j\|^2}{2t}} x_j \right) \\ &= \left(\sum_{j \in \mathcal{N}_i} e^{-\frac{\|x_i - x_j\|^2}{t}} \right) \cdot x_i - \sum_{j \in \mathcal{N}_i} e^{-\frac{\|x_i - x_j\|^2}{t}} \cdot x_j, \\ &= d_i x_i - \sum_{j \in \mathcal{N}_i} a_{ij} x_j \end{aligned}$$

We can identify graph Laplacians of arbitrary graphs as sheaf Laplacians in the same manner by setting stalks $\mathbb{R}$ and defining both restriction maps over an edge $ij \in \mathcal{E}_G$ to be multiplication by $\sqrt{a_{ij}}$. In the above case, as the parameter t approaches zero, the graph Laplacain, and thus the sheaf Laplacian is shown to approximate the Laplace-Beltrami operator, a classical operator [100] acting on real-valued functions on $\mathbb{M}$ (i.e. the global sections of $\mathcal{O}$, i.e. the scalar fields on $\mathbb{M}$). For the following, suppose $\mathbb{M} \subseteq \mathbb{R}^N$ is a k-dimensional maniold with Laplace-Beltrami operator Δ, and suppose $\{x_1, \ldots, x_n\} \subseteq \mathbb{M}$ is collection of points sampled uniformly from $\mathbb{M}$. Set $t_n = n^{-\frac{1}{k+2+\alpha}}$ where $\alpha > 0$, and let $L_n^{t_n}$ be the graph Laplacian of the complete weighted graph on $\mathcal{V}_G = \{x_1, \ldots, x_n\}$ with weights

$$a_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t_n}}.$$

Theorem 5.2 ([15])**.** Suppose $f \in C^\infty(\mathbb{M})$. Then,

$$\lim_{n\to\infty} \frac{1}{t_n(5\pi t_n)^{\frac{k}{2}}} L_n^{t_n} f(x) = \qquad \frac{1}{\text{vol}(\mathbb{M})}\Delta f(x)$$

in probability.

5.4.2. Connection Laplacians

With the graph Laplacian being the first key example of a sheaf Laplacain, we introduce a second example example, the graph connection Laplacian.

Recall, some other elements from differential geometry [86]. Suppose $\mathbb{M}$ is a k-dimensional Riemanian manifold. To each element $x \in \mathbb{M}$, the *tangent space* T_x is a vector space whose elements are equivalence classes of curves $\gamma : (-1, 1) \to \mathbb{M}$ with $\gamma(0) = x$ whose derivatives coincide. In practice, you can visualize tangent spaces as tangent lines, tangent planes etc. The *tangent bundle* is the disjoint union $T\mathbb{M} = \bigsqcup_{x\in\mathbb{M}}$ together with the projection map $T\mathbb{M} \xrightarrow{p} \mathbb{M}$. A *vector field* is a section of the tangent bundle,

$$s : \mathbb{M} \to T\mathbb{M} \quad p \circ s = \text{id}_{\mathbb{M}};$$

the map s assigns every point $x \in \mathbb{M}$ a tangent vector $v \in T_x\mathbb{M}$. Sections of the tangent bundle define a sheaf[9]

$$\begin{aligned} \Gamma : &\qquad \text{Open}(\mathbb{M}) \to \mathcal{V}ec \\ \Gamma(U) = &\qquad \{s : U \to T\mathbb{M} \mid p \circ s = \text{id}_U\} \\ \Gamma(V \subseteq U)(s) = &\qquad s_{|V} \end{aligned}$$

Indeed, $\Gamma(U)$ is a vector space called *local sections.* $\Gamma(\mathbb{M})$ is the space of vector fields defined on the entire manifold.

Approximations of the tangent bundle lead to complications. Even if points sampled from

[9] See, for instance, [24].

a manifold are close, their tangent spaces will differ, in general. High-dimensional data sets are often assumed to have been sampled from low-dimensional manifold. This presents a problem because we cannot relate data in two separate local coordinates. Fortunately, this is a topic widely studied in modern differential geometry, called *parallel transport* [94]. The following is a definition or a theorem depending on how you look at it. We leave technical details to the curious reader.

Definition 5.5 ([86])**.** Suppose $\mathbb{M}$ is a k-dimensional manifold and γ is a smooth path $\gamma : [0,1] \to \mathbb{M}$ with $\gamma(0) = x$ and $\gamma(1) = y$. Then, there is a unique invertible linear transformation

$$\mathcal{P}_\gamma : T_x\mathbb{M} \to T_y\mathbb{M}$$

called *parallel transport* sending a tangent vector $v \in T_x\mathbb{M}$ to a tangent vector $\mathcal{P}_\gamma(v) \in T_y\mathbb{M}$.

Parallel transport leads to vector diffusion and its approximated by orthogonal linear maps. The following approximates the connection Laplacian [7], a close cousin of the Laplace-Beltrami operator and a Hodge Laplacian.

Definition 5.6 (Graph Connection Laplacian [11])**.** Fix a dimenison k, and suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a weighted graph with the data of an orthogonal linear transformation

$$O_{ij} : \mathbb{R}^k \to \mathbb{R}^k$$

Then, the *graph connection Laplacian* is the matrix ∇^2

$$\begin{aligned} \nabla^2 = && (\mathbb{R}^k)^n \to (\mathbb{R}^k)^n \\ (\nabla^2 \mathbf{v})_i = && d_i x_i - \sum_{j \text{ß} \mathcal{N}_i} a_{ij} O_{ij} v_j \end{aligned}$$

$i \in \mathcal{V}_G$.

As a consequence of an approximation result [153] analogous to Proposition 5.2, parallel transport between tangent spaces of the manifold are approximated in the graph connection Laplacian by orthogonal linear transformations O_{ij} between the tangent spaces $O_{ij} : T_{x_i}\mathbb{M} \to T_{x_j}\mathbb{M}$. The graph connection Laplacian is a sheaf Laplacian. Suppose $\{O_i \in \mathbb{R}^{k\times K}\}_{i\in\mathcal{V}_G}$ is a family of orthogonal matrices indexed by the nodes of a weighted graph G. Construct the sheaf as follows

$$\begin{aligned}
\underline{\mathcal{F}}(i) &= \qquad \mathbb{R}^k, \\
\underline{\mathcal{F}}(ij) &= \qquad \mathbb{R}^k, \\
\underline{\mathcal{F}}_{i\trianglelefteq ij}(x_i) &= \qquad \sqrt(a_{ij})O_i x_i
\end{aligned}$$

Then,

$$\begin{aligned}
(L\mathbf{x})_i &= \sum_{j\in\mathcal{N}_i} \sqrt{a_{ij}} O_i^* \left(\sqrt{a_{ij}} O_i x_i - \sqrt{a_{ij}} O_j x_j\right) \\
&= d_i x_i - \sum_{j\in\mathcal{N}_i} a_{ij} O_i^* O_j x_j
\end{aligned}$$

Set $O_{ij} = O_i^* O_j$ which is orthogonal.

CHAPTER 6

Lattice-Valued Sheaves

We introduce a novel class of network sheaves, study global sections of these sheaves, as well as mimick the classical sheaf operations [24].

6.1. Tarski Sheaves

We begin with a definition of a sheaf over a graph whose stalks are complete and whose restriction maps are Galois connections.

Definition 6.1 (Tarski Sheaf)**.** Suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a graph. A *Tarski sheaf* is a functor

$$\mathcal{F} : \mathrm{P}_G \to \mathcal{L}tc$$

A Tarski sheaf assigns a

1. Complete lattice $\mathcal{F}(i)$ to every node $i \in \mathcal{V}_G$,

2. Complete lattice $\mathcal{F}(ij)$ to every edge $ij \in \mathcal{E}_G$,

3. Galois connection

$$\begin{tikzcd} \mathcal{F}(i) \arrow[r, bend left, "\mathcal{F}(i \trianglelefteq ij)_{\bullet}"] & \mathcal{F}(ij) \arrow[l, bend left, "\mathcal{F}(i \trianglelefteq ij)^{\bullet}"] \end{tikzcd}$$

   for every $i \in \mathcal{V}_G$ and $j \in \mathcal{N}_i$.

Remark 6.1 (Notation)**.** Slightly simplifying cumbersome notation, we will write $(\underline{\mathcal{F}}(i \trianglelefteq ij), \overline{\mathcal{F}}(i \trianglelefteq ij)$ for the Galois connection $(\mathcal{F}(i \trianglelefteq ij)_{\bullet}, \mathcal{F}(i \trianglelefteq ij)^{\bullet})$.

A Tarski bisheaf contains the data of both a sheaf $\underline{\mathcal{F}}$ and a cosheaf $\overline{\mathcal{F}}$ by

$$\begin{aligned} \underline{\mathcal{F}}: && \mathrm{P}_G \to \mathcal{S}up, \\ \overline{\mathcal{F}}: && \mathrm{P}_G{}^{\mathrm{op}} \to \mathcal{I}nf. \end{aligned}$$

by selecting restriction maps to be lower adjoints and corestriction maps to be the upper adjoints.

Theorem 6.1. Suppose P is a poset. Then, the following categories are equivalences of categories

$$\begin{aligned} \mathcal{L}tc^{\mathrm{P}} \simeq && \mathcal{S}up^{\mathrm{P}} \\ \mathcal{L}tc^{\mathrm{P}} \simeq && \mathcal{I}nf^{\mathrm{P}^{\mathrm{op}}} \end{aligned}$$

Proof. It suffices to show $\mathcal{S}up \simeq \mathcal{L}tc$ because we obtain a natural isomorphism $\mathcal{L}tc^{\mathrm{P}} \simeq \mathcal{S}up^{\mathrm{P}}$ from a natural isomoprhism $\mathcal{L}tc \simeq \mathcal{S}up$.

$$\Pi : \mathcal{L}tc \to \mathcal{S}up$$

is an equivalence. It is a standard fact in category theory that it suffices to show Π is full, faithful, and essentially surjective [133, Theorem 1.5.9]. Full and faithful amounts to demonstrating a bijection

$$\mathrm{Hom}_{\mathcal{L}tc}(\mathbf{K}, \mathbf{L}) \cong \mathrm{Hom}_{\mathcal{S}up}(\mathbf{K}, \mathbf{L}).$$

which follows directly from Theorem 3.1. The argument that $\mathcal{I}nf^{\mathrm{op}} \simeq \mathcal{L}tc$, and hence, that $\mathcal{L}tc^{\mathrm{P}} \simeq \mathcal{I}nf^{\mathrm{P}^{\mathrm{op}}}$, from duality. □

The consequence of Theorem 6.1 is that the data of $\mathcal{F}$ is equivalent to the data $\underline{\mathcal{F}}$ or the data $\overline{\mathcal{F}}$.

Remark 6.2 (Bisheaves)**.** The notion of a "bisheaf" was introduced in the context of sheaves and cosheaves of abelian groups on simplicial complexes [108, 120]. Their definition of a bisheaf consists of a sheaf and a cosheaf over the same base (e.g. graph) together with a compatible homomorphism between the stalks of the sheaf to the stalks of the cosheaf. In some instances, our definition conincides with theirs. For example, if $\underline{\mathcal{F}}$ is a monosheaf with associated epicosheaf $\overline{\mathcal{F}}$, the *Tarski* bisheaf $(\underline{\mathcal{F}}, \overline{\mathcal{F}})$ is a bisheaf in the sense of McPhersen-Patel [108].

6.2. New Sheaves from Old

In the following examples, sheaves of concrete objects (sets, vector spaces, hilbert spaces) will be denoted in plain font; induced Tarski sheaves will be denoted in calligraphic font.

6.2.1. From sheaves of sets

Suppose $S : \mathrm{P}_G \to \mathcal{S}et$ is a set-valued network sheaf. Such a structure assigns the data of a set to every node and edge and (arbitrary) maps between these sets over edges. Sheaves of sets are perhaps the most basic examples of sheaves and have found their way to applications in electrical engineering, quantum theory, and topological data analysis [67, 2, 46]. The key construction is the following. Suppose $f : X \to Y$ is a function. Then, there is a Galois connection

$$\begin{tikzcd} \wp(X) \arrow[r, bend left, "f(-)"] & \wp(Y) \arrow[l, bend left, "f^{-1}(-)"] \end{tikzcd}$$

sending a subset of X to its image and a subset of Y to its preimage. Consequently, a sheaf $F : \mathrm{P}_G \to \mathcal{S}et$ factors through the powerset functor, introducing a class of Tarski sheaves

$$\begin{tikzcd} \mathrm{P}_G \arrow[r, "S"] \arrow[dr, "\mathcal{S}"] & \mathcal{S}et \arrow[d, "\wp(-)"] \\ & \mathcal{L}tc \end{tikzcd}$$

The following extended example can be safely skipped.

Example 6.1 (Reeb Graphs)**.** In topological data analysis, filtrations of a metric space by sublevel sets serve as a canonical example of a topological filtration. This example, roughly, subsumes Čech filtrations of point clouds [14]. The path-connected components of space is a functor $\pi_0 : \mathcal{T}op \to \mathcal{S}et$. π_0, in general, does not have a group structure. Given a cell decomposition of $\mathbb{R}$ with 0-cells

$$Z = \{t_0 < t_1 < \cdots < t_r\}$$

and a map $f : \mathbb{X} \to \mathbb{R}$, the *Reeb cosheaf* is a cosheaf valued in $\mathcal{S}et$ with stalks $\pi_0 f^{-1} I$ for open intervals $I \subseteq \mathbb{R}$ subordinate to the cell decomposition and with restriction maps induced by the functoriality of π_0 [46]. The global sections of the Reeb cosheaf correspond to path components of $\mathbb{X}$ that are born and die at particular critical values in Z. Sections of the induced Tarski sheaf consist of path components that copersist.

6.2.2. From vector-valued sheaves

Suppose $F : \mathrm{P}_G \to \mathcal{V}ec$ is a vector-valued network sheaf. Such a structure assigns a vector space to every node and edge and linear transformations between these vector spaces over edges. Sheaves of vector spaces are key to the foundations of persistent homology [43, Section 8.2]. Suppose $T : V \to W$ is a linear transformation. Then, there is a Galois connection

$$\begin{tikzcd}
\operatorname{Sub} V \arrow[r, bend left, "T(-)"] & \operatorname{Sub} W \arrow[l, bend left, "T^{-1}(-)"]
\end{tikzcd}$$

sending a subspace of V to its image and a subspace of W to its preimage. Hence, a sheaf $F : \mathrm{P}_G$ factors through the subspace functor[10], introducing a class of Tarski sheaves

$$\begin{tikzcd}
\mathrm{P}_G \arrow[r, "F"] \arrow[dr, "\mathcal{F}"] & \mathcal{V}ec \arrow[d, "\operatorname{Sub} -"] \\
 & \mathcal{L}tc
\end{tikzcd}$$

[10] Grandis calls this functor the *transfer functor* [69], we previously called it the *Grassmanian* [63]

The construction is unaltered in the (full sub)category $\mathcal{H}il$ of Hilbert spaces and linear transformations.

Example 6.2 (Constant Sheaves)**.** Suppose V is vector space and $\underline{V}$ is the constant sheaf over G. Then, the induced Tarski sheaf is again the constant sheaf $\underline{\operatorname{Sub} V}$.

6.3. New Sup-Lattices from Old

As a prerequisite for defining the sheaf operations for $\mathcal{S}up$-sheaves, we require categorical operations on complete lattices.

6.3.1. Products & coproducts

The product in $\mathcal{S}up$ is inherited from the product in $\mathcal{P}os$. Suppose P and Q are posets. Then, the *product* P × Q is the poset with the order $(x, y) \preccurlyeq (z, w)$ if and only if $x \preccurlyeq y$ and $y \preccurlyeq w$. The *coproduct* of P and Q, denoted $\mathrm{P} \coprod \mathrm{Q}$ is the poset on the disjoint union of P and Q (as sets) with $x \preccurlyeq y$ if $x, y \in \mathrm{P}$, $x, y \in \mathrm{Q}$, and $x \preccurlyeq y$ in either P or Q. Immediately we run into difficulty: completeness of P and Q does not imply completeness of $\mathrm{P} \coprod \mathrm{Q}$.

Frequently, we take cartesian products of sets indexed over an arbitrary set I

$$\prod_{i \in I} \mathbf{L}_i.$$

Elements of the product are denoted with bold letters $\mathbf{x}$. The with individual components denoted $(x_i)_{i \in I}$. The (categorical) *product* of a family of lattices $\{\mathbf{L}\}_{i \in I}$ is the cartesian product equipped with projection maps $\pi_i : \prod_{i \in I} \mathbf{L}_i \to \mathbf{L}_i$. Projection maps have lower and upper adjoints

$$\pi_i^*, \pi_{i*} : \mathbf{L}_i \to \prod_{i \in I} \mathbf{L}_i$$

$$(\pi_i^* x)_j = \begin{cases} x_i & i = j \\ 0 & i \neq j \end{cases} \qquad (\pi_{i*} x)_j = \begin{cases} x_i & i = j \\ 1 & i \neq j \end{cases}$$

The (categorical) *coproduct* of $\{\mathbf{L}_i\}_{i\in I}$ is the cartesian product with maps $\pi_{i*} : \mathbf{L}_i \to \coprod_{i\in I} \mathbf{L}_i$ called inclusion maps.

Products and coproducts in $\mathcal{S}up$ define the chains and cochains of a Tarski sheaf. Suppose $\mathcal{F}$ is a bisheaf over G. Then, the *0-cochains* of $\mathcal{F}$ are the (complete) product lattice

$$C^0(G;\underline{\mathcal{F}}) = \prod_{i\in\mathcal{V}_G} \underline{\mathcal{F}}(i)$$

together with the projection maps

$$\pi_i : C^0(G;\underline{\mathcal{F}}) \to \underline{\mathcal{F}}(i).$$

The *0-chains* of $\mathcal{F}$ is the coproduct lattice

$$C_0(G;\overline{\mathcal{F}}) = \coprod_{i\in\mathcal{V}_G} \overline{\mathcal{F}}(i)$$

with the inclusion maps

$$\pi_{i*} : \overline{\mathcal{F}}(i) \to C_0(G;\overline{\mathcal{F}}).$$

As the product and coproduct are isomorphic, then we idenity both (up to isomorphism) as the *biproduct* $\bigoplus_{i\in\mathcal{V}_G} \mathcal{F}(i)$ [133, Appendix E.5].

6.3.2. Tensor product & internal hom

Recall, that the set of join-preserving morphisms between two complete lattice forms a complete lattice called the *internal hom* which we hereafter denote $[\mathbf{K}, \mathbf{L}]$. Suppose $\{f_i\}_{i\in I} \subseteq [\mathbf{K}, \mathbf{L}]$. Then,

$$\left(\bigvee f_i\right)(x) = \bigvee_{i\in I} f_i(x).$$

The *tensor product* $\mathbf{K} \otimes \mathbf{L}$ is defined[11]

$$\mathbf{K} \otimes \mathbf{L} \cong \qquad [\mathbf{K}, \mathbf{L}^{\mathrm{op}}]^{\mathrm{op}}.$$

The set $\mathbf{K} \otimes \mathbf{L}$ consists of order-reversing maps $f : \mathbf{K} \to \mathbf{L}$ such that $f(\bigvee S) = \bigwedge f(S)$, or, in other words, contavariant Galois connections.

Example 6.3 (Powersets)**.** If X and Y are sets, then

$$\wp(X) \otimes \wp(Y) \cong \qquad \wp(X \times Y)$$

We have the following adjunction between the internal hom and tensor product.

Proposition 6.1. Suppose $\mathbf{K}, \mathbf{L}, \mathbf{M}$ are complete lattices. Then, $[\mathbf{K} \otimes \mathbf{L}, \mathbf{M}] \cong [\mathbf{K}, [\mathbf{L}, \mathbf{M}]]$.

Proof. See [87, Chapter 1]. □

6.3.3. Equalizers & coequalizers

"solving equations" in the $\mathcal{S}up$ amounts to boils down to computing equalizers and coequalizers.

Definition 6.2 (Equalizer & Coequalizer)**.** An *equalizer* of a pair of parrellel join-preserving maps $f, g : \mathbf{K} \to \mathbf{L}$ is a suplattice $\mathbf{E}$ and map $i : \mathbf{E} \to \mathbf{K}$ with $f \circ i = g \circ i$ such that for any map $j : \mathbf{M} \to \mathbf{K}$, there is a unique map $i' : \mathbf{M} \to \mathbf{E}$ with $i \circ i' = j$ as in the following diagram

$$\begin{tikzcd}
\mathbf{E} \arrow[r, "i"] & \mathbf{K} \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & \mathbf{L} \\
\mathbf{M} \arrow[u, dashed, "i'"] \arrow[ur, "j"'] & &
\end{tikzcd} . \tag{6.1}$$

A *coequalizer* of the pair f, g is a suplattice $\mathbf{Q}$ and a map $p : \mathbf{L} \to \mathbf{Q}$ with $p \circ f = p \circ g$ such that for any map $q : \mathbf{L} \to \mathbf{M}$, there is a unique map $p' : \mathbf{Q} \to \mathbf{M}$ with $p' \circ p = q$ as in the

[11] If you prefer a definition with a universal property, see [87].

following diagram

$$\begin{tikzcd}
\mathbf{K} \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & \mathbf{L} \arrow[r, "p"] \arrow[dr, "q"'] & \mathbf{Q} \arrow[d, dashed, "p'"] \\
 & & \mathbf{M}
\end{tikzcd} . \tag{6.2}$$

Universal properties, however, do not suffice for actual computations.

Proposition 6.2. $\mathcal{S}up$ has equalizers and coequalizers.

Proof. We claim $\mathbf{E} = \{x \in \mathbf{K} \mid f(x) = g(x)\}$ with the inclusion i into $\mathbf{K}$ is the equalizer of $f, g : \mathbf{K} \to \mathbf{L}$. First, $f \circ i = g \circ i$ since i is an inclusion. Second, $\mathbf{E}$ is complete, as we now show. Suppose $\{x_i\}_{i \in I} \subseteq \mathbf{E}$, then $\bigvee_{i \in I} x_i \in \mathbf{E}$ as

$$\begin{aligned}
f\left(\bigvee_{i \in I} x_i\right) &= \bigvee_{i \in I} f(x_i) \\
&= \bigvee_{i \in I} g(x_i) \\
&= g\left(\bigvee_{i \in I} x_i\right).
\end{aligned}$$

For coequalizers, let $\mathbf{Q} = \{y \in \mathbf{L} \mid f^*(y) = g^*(y)\}$. Then, the coequalizer of $f, g : \mathbf{L} \to \mathbf{K}$ is the lattice $\mathbf{Q}^{\text{op}}$ with the projection $i^* : \mathbf{Q}^{\text{op}} \to \mathbf{L}$. □

6.4. Sections

For a $\mathcal{S}up$-sheaf, we define cohomology H^0 and H^1 as equalizers and coequalizers. We show H^0 is isomorphic to the complete lattice of global sections. Recall, *sections* are assignments $\mathbf{x} \in C^\bullet(G; \underline{\mathcal{F}})$ such that that satisfy

$$\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) = \quad x_{ij} = \quad \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j) \quad i \trianglelefteq ij \trianglerighteq j \tag{6.3}$$

for every $i \trianglelefteq ij \trianglerighteq j$. Sections inherit the order on the product lattice $C^\bullet(G; \mathcal{F})$. Let $\Gamma\left(G; \underline{\mathcal{F}}\right) \subseteq C^\bullet(G; \mathcal{F})$ denote the poset of sections. In fact, $\Gamma\left(G; \underline{\mathcal{F}}\right)$ is a complete lattice calculated as follows.

Definition 6.3. Suppose $\mathcal{F}$ be a Tarski sheaf over G. Then, the let $H^0(G;\underline{\mathcal{F}})$ and $H^1(G;\underline{\mathcal{F}})$ to be the equalizer and coequalizer[12] of the following diagram.

$$\begin{tikzcd} H^0(G;\underline{\mathcal{F}}) \arrow[r, dashed] & C^0(G;\underline{\mathcal{F}}) \arrow[r, shift left, "\delta_-"] \arrow[r, shift right, "\delta_+"'] & C^1(G;\underline{\mathcal{F}}) \arrow[r, dashed] & H^1(G;\underline{\mathcal{F}}) \end{tikzcd} \tag{6.4}$$

with

$$\begin{aligned} (\delta_-)_{ij} &= \underline{\mathcal{F}}(ij_- \trianglelefteq ij), \\ (\delta_+)_{ij} &= \underline{\mathcal{F}}(ij_+ \trianglelefteq ij). \end{aligned}$$

By Proposition 6.2,

$$\begin{aligned} H^0(G;\underline{\mathcal{F}}) &\cong \{\mathbf{x} \in C^0(G;\underline{\mathcal{F}}) \mid (\delta_-\mathbf{x})_{ij} = (\delta_+\mathbf{x})_{ij} \quad \forall ij \in \mathcal{E}_G\} \\ &= \{\mathbf{x} \in C^0(G;\underline{\mathcal{F}}) \mid \underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) = \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j) \quad \forall ij \in \mathcal{E}_G\}. \\ &\cong \{\mathbf{x} \in C^0(G;\underline{\mathcal{F}}) \mid \underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) = x_{ij} = \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j)\} \\ &= \Gamma\left(G;\underline{\mathcal{F}}\right). \end{aligned}$$

Theorem 6.2 (Global Sections Theorem)**.** Suppose $\mathcal{F}$ is a Tarski sheaf over G. Then, $H^0(G;\underline{\mathcal{F}})$ is a complete quasi-sublattice of $C^0(G;\underline{\mathcal{F}})$

Warning 6.1. Beware, $H^0(G;\underline{\mathcal{F}})$ is not in general a sublattice. $H^0(G;\underline{\mathcal{F}})$ is an equalizer. Hence, the unique map $H^0(G;\underline{\mathcal{F}}) \to C^0(G;\underline{\mathcal{F}})$ is injective (see [133, Exercise 3.1.vi]). However, joins and meets in $H^0(G;\underline{\mathcal{F}})$ do not (in general) coincide with joins and meets in $C^0(G;\underline{\mathcal{F}})$.

.

[12] See Joyal [87, Chapter 1] for a proof that $\mathcal{S}up$ has all limits and colimits.

6.5. Sheaf Operations

We construct four of the six classical "sheaf operations:" *tensor product sheaves, sheaf hom, pullback and pushforward.* While these four sheaf operations do reappear, we include this exposition to inspire future work.

6.5.1. Sheaf morphisms & constant sheaves

In order to define tensor product sheaves and sheaf hom, we first recall morphisms between sheaves. A *sheaf morphism* $\underline{\mathcal{F}} \overset{\varphi}{\Longrightarrow} \underline{\mathcal{G}}$ between $\underline{\mathcal{F}}$ and $\underline{\mathcal{G}}$ consists of the data of join-preserving maps $\varphi_i : \mathcal{F}(i) \to \mathcal{G}(i)$ for every $i \in \mathcal{V}_G$ and join-preserving maps $\varphi_{ij} : \underline{\mathcal{F}}(ij) \to \underline{\mathcal{G}}(ij)$ for every $ij \in \mathcal{E}_G$ such that for every $i \in \mathcal{V}_G$ and $j \in \mathcal{N}_i$

$$\underline{\mathcal{G}}(i \trianglelefteqslant ij) \circ \varphi_i = \qquad\qquad \varphi_{ij} \circ \underline{\mathcal{F}}(i \trianglelefteqslant ij). \tag{6.5}$$

The set of all sheaf morphisms is denoted $\operatorname{Hom}(\underline{\mathcal{F}}, \underline{\mathcal{G}})$.

Proposition 6.3. $\operatorname{Hom}(\underline{\mathcal{F}}, \underline{\mathcal{G}})$ is a complete lattice.

Proof. Suppose I is an arbitrary indexing set and $\{\varphi^\alpha\}_{\alpha \in I}$ is a family of sheaf morphisms. For $i \in \mathcal{V}_G$, we write

$$\left(\bigvee_{\alpha \in I} \varphi_i^\alpha\right)(x_i) = \qquad\qquad \bigvee_{\alpha \in I} \varphi_i^\alpha(x_i),$$

and similarly for joins of the family φ_{ij}^α for $ij \in \mathcal{E}_G$. It follows the diagram

$$\begin{tikzcd}
\underline{\mathcal{F}}(i) \arrow[r, "\underline{\mathcal{F}}(i \trianglelefteqslant ij)"] \arrow[d, "\bigvee \varphi_i^\alpha"] & \underline{\mathcal{F}}(ij) \arrow[d, "\bigvee \varphi_{ij}^\alpha"] & \underline{\mathcal{F}}(j) \arrow[l, "\underline{\mathcal{F}}(j \trianglelefteqslant ij)"'] \arrow[d, "\bigvee \varphi_j^\alpha"] \\
\underline{\mathcal{G}}(i) \arrow[r, "\underline{\mathcal{G}}(i \trianglelefteqslant ij)"] & \underline{\mathcal{G}}(ij) & \underline{\mathcal{G}}(j) \arrow[l, "\underline{\mathcal{F}}(j \trianglelefteqslant ij)"']
\end{tikzcd}$$

commutes for all $ij \in \mathcal{E}_G$ because the restriction maps are join-preserving. □

Recall, $\mathbf{B} = \{0 < 1\}$, the boolean lattice with two elements. The *boolean constant sheaf* over G is the sheaf $\underline{\mathcal{B}}$ with $\underline{\mathcal{B}}(i) = \mathbf{B}$ for all $i \in \mathcal{V}_G$, $\underline{\mathcal{B}}(ij) = \mathbf{B}$ for all $ij \in \mathcal{E}_G$, and $\underline{\mathcal{B}}(i \trianglelefteq ij) = \mathrm{id}_{\mathbf{M}}$.

Proposition 6.4. Suppose $\underline{\mathcal{F}}$ is a $\mathcal{S}up$-sheaf over a graph G. Then,

$$\Gamma\left(G; \underline{\mathcal{F}}\right) \cong \qquad \operatorname{Hom}\left(\underline{\mathcal{B}}, \underline{\mathcal{F}}\right).$$

Proof. A sheaf morphism φ is the data of a join-preserving map $\varphi_i : \mathbf{B} \to \underline{\mathcal{F}}(i)$ for every $i \in \mathcal{V}_G$ and a join-preserving map $\varphi_{ij} : \mathbf{B} \to \underline{\mathcal{F}}(ij)$ for every $ij \in \mathcal{E}_G$ satisfying (6.5). Therefore, φ_i "picks out" and $x_i \in \underline{\mathcal{F}}(i)$ for every node and φ_{ij} "picks out" a $x_{ij} \in \underline{\mathcal{F}}(ij)$ for every edge, such that $\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) = x_{ij}$. This is precisely the data of a global section. □

Thus, we have another proof that sections form a complete lattice. Another consequence is that the isomorphism in Proposition 6.4 implies a sheaf morphism $\underline{\mathcal{F}} \overset{\varphi}{\Rightarrow} \underline{\mathcal{G}}$ induces a join-preserving map on the lattice of global sections

$$\Gamma\left(G; \underline{\mathcal{F}}\right) \xrightarrow{\Gamma(G;\varphi)} \qquad \Gamma\left(G; \underline{\mathcal{G}}\right).$$

via precomposition.

Theorem 6.3. Suppose P is a poset. Then, $\Gamma(G; -)$ is a functor

$$\Gamma(G; -) : \mathcal{S}up^{\mathrm{P}} \to \mathcal{S}up.$$

6.5.2. Tensor product & sheaf hom

Definition 6.4 (Tensor Product Sheaf)**.** Suppose $\underline{\mathcal{F}}$ and $\underline{\mathcal{G}}$ are sheaves over a graph G. The *tensor product sheaf* $\underline{\mathcal{F}} \otimes \underline{\mathcal{G}}$ is the functor with

$$\left(\underline{\mathcal{F}} \otimes \underline{\mathcal{G}}\right)(i) = \qquad \underline{\mathcal{F}}(i) \otimes \underline{\mathcal{G}}(j)$$

$$\left(\underline{\mathcal{F}} \otimes \underline{\mathcal{G}}\right)(ij) = \qquad \underline{\mathcal{F}}(ij) \otimes \underline{\mathcal{G}}(ij)$$

for all $i \in \mathcal{V}_G$, $ij \in \mathcal{E}_G$ and restriction maps

$$\left(\underline{\mathcal{F}} \otimes \underline{\mathcal{G}}\right)(i) \xrightarrow{\left(\underline{\mathcal{F}} \otimes \underline{\mathcal{G}}\right)(i \trianglelefteq ij)} \left(\underline{\mathcal{F}} \otimes \underline{\mathcal{G}}\right)(ij)$$

sending $\underline{\mathcal{F}}(i) \xrightarrow{f} \underline{\mathcal{G}}(i)^{\mathrm{op}}$ to the composition

$$\begin{tikzcd}
\overline{\mathcal{F}}(ij) \arrow[r, "\overline{\mathcal{F}}(i \trianglelefteq ij)"] \arrow[rrrr, bend right, "(\underline{\mathcal{F}} \otimes \underline{\mathcal{G}})(i \trianglelefteq ij)(f)"'] & \overline{\mathcal{F}}(i) \arrow[r, equal] & \underline{\mathcal{F}}(i) \arrow[r, "f"] & \underline{\mathcal{G}}(i)^{\mathrm{op}} \arrow[r, "\underline{\mathcal{G}}(i \trianglelefteq ij)^{\mathrm{op}}"] & \underline{\mathcal{G}}(ij)^{\mathrm{op}}
\end{tikzcd}$$

6.5.3. Pullback & pushforward

Suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$. A *graph homomorphism* $p : G \to H$ is an strict order-preserving map

$$p : \mathrm{P}_G \to \mathrm{P}_H.$$

Suppose p is a graph homomorphism $p : G \to H$ and $\underline{\mathcal{F}}$ is a $\mathcal{S}up$-sheaf over H. The *pullback* of $\underline{\mathcal{F}}$ over p is the sheaf denoted $p^*\underline{\mathcal{F}}$ with

$$\begin{aligned}
p^*\underline{\mathcal{F}}(i) &= & \underline{\mathcal{F}}\left(p(i)\right) \\
p^*\underline{\mathcal{F}}(ij) &= & \underline{\mathcal{F}}\left(p(ij)\right), \\
p^*\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) &= & \underline{\mathcal{F}}\left(p(i) \trianglelefteq p(ij)\right).
\end{aligned}$$

The definition of the *pushforward* of $\underline{\mathcal{F}}$, denoted $p_*\underline{\mathcal{F}}$ is more involved. Stalks are

$$p_*\underline{\mathcal{F}}(i) = \qquad \lim_{k \trianglelefteq p(ij)} \underline{\mathcal{F}}(ij).$$

The cannonical example of a pushforward is the pushforward of the map $p : G \to *$ which is isomorphic to $\Gamma\left(G; \underline{\mathcal{F}}\right)$. A canonical example of a pullback is the pullback of an embedding $H \hookrightarrow G$ called the restriction of $\underline{\mathcal{F}}$ to H.

CHAPTER 7

The Tarski Laplacian

The graph Laplacian is an operator that acts on scalar fields while the connection Laplacian acts on vector fields. Their common denominator is the sheaf Laplacian [77]. In this chapter, we motivate a few candidate Laplacians for Tarski sheaves from several angles. First, from the vantage of lattice-valued consensus, then as an analogy to parallel transport, then finally as an homage to Hodge theory. Along the way, we present the Hodge-Tarski Fixed Point Theorem (Theorem 7.1) and provide algorithms to compute global sections in various settings.

7.1. Consensus

One practical motivation of the Tarski Laplacian is consensus.

Definition 7.1 (Multi-Agent Lattice Consensus)**.** Let $\mathbf{L}$ be a lattice. Suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a graph. A *consensus* function is a map $\chi : \mathbf{L}^{\mathcal{V}_G} \to \mathbf{L}$. A *consensus protocol* is a map $\chi : \mathbf{L}^{\mathcal{V}_G} \to \mathbf{L}$ (consensus function) together with an algorithm whose input is an arbitrary signal $\mathbf{x} : \mathcal{V}_G \to \mathbf{L}$ and whose (desired) output is $\chi(\mathbf{x})$.

Example 7.1 (Meet-Consensus)**.** Meet-consensus takes place in a sequence of rounds. Each node, every node its state $x_i \in \mathbf{L}, i \in \mathcal{V}_G$ to each of its neighbors $j \in \mathcal{N}_i$. At the conclusion of the round, a node i computes the meet of its received messages as well as (possibly) x_i. In a simple example (Figure 7.1), two nodes (left, right) share a communication link with a common node (center). At $t = 0$ (light), each node computes meets with neighbors. At $t = 2$ (dark), a consensus is reached. The updates are defined in a recursion equation

$$x_i[t+1] = \qquad x_i[t] \wedge \left(\bigwedge_{j \in \mathcal{N}_i} x_j[t] \right)$$

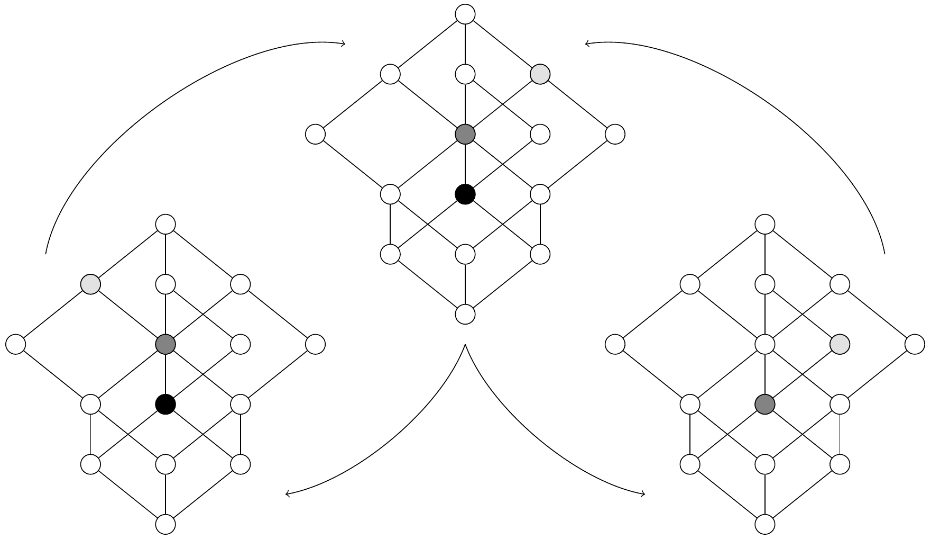


Figure 7.1: A meet-consensus algorithm.

Algorithm 1: Meet-consensus; in join-consensus, replace $\wedge$ with $\vee$.

```
Input: Graph $G = (\mathcal{V}_G, \mathcal{E}_G)$; $x_i \in \mathbf{L}, i \in \mathcal{V}_G$
Output: $x_i \in \mathbf{L}, i \in \mathcal{V}_G$
repeat
    for $i \in \mathcal{V}_G$ do in parallel
        for $j \in \mathcal{N}_i$ do
            $x_i \leftarrow x_i \wedge x_j$
        end
    end
until $x_i = x_j \ \forall \ ij \in \mathcal{E}_G$
```

Proposition 7.1. Suppose G is connected and $\mathbf{L}$ is a finite lattice. The meet-consensus algorithm (7.1) converges to $\left(\bigwedge_{j \in \mathcal{V}_G} x_j\right)_{i \in \mathcal{V}_G}$ in finite time.

Proof. After one round, x_i is updated with the meet of x_i with each of the x_j for each neighbor $j \in \mathcal{N}_i$. Let $\mathcal{N}_i^k$ denote the k-hop neighbors of i. It follows that

$$x_i[k] = \bigwedge_{j \in \mathcal{N}_i^k \cup i} x_j[0].$$

Because G is connected, for any $i \in \mathcal{V}_G$ it is guaranteed that $\mathcal{N}_i^k \cup i \supseteq \mathcal{V}_G$. Hence,

$$x_i[k] = \bigwedge_{j \in \mathcal{V}_G} x_j[0]$$

for $k \geqslant \operatorname{diam}(G)$. □

Consensus can easily be recast as a global section problem: consensus is the lattice of sections of the constant sheaf.

Proposition 7.2. Suppose $\underline{\mathbf{L}}$ is the constant sheaf over G. Then,

$$\Gamma\left(G; \underline{\mathbf{L}}\right) = \{(x_i)_{i \in \mathcal{V}_G} \mid x_i = x_j \ \forall \ ij \in \mathcal{E}_G\}.$$

Proof. The global section condition dictates that $\operatorname{id}(x_i) = \operatorname{id}(x_j)$ for all $ij \in \mathcal{E}_G$. □

The lattice structure of $\Gamma\left(G; \underline{\mathbf{L}}\right)$ is "the same" as $\mathbf{L}$ (under an isomorphism) but not equal.

Proposition 7.3. Suppose G is a graph, $\mathbf{L}$ is a lattice, and

$$S : C^0(G; \underline{\mathbf{L}}) \to C^0(G; \underline{\mathbf{L}}) \tag{7.1}$$

$$(S\mathbf{x})_i = \bigwedge_{j \in \mathcal{N}_j} x_j \tag{7.2}$$

Then,

$$\operatorname{Post}(S) = \Gamma\left(G; \underline{\mathbf{L}}\right).$$

Proof. One direction is trivial. By Proposition 7.2, if $\mathbf{x} \in C^0(G; \underline{\mathbf{L}})$ is locally constant, then $(S\mathbf{x})_i = x_i$, hence, $S\mathbf{x} \succcurlyeq \mathbf{x}$. Now suppose $\mathbf{x} \in \operatorname{Post}(S)$. Then, for all $i \in \mathcal{V}_G$, $x_i = x_i \wedge \bigwedge_{j \in \mathcal{N}_i} x_j$ which is equivalent to $x_i \preccurlyeq \bigwedge_{j \in \mathcal{N}_i} x_j$ for all $i \in \mathcal{V}_G$. Suppose $ij \in \mathcal{E}_G$ is a particular edge. Then, both

$$\begin{aligned} x_i \preccurlyeq & \bigwedge_{j \in \mathcal{N}_i} x_j \preccurlyeq & x_j, \\ x_j \preccurlyeq & \bigwedge_{i \in \mathcal{N}_j} x_i \preccurlyeq & x_i. \end{aligned}$$

Hence, $x_i = x_j$ for all $ij \in \mathcal{E}_G$, by the anti-symmetry axiom of posets, and $\mathbf{x}$ is a global section of the constant sheaf. □

Beyond meet-consensus (or, dually, join-consensus), are there other consensus algorithms? To answer this question, we first describe sheaves whose sections correspond to as approximations of the constant sheaf. The following definition is due to Hansen [73, Chapter 7].

Definition 7.2 (Approximation)**.** Let G be a graph and $\underline{\mathcal{G}}$ and $\underline{\mathcal{F}}$ be Tarski sheaves over G. Then, $\underline{\mathcal{F}}$ is an approximation to $\underline{\mathcal{G}}$ if there is there is a sheaf morphism $\underline{\mathcal{F}} \overset{\varphi}{\Longrightarrow} \underline{\mathcal{G}}$ such that $\underline{\mathcal{F}}(i) \cong \underline{\mathcal{F}}(j)$ for all $i \in \mathcal{V}_G$ and such that the induced join-preserving map

$$\Gamma\left(G; \underline{\mathcal{F}}\right) \to \qquad \Gamma\left(G; \underline{\mathcal{G}}\right)$$

is an isomorphism.

An approximation of the constant sheaf $\underline{\mathbf{L}}$ consists of a sheaf whose vertex stalks are $\mathbf{L}$ and whose lattice of sections is the same as $\Gamma\left(G; \underline{\mathbf{L}}\right)$, isomorphic to $\mathbf{L}$ if G is connected. In other words, approximations to the constant sheaf also model consensus. An approximation of $\underline{\mathbf{L}}$ is a sheaf $\underline{\mathcal{F}}$ characterized by the following diagram

$$\begin{tikzcd} \underline{\mathcal{F}}(i) \arrow[r, "\underline{\mathcal{F}}(i \trianglelefteq ij)"] \arrow[d, equal] & \underline{\mathcal{F}}(ij) \arrow[d, "\varphi_{ij}"] & \underline{\mathcal{F}}(j) \arrow[l, "\underline{\mathcal{F}}(j \trianglelefteq ij)"'] \arrow[d, equal] \\ \mathbf{L} \arrow[r, "\text{id}"] & \mathbf{L} & \mathbf{L} \arrow[l, "\text{id}"'] \end{tikzcd}$$

.

There are more questions about lattice consensus than answers. If the consensus operator S is a Laplacian for the constant sheaf, what is the "sheaf Laplacian" of an approximation to the constant sheaf? or an arbitrary Tarski sheaf? Furthermore, if $\underline{\mathcal{F}}$ approximates $\underline{\mathbf{L}}$ and $\chi : C^0(\underline{\mathcal{F}}; G) \to \mathbf{L}$ is a consensus function, does there exist "sheaf dynamics" converging to the $\chi(\mathbf{x}[0])$? There is also a utility for consensus "in-between" meet- and join- consensus. In some applications, for instance, earliest rendezvous [122], min-consensus is desired. On the other hand, consider the task of forming teams amongst a set of agents. Cast as a consensus problem on the lattice of subpartitions, it would not be sensible to achieve either the finest or the coarsest organization possible.

Example 7.2 (Causal Inference)**.** Suppose a set of random variables is collected in a partial order P. For instance, random variables could indicate (transitive, reflexive, anti-symmetric) causal relationships between features. From join-consensus on the lattice of downsets $\mathcal{D}(\mathrm{P})$, a subset of P can be recovered from the poset of join-irreducibles of $\mathcal{D}(\mathrm{P})$ (isomorphic to P, Theorem 2.3), thus defining a weak consensus function $\chi : \mathrm{P}^n \to \wp(\mathrm{P})$.

Example 7.3 (Distributed Optimization)**.** A multitude of agent-based tasks are formulated or equivalent to distributed optimization [47]. Suppose $\{1, 2, \ldots, n\}$ is a group of agents with objective function $f_i,\ i \in \{1, 2, \ldots, n\}$. In the simplest setting, *distributed optimzation* is unconstrained optimization problem

$$\min_{\mathbf{x}} \sum_{i=1}^{n} f_i(\mathbf{x}) \tag{7.3}$$

Consensus and distributed optimization are intertwined through the observation that (7.4) is equivalent to

$$\begin{gathered} \min_{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n} \sum_{i=1}^{n} f_i(\mathbf{x}_i) \\ \text{subject to} \\ \mathbf{x}_i = \mathbf{x}_j \quad \forall i, j \in \{0, 1, \ldots, n\} \end{gathered} \tag{7.4}$$

Distributed optimization, thus, reduces to consensus and optimization of the local objective functions $f_i, i \in \{1, 2, \ldots, n\}$.

Recent work generalizes distributed optimization to network sheaves over G with euclidean stalks [75]:

$$\begin{gathered} \min_{\mathbf{x} \in C^0(G; \underline{\mathcal{F}})} \sum_{i \in \mathcal{V}_G} f_i(\mathbf{x}_i) \\ \text{subject to} \\ \mathbf{x} \in H^0(G; \underline{\mathcal{F}}). \end{gathered} \tag{7.5}$$

where each f_i is a convex function $f_i : \underline{\mathcal{F}}(i) \to \mathbb{R}$.

Optimization algorithms maximizing [119] or minimizing [159] real-valued functions on lattices are restricted to certain classes of functions (e.g. submodular, supermodular, increasing difference). An algorithm solving (7.5) for $\underline{\mathcal{F}}$, a Tarski sheaf, is an open problem whose solution, we speculate, will involve clever modulations between the tasks of running lattice consensus and performing "gradient" updates of the local objective functions $f_i, i \in \mathcal{V}_V$.

7.2. Parallel Transport

Schreiber and Waldorf [147] and others [95] have argued parallel transport on a manifold $\mathbb{M}$ valued in an arbitrary category $\mathcal{D}$ is modeled by a functor

$$P : \mathrm{Path}(\mathbb{M}) \to \mathcal{D}.$$

from the path groupoid of $\mathbb{M}$. Path($\mathbb{M}$) is a category whose objects are points from $\mathbb{M}$ and whose morphisms are, roughly, homotopy classes of paths between points [80]. In the sequel, given a graph G and a Tarski sheaf $\mathcal{F}$ on G, we construct a functor

$$P^{\mathcal{F}} : \mathrm{Free}(G) \to \mathcal{P}os$$

sending a path $\gamma : i \to j$ to a order-preserving map $P^{\mathcal{F}}_{\gamma} : \mathcal{F}(i) \to \mathcal{F}(j)$ simulating parallel transport. In our view, parallel transport models message-passing of local sections of agents connected by a path in a communication network.

Definition 7.3 (Parallel Transport)**.** Suppose G is a graph and $\mathcal{F}$ is a Tarski sheaf over G. Suppose $j \in \mathcal{N}_i$. Then, the *parallel transport* along the path $i \to j$ is the map

$$\begin{aligned} P^{\mathcal{F}}_{i \to j} : \qquad & \mathcal{F}(i) \to \mathcal{F}(j) \\ P^{\mathcal{F}}_{i \to j}(x_i) = \qquad & \overline{\mathcal{F}}(j \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i). \end{aligned}$$

If γ is an arbitrary path from i to j, then $P^{\mathcal{F}}_{\gamma}$ is defined as the (unique) composition of parallel transport maps along paths of length one. The parallel transport of the trivial path ε_i based at $i \in \mathcal{V}_G$ is defined to be the identity map on $\mathcal{F}(i)$.

Parallel transport is functorial by construction (i.e. $P^{\mathcal{F}}_{\gamma_1 \cdot \gamma_2} = P^{\mathcal{F}}_{\gamma_2} \circ P^{\mathcal{F}}_{\gamma_1}$). Furthermore, parallel transport is a functor into the catgory of posets and order-preserving maps because $P^{\mathcal{F}}_{i \to j}$, as a composition of an join- and meet- preserving map, is not (in general) join- or meet-preserving. If γ and γ' are two distinct paths from i to j, in is not the case (in general) that

$$P^{\mathcal{F}}_{\gamma}(x_i) = P^{\mathcal{F}}_{\gamma}(x_j) \quad \forall i, j \in \mathcal{V}_G.$$

Whenever the above holds, $\mathcal{F}$ is said to be flat.

7.2.1. Holonomy

Having defined parallel transport of Tarski sheaves, an abundance of potential geometric construction wait at the door including connections, curvature, characteristic classes, and more. Immediately, we are in a position to define holonomy, a construction which, roughly speaking, measures the degree to which parallel transport around a closed loop fails to preserve orientations of tangent vectors.

Definition 7.4 (Holonomy)**.** Suppose G is a graph. Let $\Omega_i(G)$ denote the loops in G based at at particular $i \in \mathcal{V}_G$. Then, the *holonomy* of $\mathcal{F}$ is the ordered monoid

$$\mathrm{Hol}_i(G; \underline{\mathcal{F}}) = \qquad \{P_\gamma^{\mathcal{F}} \mid \gamma \in \Omega_i(G)\}$$

Holonomy is a monoid because the composition of loops yields a composition of transport maps and the trivial path ε_i yields an identity. It is a poset because $\mathrm{Hol}_i(G; \underline{\mathcal{F}})$ inherits the order of the poset of order-preserving maps $\mathrm{End}_{\mathcal{P}os}(\underline{\mathcal{F}}(i))$.

7.2.2. Global sections

Parallel transport characterizes sections. We first show that sections facilitate the transport of "tangent vectors" (lattice elements) to compatible "tangent vectors" (lattice elements) along any (network) path.

Lemma 7.1 (Section Path Lemma)**.** Suppose $\mathcal{F}$ is a Tarski sheaf over G and $\mathbf{x} \in \Gamma\left(G; \underline{\mathcal{F}}\right)$ is a section. Then, for every path γ from i to j,

$$P_\gamma^{\mathcal{F}}(x_i) \succcurlyeq \qquad x_j.$$

Proof. We proceed by induction on path length:

Base Case. Suppose γ is the path $i \to j$ (of length one. Then, as $\mathbf{x}$ is a section,

$$\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \succcurlyeq \qquad \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j)$$

implies

$$\overline{\mathcal{F}}(j \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \succcurlyeq \qquad x_j.$$

Inductive Step. Suppose γ is a path of length ℓ. Suppose, without loss of generality, that γ is a path from i to j, and γ' is a path from the same i to j'. By the inductive hypothesis,

$P^{\mathcal{F}}_{\gamma}(x_i) \succcurlyeq x_j$. Then,

$$\begin{aligned} P^{\mathcal{F}}_{\gamma'}(x_i) \succcurlyeq & \bigwedge_{j' \in \mathcal{N}_j} P^{\mathcal{F}}_{j \to j'} P^{\mathcal{F}}_{\gamma}(x_i) \\ \succcurlyeq & \bigwedge_{j' \in \mathcal{N}_j} P^{\mathcal{F}}_{j \to j'}(x_j) \\ \succcurlyeq & \; x_{j'}. \end{aligned}$$

□

For path of length one, *neighborhood paths*, Lemma 7.1 has a partial converse.

Lemma 7.2 (Neighborhood Path Lemma)**.** Suppose $\mathcal{F}$ is a Tarski sheaf over G, and supoose

$$P^{\mathcal{F}}_{i \to j}(x_i) \succcurlyeq \quad x_j, \quad i \in \mathcal{V}_G, j \in \mathcal{N}_i$$

for all $\mathbf{x} \in C^0(G; \underline{\mathcal{F}})$. Then, $\mathbf{x} \in \Gamma\,(G; \underline{\mathcal{F}})$.

Proof. By assumption, both of the following hold for an $\mathbf{x} \in C^0(G; \underline{\mathcal{F}})$ over all $ij \in \mathcal{E}_G$

$$\begin{aligned} \overline{\mathcal{F}}(j \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \succcurlyeq & \quad x_j, \\ \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(j \trianglelefteq ij)(x_j) \succcurlyeq & \quad x_i. \end{aligned}$$

Therefore, because by the property of Galois connection,

$$\begin{aligned} \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j) \succcurlyeq & \quad \underline{\mathcal{F}}(i \trianglelefteq ij)(x_i), \\ \underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \succcurlyeq & \quad \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j), \end{aligned}$$

which implies equality. □

Remark 7.1 (Proof of Lemma 7.2)**.** The argument in the proof of the Neighborhood Lemma breaks downs when you try to generalize it to arbitrary paths. For instance, if $\gamma = (i, j, k)$

is a path between distinct nodes, then

$$P_\gamma^{\mathcal{F}}(x_i) = \quad \overline{\mathcal{F}}(k \trianglelefteq jk)\underline{\mathcal{F}}(j \trianglelefteq jk)\overline{\mathcal{F}}(j \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \curlyeqsucc \quad x_k$$

does not imply that

$$P_\gamma^{\mathcal{F}}(x_k) = \quad \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(j \trianglelefteq ij)\overline{\mathcal{F}}(j \trianglelefteq jk)\underline{\mathcal{F}}(k \trianglelefteq jk)(x_k) \curlyeqsucc \quad x_i,$$

only that

$$\underline{\mathcal{F}}(j \trianglelefteq jk)\overline{\mathcal{F}}(j \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \curlyeqsucc \quad \underline{\mathcal{F}}(k \trianglelefteq jk)(x_k).$$

7.3. The Hodge-Tarski Theorem

Finally, in this section, we define a sheaf Laplacian for Tarski sheaves culminating in a Hodge-style fixed point theorem computing global sections.

7.3.1. The Tarski Laplacian

Definition 7.5. Suppose $\mathcal{F}$ is a Tarski sheaf over G. The *Tarski Laplacian* is an operator defined

$$\begin{aligned} L : &\quad C^0(G;\underline{\mathcal{F}}) \longrightarrow C^0(G;\underline{\mathcal{F}}) \\ (L\mathbf{x})_i = &\quad \bigwedge_{j \in \mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(j \trianglelefteq ij)(x_j). \end{aligned}$$

Recall, the *meet projection* of a lattice $\mathbf{L}$ onto a subset $N \subseteq \{1, 2, \ldots, n\}$ is the map

$$\begin{aligned} \chi : &\quad \mathbf{L}^n \to \mathbf{L} \\ \chi(\mathbf{x}) = &\quad \bigwedge_{i \in N} x_j \end{aligned}$$

Then, we note immediately, the Tarski Laplacian is a composition of two operations: (1)

parallel transport, and (2) meet-projection. Therefore,

$$(L\mathbf{x})_i = \bigwedge_{j \in \mathcal{N}_i} P^{\mathcal{F}}_{j \to i}(x_j). \tag{7.6}$$

Also, recall, the conditions of the Tarski Fixed Point Theorem 2.2.

Lemma 7.3. The Tarski Laplacian is an order-preserving map on a complete lattice.

Proof. By a similar argument that lattice polynomials are order-preserving [18], L is a composition of order-preserving function. $C^0(G; \underline{\mathcal{F}})$ is the product of complete lattices, hence, complete. □

7.3.2. Diffusion

For the following proposition, let $\mathcal{N}_i^k$ denote the *k-hop neighborhood* of $i \in \mathcal{V}_G$.[13] Suppose $W \subseteq \mathcal{V}_G$ is a subset of nodes. Then, π_W is the projection of $C^0(G; \underline{\mathcal{F}})$ onto W.

Proposition 7.4 (Locality)**.** Suppose $\mathcal{F}$ is a Tarski sheaf over G with Tarski Laplacian L. Suppose $i \in \mathcal{V}_G$. Then, the Tarski Laplacin is restricted to the complement of the k-hope neighborhood

$$L^k_{\mathcal{V}_G \setminus \mathcal{N}_i^k} : \qquad \prod_{j \in \mathcal{V}_G \setminus \mathcal{N}_i^k} \underline{\mathcal{F}}(j) \to \prod_{j \in \mathcal{V}_G \setminus \mathcal{N}_i^k} \underline{\mathcal{F}}(j)$$

is the identity.

As an operator, the Tarski Laplacian is said to be *local* in the sense that $(L^k\mathbf{x})_i$ only depends only on the assignment to the nodes $j \in \mathcal{N}_i^k$. Familiar examples of other local operators include the graph Laplacian, graph connection Laplacian, and the linear sheaf Laplacian.

[13] Let $\mathcal{N}_i^0 = \{i\}$.

7.3.3. Similarities to the connection Laplacian

Recall, in the connection Laplacian,

$$(L\mathbf{x})_i = \frac{1}{d_i}\sum_{j\in\mathcal{N}_i} w_{ij}O_{ij}x_j,$$

tangent vectors of neighbors $\{\mathbf{x}_j\}_{j\in\mathcal{N}_i}$ are transformed by orthogonal maps O_{ij} approximating parallel transport between $T_{x_i}\mathbb{M}$ and $T_{x_i}\mathbb{M}$. Next, node messages $\{O_{ij}\mathbf{x}_j\}_{j\in\mathcal{N}_i}$ are received and aggregated with via a weighted sum.

In the Tarski Laplacian (7.6), lattice elements of neighbors $\{x_j\}_{j\in\mathcal{N}_i}$ are transformed by order-preserving maps $P^{\mathcal{F}}_{j\to i}$ subordinate to $\mathcal{F}$ that model parallel transport . Next, node messages $\{P^{\mathcal{F}}_{j\to i}(x_j)\}_{j\in\mathcal{N}_i}$ are received and aggregated with a meet-projection onto $\mathcal{N}_j$.

7.3.4. A fixed point theorem

Recall, the Hodge Theorem (Theorem 5.1) characterizes the fixed points of heat equation (5.5). We now state a culminating result characterising the suffix points of the Tarski Laplacian with the sections $\Gamma\left(G;\underline{\mathcal{F}}\right)$ of a Tarski sheaf $\underline{\mathcal{F}}$ over G.

Theorem 7.1 (Tarski-Hodge Theorem)**.** Suppose $\mathcal{F}$ is a Tarski sheaf over a graph G with Tarski Laplacian $L : C^0(G;\underline{\mathcal{F}}) \to C^0(G;\underline{\mathcal{F}})$. Then,

$$\mathrm{Post}(L) = H^0(G;\underline{\mathcal{F}}). \tag{7.7}$$

The proof relies on the theory of parallel transport from Section 7.2.

Proof of Tarski-Hodge Theorem. By the Section Path Lemma (Lemma 7.1) and the Neighborhood Lemma (Lemma 7.2),

$$\mathbf{x} \in \Gamma\left(G;\underline{\mathcal{F}}\right)$$

if and only if

$$P^{\mathcal{F}}_{j\to i}(x_j) \succcurlyeq \qquad x_i$$

holds for all $i \in \mathcal{V}_G$, $j \in \mathcal{N}_i$. By the universal property of meets (Lemma 3.3),

$$P^{\mathcal{F}}_{j\to i}(x_j) \succcurlyeq \qquad x_i$$

for all $j \in \mathcal{N}_i$ if and only if

$$\bigwedge_{j\in\mathcal{N}_i} P^{\mathcal{F}}_{j\to i}(x_j) \succcurlyeq \qquad x_i.$$

Therefore, $(L\mathbf{x})_i \succcurlyeq x_i$ for every $i \in \mathcal{V}_V$ precisely if $\mathbf{x} \in H^0(G; \underline{\mathcal{F}})$. □

Equivalently, we may rewrite the above as the fixed points of the operator $L \wedge \mathrm{id}$ because $L(\mathbf{x}) \succcurlyeq \mathbf{x}$ if and only if $L(\mathbf{x}) \wedge \mathbf{x}$.

Corollary 7.1. $\mathrm{Fix}(L \wedge \mathrm{id}) = H^0(G; \underline{\mathcal{F}})$.

Remark 7.2. Indeed, the Hodge-Tarski Theorem (Theorem 7.1) is a "Hodge-style fied point theorem." The Euler discretization of the heat equation (5.5) (step-size $\alpha > 0$) is the following

$$\mathbf{x}[t+1] - \mathbf{x}[t] = \qquad -\alpha\Delta_k\mathbf{x} \tag{7.8}$$

It follows

$$\mathrm{Fix}(\mathrm{id} - \alpha\Delta) \cong \qquad H^k(C^\bullet)$$

by the isomorphism between Ker (Δ_K) and $H^0(C^\bullet)$ in the Hodge Theorem (Theorem 5.1).

Then, compare (7.8) to

$$\mathrm{Fix}(\mathrm{id} \wedge L) = \qquad\qquad H^k(G; \underline{\mathcal{F}}). \tag{7.9}$$

A second (also obvious) corollary is a consequence of the isomorphism between global sections and H^0 (Proposition 5.1).

Corollary 7.2. $\mathrm{Fix}(L \wedge \mathrm{id}) \cong \Gamma\,(G; \underline{\mathcal{F}})$.

A final corollary offers a third an alternate proof to Theorem 6.2 that the global sections of a Tarski sheaf are a complete quasi-sublattice of $\Gamma\,(G; \underline{\mathcal{F}})$.

Corollary 7.3. $H^0(G; \underline{\mathcal{F}})$ is a complete lattice. Furthermore, there is an order-embedding of $\Gamma\,(G; \underline{\mathcal{F}})$ into the complete lattice $C^0(G; \underline{\mathcal{F}})$ of 0-cochains.

Proof. The Tarski Laplacian is an order-preserving map on a complete lattice $C^0(G; \underline{\mathcal{F}})$ (Proposition 7.3). Therefore, by Corollary 7.3 and the Tarski Fixed Point (Theorem 2.2), the isomorphism between $\Gamma\,(G; \underline{\mathcal{F}})$ and $H^0(G; \underline{\mathcal{F}})$ followed by the inclusion of $H^0(G; \underline{\mathcal{F}})$ into $C^0(G; \underline{\mathcal{F}})$ is an order-embedding. □

7.4. Towards Operator Theory

By Corollary 7.3, global sections of Tarski sheaves have arbitrary meets and joins, forming a complete lattice. In the context of multi-agent systems, this implies the set of consistent assignments of lattice-valued data over a network of agents forms a lattice and results in two merging operations (meet/join) between consistent assignments, sections. In this section, we introduce two new operators: the closure and the Helmholtzian. We discuss a strategy for computing the explicit lattice structure of $H^0(G; \underline{\mathcal{F}})$ of a Tarski sheaf.

7.4.1. A closure operator

We introduce another local operator on $C^0(G; \underline{\mathcal{F}})$ that is *not* a Laplacian.

Definition 7.6 (Closure)**.** The *closure* is the endomorphism

$$\begin{aligned} E: \quad & C^0(G;\underline{\mathcal{F}}) \to C^0(G;\underline{\mathcal{F}}) && (7.10) \\ (E\mathbf{x})_i = \quad & \bigwedge_{j\in\mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i). && (7.11) \end{aligned}$$

Recall, if $(f_\bullet, f^\bullet)$ is a Galois connection, then $f^\bullet f_\bullet \succcurlyeq \mathrm{id}$ (Proposition 3.1). The closure, then, "intensifies" local data. To be precise, E is a closure operator on the complete lattice $C^0(G;\underline{\mathcal{F}})$.

Theorem 7.2 (Closure Theorem)**.** The operator E is a closure operator on $C^0(G;\underline{\mathcal{F}})$.

Proof. It is trivial to check E is order-preserving. By Proposition 3.1 and the universal property of meets (Lemma 3.3), $(E\mathbf{x})_i \succcurlyeq x_i$ for all $i \in \mathcal{V}_G$. Hence, E is inflationary. To show E is idempotent, we show $E^2(\mathbf{x}) \preccurlyeq \mathbf{x}$ and $E^2(\mathbf{x}) \succcurlyeq \mathbf{x}$. For the first inequality,

$$\begin{aligned} (E^2\mathbf{x})_i = \quad & \bigwedge_{j\in\mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)\Big(\bigwedge_{j\in\mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i)\Big) \\ \preccurlyeq \quad & \bigwedge_{j\in\mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)\overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \\ = \quad & \bigwedge_{j\in\mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \\ = \quad & (E\mathbf{x})_i, \end{aligned}$$

where the penultimate equality follows from Lemma 3.2. For the second inequality,

$$\begin{aligned} (E^2\mathbf{x})_i = \quad & \bigwedge_{j\in\mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij) \bigwedge_{j\in\mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \\ \succcurlyeq \quad & \bigwedge_{j\in\mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij)\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \end{aligned}$$

by $E\mathbf{x} \succcurlyeq \mathbf{x}$ and monotonicity. Hence, $E^2 = E$. □

The fixed points of the closure operator E is a quasi-sublattice of $C^0(G; \underline{\mathcal{F}})$.

Corollary 7.4. $\mathrm{Fix}(E)$ is a complete lattice.

Proof. Apply Theorem 2.1. □

Moreover, we can explicitly compute meets and joins with

$$\begin{aligned} \mathbf{x} \wedge_{\mathrm{Fix}(E)} \mathbf{y} &= \mathbf{x} \wedge \mathbf{y}, \\ \mathbf{x} \vee_{\mathrm{Fix}(E)} \mathbf{y} &= E(\mathbf{x} \vee \mathbf{y}). \end{aligned}$$

The following establishes a connection between the closure and the Tarski Laplacian.

Lemma 7.4. Suppose $\mathbf{x} \in H^0(G; \underline{\mathcal{F}})$. Then, $L(\mathbf{x}) = E(\mathbf{x})$.

Proof. Let $\mathbf{x} \in H^0(G; \underline{\mathcal{F}})$. Then, for all $i \in \mathcal{V}_G$,

$$\begin{aligned} (L\mathbf{x})_i &= \bigwedge_{j \in \mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij) \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j) \\ &= \bigwedge_{j \in \mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij) \underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \end{aligned}$$

as $\underline{\mathcal{F}}(j \trianglelefteq ij)(x_j) = \underline{\mathcal{F}}(i \trianglelefteq ij)(x_i)$ for every $ij \in \mathcal{E}_G$. □

7.4.2. Meets & joins of sections

Lemma 7.4 suggests that one may attempt to restrict L to $H^0(G; \underline{\mathcal{F}})$ to obtain a closure operator. The following shows that $H^0(G; \underline{\mathcal{F}})$ is invariant under L. A similar result was shown by Cunninghame-Green & Butkovic [41] in a highly-specialized case (Figure 1.4) of max-plus restriction maps over the network with two nodes and one edge.

Lemma 7.5. Suppose $\mathbf{x} \in H^0(G; \underline{\mathcal{F}})$. Then, $L(\mathbf{x}) \in H^0(G; \underline{\mathcal{F}})$.

Proof. By the Hodge-Tarski Theorem (Theorem 7.1), it suffices to show that $L(\mathbf{x}) \succcurlyeq \mathbf{x}$ which follows by monotonicity of L and transitivity. □

Warning 7.1. It would be tempting to conclude that $L|_{H^0(G;\underline{\mathcal{F}})} : H^0(G;\underline{\mathcal{F}}) \to H^0(G;\underline{\mathcal{F}})$ is a closure operator, and, then, one may compute meets and joints in $H^0(G;\underline{\mathcal{F}})$ explicitly by Theorem 2.1. However, this argument is invalid because, in the definition of the Tarski Laplacian, the product lattice structure on $C^0(G;\underline{\mathcal{F}})$ does not, in general, coincide with the lattice structure on $H^0(G;\underline{\mathcal{F}})$. Hence, this argument is circular.

7.4.3. The Tarski Helmholtzian

The Tarski Laplacian acts on $C^0(G;\underline{\mathcal{F}})$. We define another operator, analogous to the graph Helmholtzian [104] which acts on $C^1(G;\underline{\mathcal{F}})$.

Definition 7.7 (Helmholtzian)**.** Suppose $\mathcal{F}$ is a Tarski sheaf over G. The *Tarski Helmholtzian* is the operator $\mathfrak{L} : C^1(G;\overline{\mathcal{F}}) \to C^1(G;\overline{\mathcal{F}})$ defined

$$(\mathfrak{L}\mathbf{x})_{ij} = \left(\bigvee_{j'\in\mathcal{N}_i - j} \underline{\mathcal{F}}(i \trianglelefteq ij)\overline{\mathcal{F}}(i \trianglelefteq ij')(x_{ij'})\right) \vee \left(\bigvee_{i'\in\mathcal{N}_j - i} \underline{\mathcal{F}}(j \trianglelefteq ij)\overline{\mathcal{F}}(j \trianglelefteq i'j)(x_{i'j})\right)$$

We conjecture that the lattice $H^1(G;\underline{\mathcal{F}})$ constructed as a coequalizer (5.2) coincides with the prefix points of the Tarski Helmholtzian.

Conjecture 7.1. Suppose $\mathcal{F}$ is a Tarski sheaf over G with Tarski-Helmholtz operator $\mathfrak{L}$: $C_1(G;\underline{\mathcal{F}}) \to C_1(G;\underline{\mathcal{F}})$. Then,

$$\mathrm{Pre}(\mathfrak{L}) = \qquad H^1(G;\underline{\mathcal{F}}). \tag{7.12}$$

7.5. Algorithms

We end this chapter with a discussion of two different algorithms that are inspired by the Tarski Laplacian.

7.5.1. Heat Flow

The Tarski Laplacian inspires a discrete-time heat equation on $C^0(G;\underline{\mathcal{F}})$:

$$\mathbf{x}[t+1] = (L \wedge \mathrm{id})\mathbf{x}[t] \tag{7.13}$$

with initial condition $\mathbf{x}[0] \in C^0(G;\underline{\mathcal{F}})$. Locally, these updates can be cast as a distributed algorithm (Algorithm 2). In order to guarantee the algorithm terminates, we require an assumption.

Assumption 7.1. (DCC) The product lattice $C^0(\underline{\mathcal{F}};G)$ satisfies the descending chain condition (Definition 2.3).

Algorithm 2: Heat Flow

```
Input: Bisheaf $\mathcal{F}$ over graph $G = (\mathcal{V}_G, \mathcal{E}_G)$; initial $x_i \in \underline{\mathcal{F}}(i), i \in \mathcal{V}_G$
Output: $x_i \in \underline{\mathcal{F}}(i), i \in \mathcal{V}_G$
repeat
    for $i \in \mathcal{V}_G$ do in parallel
        for $j \in \mathcal{N}_i$ do
            $x_i \leftarrow x_i \wedge \overline{\mathcal{F}}(i \trianglelefteqslant ij)\underline{\mathcal{F}}(j \trianglelefteqslant ij)(x_j)$
        end
    end
until $\underline{\mathcal{F}}(i \trianglelefteqslant ij)(x_i) = \underline{\mathcal{F}}(j \trianglelefteqslant ij)(x_j)\ \forall\ ij \in \mathcal{E}_G$
```

Proposition 7.5 (Completeness)**.** Suppose $\mathcal{F}$ is a Tarski sheaf over G satisfying Assumption 7.1. Then, Algorithm 2 converges in finitely many iterations.

Proof. $L \wedge \mathrm{id} \preccurlyeq \mathrm{id}$ (trivially) and the recursion $\mathbf{x}[t] = (L \wedge \mathrm{id})^t \mathbf{x}[0]$ imply $\mathbf{x}[s] \preccurlyeq \mathbf{x}[t]$ whenever $s \leqslant t$. Hence, $\{\mathbf{x}[t]\}_{t \geqslant 0}$ forms a descending chain in $C^0(G;\underline{\mathcal{F}})$. □

Corollary 7.5. Suppose $\mathcal{F}$ is a Tarski sheaf over G satisfying Assumption 7.1, then Algorithm 2 converges to some $\mathbf{x} \in H^0(G;\underline{\mathcal{F}})$.

Proof. By Proposition 7.5, the Algorithm 2 converges. Hence, there is a t_0 such that $(L \wedge \mathrm{id})^t \mathbf{x}[t] = (L \wedge \mathrm{id})^t \mathbf{x}[t_0]$ for all $t \geqslant t_0$. Therefore, by the Tarski-Hodge Theorem, $(L \wedge \mathrm{id})^t \mathbf{x}[t_0] \in H^0(G; \underline{\mathcal{F}})$. □

If a poset is graded, then it satisfies the descending chain condition.

Corollary 7.6. Suppose $C^0(G; \underline{\mathcal{F}})$ is graded. Then, Algorithm 2 also terminates in finite time and converges to a section.

The heat flow alogithm (Algorithm 2) is depicted visually (Figure 7.2).

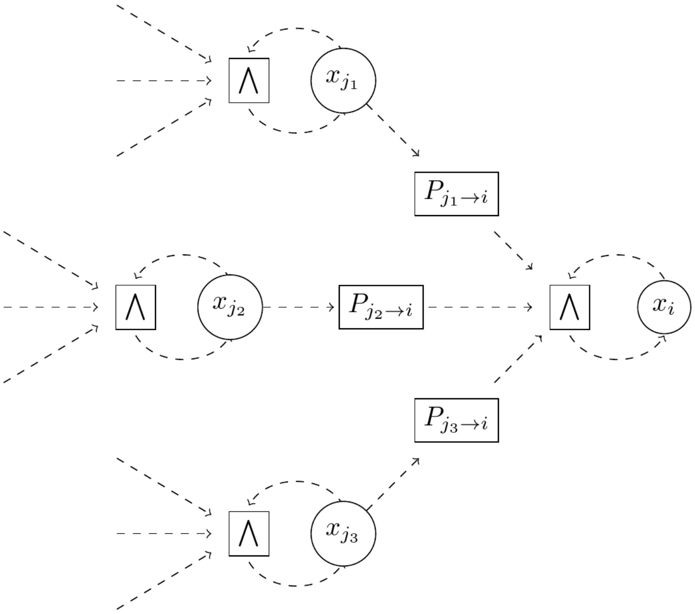


Figure 7.2: Heat flow on a Tarski Sheaf.

7.5.2. Gossip

Gosip algorithms are a general class of decentralized algorithms that are broadly characterized as having each exchange information with one or a few neighboring nodes each round. The advantages of gossip over Laplacian-based methods include fault tolerance and a relaxation of assumption that nodes update their states synchronously [90]. In order to describe an asynchronous version of the Heat Flow algorithm, we will need some notation. Let $\tau : \mathbb{N} \to \wp(\mathcal{V}_G)$ be a *broadcast sequence.* At time t, the nodes $\tau(t) \subseteq \mathcal{V}_G$ broadcast to all neighbors $j \in \mathcal{N}_i$. This is quite general. For instance, if $\tau(t) = \mathcal{V}_G$, then we recover Algorithm 2. However, if $\tau(t)$ is a single node, then we have a truly asynchronous algorithm. Additionally, we require nodes to continually exchange information.

Assumption 7.2. (Liveness) For every $i \in \mathcal{V}_G$ and $t_0 \in \mathbb{N}$, there exist $t \geqslant t_0$ such that $i \in \tau(t)$.

The Heat Flow algorithm (Algorithm 2) is modified (Algorithm 3) by aggregating parallel transported lattice-valued information from only the nodes that are broadcasting at a given time.

Algorithm 3: Gossip

```
Input: Bisheaf $\mathcal{F}$ over graph $G = (\mathcal{V}_G, \mathcal{E}_G)$; initial $x_i \in \mathcal{F}(i), i \in \mathcal{V}_G$; broadcast
       sequence $\tau : \mathbb{N} \to \wp(\mathcal{V}_G)$.
for $t \in \mathbb{N}$ do
    for $i \in \tau(t)$ do
        for $j \in \mathcal{N}_i$ do
            sendTo($j$) $\leftarrow \underline{\mathcal{F}}(i \trianglelefteq ij)(x_i)$
            receiveFrom($i$) $\leftarrow$ sendTo($j$)
            $x_j \leftarrow x_j \wedge \overline{\mathcal{F}}(j \trianglelefteq ij)\,($receiveFrom($i$)$)$
        end
    end
end
```

Alternatively, we may rewrite the algorithm as a recursion equation

$$\mathbf{x}[t+1] = L_t\big(\mathbf{x}[t]\big) \wedge \mathbf{x}[t] \tag{7.14}$$

called the *gossip equation* with the time-varying Tarski Laplacian L_t defined as follows.

Definition 7.8 (Time-Varying Tarski Laplacian)**.** Suppose G is a graph and $\mathcal{F}$ is a Tarski sheaf over G and $\tau : \mathbb{N} \to \wp(\mathcal{V}_G)$ satisfies Assumption 7.2. The *asynchronous Tarski*

Laplacian:[14] is the time-varying operator

$$
\begin{aligned}
\tilde{L}_t : \quad & \mathbb{N} \times C^0(G; \underline{\mathcal{F}}) \to C^0(G; \underline{\mathcal{F}}) \\
\left(\tilde{L}_t\right)_i = \quad & \bigwedge_{j \in \tau(t) \cap \mathcal{N}_i} \overline{\mathcal{F}}(i \trianglelefteq ij) \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j).
\end{aligned}
\tag{7.15}
$$

We now state a equilibrium result for global sections for gossip.

Theorem 7.3 (Theorem 1 [134])**.** Suppose G is a graph and $\mathcal{F}$ is a Tarski sheaf over G. Suppose $\tau : \mathbb{N} \to \wp(\mathcal{V}_G)$ is a firing sequence satisfying Assumption 7.1. Suppose $\tilde{L}_t$ is the asynchronous Tarski Laplacian. Then, if τ is a broadcast sequence satisfying Assumption 7.2, the sections $H^0(G; \underline{\mathcal{F}})$ coincide with the time-independent solutions of the recursion equations

$$
\mathbf{x}[t+1] = \quad (L_t \wedge \mathrm{id})\mathbf{x}[t] \tag{7.16}
$$

with initial condition $\mathbf{x}[0] \in C^0(G; \underline{\mathcal{F}})$.

Corollary 7.7. Suppose G, $\mathcal{F}$ and τ satisfy Assumptions 7.1 and 7.2. Then, Algorithm 3 terminates in a finite number of iterations and converges to an $\mathbf{x} \in H^0(G; \underline{\mathcal{F}})$.

In Appendix B, we review some experimental results on the convergence rate of the gossip algorithm (Algorithm 3).

[14] If $\tau(t) \cap \mathcal{N}_i = \emptyset$, then $(\tilde{L}_t)_i = 1$ because $\bigwedge \emptyset = 1$.

CHAPTER 8

Signals

Signal processing is a broad area of engineering devoted to the study of various sorts of signals. Some examples of signals include audio, video, speech, image, communication, GPS, sonar, lidar, and radar, among others. In our view, signal processing is synonymous with information processing.

At a high level of abstraction, a signal is a map $x : \Omega \to \mathcal{C}$ where Ω is a domain and $\mathcal{C}$ is an object in a data category. Classically, $\mathcal{C}$ is a finite-dimensional vector space and each dimension of $\mathcal{C}$ is called a *channel* or *feature*, however, the same terminology may be used if $\mathcal{C}$ is an arbitrary object in a data category. The set of signals over Ω valued in $\mathcal{C}$ is denoted $\mathcal{X}(\Omega, \mathcal{C})$.

Examples 8.1. Some traditional examples:

> *Waveforms.* Suppose T is the period of the waveform. Then, Ω is a the interval $[0, T]$ with and $\mathcal{C} = \mathbb{R}$ (electric potential, displacement etc.).
>
> *Images.* Ω is a grid $\mathbb{Z} \times \mathbb{Z}$ (pixels) and $\mathcal{C} = \mathbb{R}^d$ (intensities) where usually $d = 1$ (grayscale) or 3 (RBG).
>
> *Networks.* Ω is a graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $\mathcal{C} = \mathbb{R}^d$; each channel is a feature. In practice, channels are often features generated from data and the graph is constructed from a similarity measure between nodes/agents (e.g. Pearson correlation).

In Chapter 6 we introduced lattice-valued sheaves as a way to generalize sheaves valued in sets and vector spaces. If every stalk in a lattice-valued sheaf over a graph G is the same lattice, say $\mathbf{L}$, then it is perfectly reasonable to identify 0-cochains $C^0(G; \underline{\mathcal{F}})$ with signals $\mathcal{X}(\mathcal{V}_G, \mathbf{L})$. We might, then, hope for some of the themes of signal processing to port to lattice-valued sheaves.

8.1. Algebraic Signal Processing

Algebraic signal processing (ASP) is a unified signal processing framework based on the representation theory of associative algebras [51]. One limitation of ASP, however, is that $\mathcal{X}(\Omega, \mathcal{C})$ must be a vector space in this perspective. We first basics of representation theory.

Definition 8.1 (Algebras)**.** Let $\Bbbk$ be a field. An *associative algebra* $\Bbbk$, or $\Bbbk$-algebra for short, is a vector space A equipped with an associative bilinear map[15]

$$\cdot : A \times A \to A$$

A $\Bbbk$-algebra is *unital* if there exists an element $1 \in A$ with $a \cdot 1 = 1 \cdot a = a$. An *algebra homomorphism* between algebras $(A, \cdot)$ and $(A', \cdot')$ is a linear transformation $\rho : A \to A'$ such that $\rho(a \cdot b) = \rho(a) \cdot' \rho(b)$.

We provide several examples of algebras whose representationsa are of interest in engineering.

Example 8.1 (Polynomials)**.** Polynomials with coeficients in $\Bbbk$ form an algebra denoted $\Bbbk[t]$; multiplication is given by multiplication of polynomials. Multinomials, in the same way, form an algebra $\Bbbk[t_1, t_2, \ldots, t_n]$. The unit is the constant polynomial $1 \in \Bbbk[t]$ or $\Bbbk[t_1, t_2, \ldots, t_n]$.

Example 8.2 (Endomorphism Algebra)**.** Suppose V is a vector space. Then, $\operatorname{End} V$ is a $\Bbbk$-algebra with composition. If V is finite-dimensional, $\operatorname{End} V$ is isomorphic to the algebra $\mathrm{Mat}_{n\times n}(\Bbbk)$ of $n \times n$ matrices with coefficients in $\Bbbk$. The unit is the identity linear transformation.

Example 8.3 (Monoid Algebra)**.** Let $(M, \star, e)$ be a monoid. The *monoid algebra* denoted $\Bbbk[M]$ (sometimes called the *monoid ring*) is the free vector space generated by formal linear

[15] Equivalently, a linear map $A \otimes A \to A$

combinations of elements of M

$$\sum_{m\in M} h_m[m], \quad |\{h_m \in \Bbbk \mid h_m \neq 0\}| < \infty.$$

Multiplication in $\Bbbk[M]$ is defined on basis elements

$$[m][n] = [m \cdot n],$$

then extended by linearity. The algebra $\Bbbk[M]$ is unital with unit $[e]$. Equivalently, the monoid algebra consists of the vector space of functions $\Bbbk^M$ with multiplication

$$(f \circledast g)(m) = \sum_{p\cdot q = m} f(p)g(q) \quad f, g \in \Bbbk^M, \quad m, p, q \in M.$$

The monoid algebra is integral in the study of representations of finite monoids [155], a generalization of the theory of representations of finite groups [150].

Example 8.4 (Monoid Algebra)**.** Suppose M is the semilattice $M = \{x, y, z, 1\}$ with the multiplication table

$\wedge$	x	y	z	1
x	x	z	z	x
y	z	y	z	y
z	z	z	z	z
1	x	y	z	1

Then, suppose $h = [x] - [y]$ and $h' = [1] - 2[y] - [z]$, then

$$\begin{aligned} h \cdot h' &= \\ &= ([x] - [y]) \cdot ([1] - 2[y] - [z]) \\ &= [x \wedge 1] - 2[x \wedge y] - [x \wedge z] - [y \wedge 1] + 2[y \wedge y] + [y \wedge z] \\ &= [x] - 2[z] - [z] - [y] + 2[y] + [z] \\ &= [x] - 2[z] + [y] \end{aligned}$$

Example 8.5 (Incidence Algebra)**.** Suppose P is a locally finite[16] poset. Then, the *incidence algebra* of P, denoted $\Bbbk$P is a $\Bbbk$-algebra over the vector space of functions

$$f : \mathbb{I}\mathrm{P} \to \Bbbk$$

with multiplication

$$(f \circledast g)([x, y]) = \sum_{x \preccurlyeq y \preccurlyeq z} f([x, z])g([z, y]).$$

Equivalently, $\Bbbk$P is a monoid algebra on $\mathbb{I}$P with the appropriate monoid stucture.

Example 8.6 (Path Algebra)**.** A *quiver* is a tuple $Q = (\mathcal{V}_Q, \mathcal{E}_Q, h, t)$ consisting of sets $\mathcal{V}_Q, \mathcal{E}_Q$ (vertices, arrows) and maps $h, t : \mathcal{E}_Q \to \mathcal{V}_Q$ (head, tail). A path in a quiver is a sequence $\gamma = a_0 a_1 \dots a_\ell$ of elements of $\mathcal{E}_Q$ such that $h(a_i) = t(a_{i+1})$. Trivial paths e_i are indexed by $\mathcal{V}_Q$. Let Path(Q) denote the set of paths in Q. Then, the *path algebra* $\Bbbk Q$ is the free vector space generated by Path(Q) with a product of basis elements

$$\gamma_1 \cdot \gamma_2 = \begin{cases} \gamma_1\gamma_2 & h(\gamma_1) = t(\gamma_2) \\ 0 & h(\gamma_1) \neq t(\gamma_2) \end{cases}$$

[16] A poset is locally finite if every closed interval $[x, y]$ is finite poset. A finite poset is of course locally finite

Suppose A is a $\Bbbk$-algebra. A *representation of A* is a vector space V together with an algebra homomorphism $\rho : A \to \operatorname{End}(V)$. Representations of A form a category $\mathcal{R}ep_A$ with the following morphisms.

Definition 8.2 (Intertwining Maps)**.** Suppose (V, ρ) and (V, ρ') are representations of A. Then, an *intertwining map* is a linear transformation $\theta : V \to V'$ such that

$$\rho'_a \circ \theta = \qquad \theta \circ \rho_a$$

for all $a \in A$.

In the above examples, signal spaces are naturally paired with each algebra with a representation, leading to the following definition.

Definition 8.3 (Algebraic Signal Model)**.** An *algebraic signal model* is a triple $(A, \mathcal{X}, \rho)$. A is a $\Bbbk$-algebra whose elements are called *filters*; $\mathcal{X}$ is vector space whose elements are called *signals*; ρ is a representation

$$\rho : \qquad A \to \operatorname{End}(\mathcal{X}).$$

Given a filter $h \in A$, the linear tranformation[17] $\rho_h : \mathcal{X} \to \mathcal{X}$ is an *instantiation* of h. Acting on a signal $x \in \mathcal{X}$ by ρ_h is called *convolution*.

Remark 8.1. The original formulation of algebraic signal processing regards an algebraic signal model as an algebra A of filters together with an A-module. However, the data of an A module is the same as the data of a representation $\rho : A \to \operatorname{End}(\mathcal{X})$ by standard facts of representation theory [51]. We prefer to think of representations as maps to emphasize relationships between models.

We now show that familiar and unfamiliar signal processing frameworks fall under the banner of algebraic signal processing.

[17] We often write $\rho_h : \mathcal{X} \to \mathcal{X}$ for the linear transformation that is the image of an element $h \in A$ under ρ.

Example 8.7 (Graph Signal Processing [129])**.** Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ be a weighted graph with (graph) shift operator $S : \mathbb{R}^{\mathcal{V}_G} \to \mathbb{R}^{\mathcal{V}_G}$. There are several common choices for S, including the (normalized) adjacency or Laplacian matrix. A *graph filter* is an element of the algebra $\Bbbk[t]$. The signal space $\mathcal{X}(\mathcal{V}_G, \mathbb{R})$ consists is the vector space of functions $\mathbb{R}^{\mathcal{V}_G}$. Convolution, then, is a representation

$$\rho : \mathbb{R}[t] \to \operatorname{End}\left(\mathbb{R}^{\mathcal{V}_G}\right).$$

Because $\mathbb{R}[t]$ is generated by the single mononomial t, a representation is uniquely determined by $\rho(t)$ which we chose to be the graph shift operator. Hence, a graph filter is given by

$$\begin{aligned} \mathbf{z} = && \mathbf{h} \circledast \mathbf{x} \\ = && \rho_{\mathbf{h}} \cdot \mathbf{x} \\ = && \left(\sum_{k=0}^{\infty} h_k S^k\right) \cdot \mathbf{x} \end{aligned}$$

Let's consider a specific example of a graph filter, the *heat kernel* [126]. The heat kernal is a parameterized linear graph filter denoted H_t which acts on graph signals as

$$H_t \cdot \mathbf{x} = \left(\sum_{k=0}^{\infty} \frac{t^k e^{-t}}{k!} S^k\right) \cdot \mathbf{x}, \quad t \geqslant 0.$$

A graph filter, in practice, will typically have $h_k = 0$ for all $k \geqslant K$ where $K \ll n$. While there is a computational justification for this, there is also a theoretical one. Suppose $n = |\mathcal{V}_G|$. Then, by the Cayley-Hamilton Theorem, we can always write S^n as a linear combination of $\{I, S, S^2, \ldots, S^{n-1}\}$. Hence, we have an alternate description of graph filters as a representation of a smaller algebra

$$\rho : \Bbbk[t]/(t^n - 1) \to \operatorname{End}(\Bbbk^{\mathcal{V}_G})$$

where $\Bbbk[t]/(t^n - 1)$ is the algebra of polynomials of degree at most $n - 1$.

Example 8.8 (Quiver Signal Processing [130])**.** Suppose $Q = (\mathcal{V}_Q, \mathcal{E}_Q, h, t)$ is a quiver. A *representation* $\boldsymbol{\pi}_\bullet$ of Q is an assignment of a vector space $\boldsymbol{\pi}(v)$ for every $e \in \mathcal{V}_Q$ and a linear transformation $\boldsymbol{\pi}(e) : \boldsymbol{\pi}(t(e)) \to \boldsymbol{\pi}(h(e))$ for every $e \in \mathcal{E}_Q$. The *total space* of a representation is the direct sum

$$\mathrm{Tot}(\boldsymbol{\pi}) = \bigoplus_{i \in \mathcal{V}_Q} \boldsymbol{\pi}(i).$$

It is a fact that representations of Q are equivalent to representations of $\Bbbk Q$ [128]. In one direction, suppose $\boldsymbol{\pi}$ is a representation of Q. Then, we define

$$\rho^{\boldsymbol{\pi}} : \Bbbk Q \to \mathrm{End}\left(\mathrm{Tot}(\boldsymbol{\pi})\right)$$

by sending paths $\gamma = e_1 e_2 \dots e_\ell$ to compositions of linear maps

$$\boldsymbol{\pi}(\gamma) = \boldsymbol{\pi}(e_\ell) \cdots \boldsymbol{\pi}(e_2) \boldsymbol{\pi}(e_1),$$

and extending $\rho^{\boldsymbol{\pi}}$ linearly to the entire domain $\Bbbk Q$. Acting on $\mathrm{Tot}(\boldsymbol{\pi})$,

$$(\rho^{\boldsymbol{\pi}}([\gamma]) \cdot \mathbf{x})_j = \begin{cases} \boldsymbol{\pi}(\gamma) \cdot x_i & h(\gamma) = j,\ t(\gamma) = i \\ 0 & \text{otherwise} \end{cases}$$

Hence, repesentations of Q are in coorespondence with algebraic signal models, $(A, \mathcal{X}, \rho) = (\Bbbk Q, \mathrm{Tot}(\boldsymbol{\pi}), \rho^{\boldsymbol{\pi}})$.

Example 8.9 (Monoid Algebra)**.** Suppose M is a monoid. Then, in monoid signal processing, $A = \Bbbk\,[G]$ and $\mathcal{X} = \Bbbk^M$. Monoid convolution is a representation

$$\rho : \Bbbk\,[M] \to \mathrm{End}(\Bbbk^M). \tag{8.1}$$

Shift operators coorespond to generators of $\Bbbk[M]$ which, in turn, coorespond to the generators of M (if they exist).

8.2. Lattice Signal Processing

Narrowing our focus to lattices, we first study signals whose domain is a (semi)lattice, then signals whose channels are lattices, and, finally, signals whose domain and channels are lattices.

8.2.1. Signals on semilattices

Püschel and Wendell proposed [131] a signal processing framework on meet- or join- semilattices. In this case, Ω is a semilattice and $\mathcal{C}$ is the real or complex number. In our view, their key definition is convolution of semilattice signals, although, in a series of paper, they build up a complete theory including Fourier transforms [132], filters for denoising [148], and an algorithm for sampling [164].

Definition 8.4. Suppose $(\mathrm{S}, \wedge, 1)$ is a (meet-) semilattice.[18] Suppose $\mathbf{f} \in \Bbbk^{\mathrm{S}}$ is a signal and $\mathbf{h} \in \Bbbk^{\mathrm{S}}$ is a filter. Then, *convolution* of $\mathbf{f}$ by h is the following operation

$$(\mathbf{h} \circledast \mathbf{f})_y = \sum_{x \in \mathrm{S}} h_x f_{x \wedge y}. \tag{8.2}$$

If there is any ambiguity, convolutions are written $\mathbf{h} \circledast_\wedge \mathbf{f}$ or $\mathbf{h} \circledast_\vee \mathbf{f}$ depending on whether S is a meet- or join- semilattice.

Convolution can be written in more compact form by means of a *shift operator*

$$\begin{aligned} T_x : \quad & \Bbbk^{\mathrm{S}} \to \Bbbk^{\mathrm{S}} \\ (T_x \mathbf{f})_y = \quad & f_{x \wedge y}. \end{aligned}$$

[18] Without loss of generality, since S^{op} is a bounded join-semilattice.

Thus,

$$\mathbf{h} \circledast - = \sum_{x \in \mathrm{S}} h_x T_x. \tag{8.3}$$

is a map sending a filter $\mathbf{j}$ to an endomorphism on $\Bbbk^{\mathrm{S}}$. Let us consider some properties of the shift operators.

Lemma 8.1. Suppose $x, y \in \mathbf{L}$. Then,

1. T_x is linear.

2. $T_{x \wedge y} = T_x T_y$.

3. T_x is idempotent.

4. $T_x T_y = T_y T_x$.

Proof. Linearity is trivial. For (2),

$$\begin{aligned}(T_x T_y \mathbf{f})_z &= (T_y \mathbf{f})_{z \wedge x} \\ &= f_{(z \wedge x) \wedge y} \\ &= f_{(x \wedge y) \wedge z} \\ &= (T_{x \wedge y} \mathbf{f})_z.\end{aligned}$$

Because the meet operation in a semilattice is idempotent and commutative, (3) and (4) follow from (2). □

The following established associativity of filtering.

Lemma 8.2. Suppose $\mathbf{h}, \mathbf{h}' \in \Bbbk^{\mathrm{S}}$ are filters and $f \in \Bbbk^{\mathrm{S}}$ is a signal. Then,

$$\mathbf{h} \circledast (\mathbf{h}' \circledast \mathbf{f}) = \mathbf{h}' \circledast (\mathbf{h} \circledast \mathbf{f}).$$

Proof. Let $x, y \in \mathrm{S}$.

$$\mathbf{h} \circledast \mathbf{f} = \sum_{x \in \mathrm{S}} h_x T_x \mathbf{f}$$

so that

$$\begin{aligned} \mathbf{h} \circledast (\mathbf{h}' \circledast \mathbf{f}) &= \sum_{y \in \mathrm{S}} h_y T_y \left(\sum_{x \in \mathrm{S}} h'_x T_x \mathbf{f} \right) \\ &= \sum_{y \in \mathrm{S}} \sum_{x \in \mathrm{S}} h_y h'_x T_x T_y \mathbf{f} \end{aligned}$$

because T_y is linear. Then, by Lemma 8.2,

$$\sum_{y \in \mathrm{S}} \sum_{x \in \mathrm{S}} h_y h'_x T_x T_y = \sum_{y \in \mathrm{S}} \sum_{x \in \mathrm{S}} h'_x h_y T_y T_x.$$

Reversing the process implies $\mathbf{h} \circledast (\mathbf{h}' \circledast \mathbf{f}) = \mathbf{h}' \star (\mathbf{h} \circledast \mathbf{f})$. □

It follows that semilattice convolution is an algebraic signal model with $A = \Bbbk[\mathrm{S}]$, $\mathcal{X} = \Bbbk^{\mathrm{S}}$, and

$$\begin{aligned} \rho : &\quad \Bbbk[\mathrm{S}] \to \mathrm{End}\left(\Bbbk^{\mathrm{S}}\right) \qquad (8.4) \\ \rho\left(\sum_{x \in \mathrm{S}} h_x [x] \right) = &\quad \sum_{x \in \mathrm{S}} h_x T_x. \end{aligned}$$

Proposition 8.1. Convolution acting on the left (or right by commutativity) $\mathbf{h} \circledast - : \Bbbk^{\mathrm{S}} \to \Bbbk^{\mathrm{S}}$ (8.2) coincides with the algebra homomorphism ρ_h (8.4) applied to an algebraic filter $h \in A$.

Proof. Identity a filter $\mathbf{h} \in \Bbbk^{\mathrm{S}}$ with the coefficients of multinomial

$$h = \sum_{x \in \mathrm{S}} h_x [x].$$

Then, for $h \in \Bbbk[\mathrm{S}]$, $\mathbf{f} \in \Bbbk^{\mathrm{S}}$ define $\rho_h \cdot \mathbf{f}$ by $\mathbf{h} \circledast \mathbf{f}$. By Lemma 8.1, $\rho_h \in \mathrm{End}(\Bbbk^{\mathrm{S}})$. However, we still need to show that ρ is an algebra homomorphism. Suppose $h, h' \in \Bbbk[\mathrm{S}]$. Then,

$$\begin{aligned} h \cdot h' = && \left(\sum_{x \in \mathrm{S}} h_x[x]\right) \cdot \left(\sum_{y \in \mathrm{S}} h'_y[y]\right) \\ = && \sum_{z = x \wedge y} h_x h'_y T_z \end{aligned}$$

Lemma 8.2 and applying ρ yields

$$\begin{aligned} \rho_{h \cdot h'} = && \sum_{z = x \wedge y} h_x h'_y \rho([z]) \\ = && \sum_{x, y \in \mathrm{S}} h_x h'_y T_{x \wedge y} \\ = && \sum_{x \in \mathrm{S}} \sum_{y \in \mathrm{S}} h_x h'_y T_x T_y \\ = && \sum_{x \in \mathrm{S}} h_x T_x \left(\sum_{x \in \mathrm{S}} h'_y T_y\right) \\ = && \rho_h \rho_{h'}. \end{aligned}$$

Linearity in h is trivial. □

8.2.2. Signals on lattices

We argue signal processing on lattices is not equivalent to signal processing on semilattices. Suppose $(\mathbf{L}, \wedge, \vee, 0, 1)$ is a bounded lattice. We view $\mathbf{L}$ as a bounded semilattice, and thus a monoid, in two seperate ways: a meet-semilattice denoted $(L_\wedge, \wedge, 1)$ and a join-semilattice $(L_\vee, \vee, 0)$. However, the absorption identities (Definition 2.5) further restrict the structure of a lattice than merely a poset that is simulatneously a meet- and join- semilattice. For one, there are two families of shift operators, one for each element $x \in \mathbf{L}$

$$\begin{aligned} (T_x^\vee \mathbf{f})_y = && f_{x \vee y} \\ (T_x^\wedge \mathbf{f})_y = && f_{x \wedge y} \end{aligned}$$

Each family instantiates filters

$$\begin{aligned} \rho^{\wedge} : \quad & \Bbbk[L_{\wedge}] \to \mathrm{End}(\Bbbk^{\mathbf{L}}) \\ \rho^{\vee} : \quad & \Bbbk[L_{\vee}] \to \mathrm{End}(\Bbbk^{\mathbf{L}}) \end{aligned}$$

Given a bounded lattice $(\mathbf{L}, \wedge, \vee, 0, 1)$, we investigate the relationship between the algebraic signal models

$$\begin{aligned} \mathcal{M}_{\wedge} = \quad & (\Bbbk[L_{\wedge}], \Bbbk^{\mathbf{L}}, \rho^{\wedge}), \\ \mathcal{M}_{\vee} = \quad & (\Bbbk[L_{\vee}], \Bbbk^{\mathbf{L}}, \rho^{\vee}). \end{aligned}$$

Proposition 8.2. Suppose $\mathbf{L}$ is a lattice with shift operators $\{T_a^{\wedge}, T_a^{\vee}\}_{a \in \mathbf{L}}$ defined

$$\begin{aligned} (T_a^{\wedge}\mathbf{f})_x = \quad & f_{x \wedge a}, \\ (T_a^{\vee}\mathbf{f})_x = \quad & f_{x \vee a}. \end{aligned}$$

Then,

$$T_x^{\vee} T_x^{\wedge} = \quad T_x^{\wedge} T_x^{\vee}.$$

Furthermore,

$$T_x^{\vee} T_x^{\wedge} \mathbf{f} = \quad f_x \mathbf{1}$$

where $\mathbf{1} \in \Bbbk^{\mathbf{L}}$ is the constant unit signal.

Proof. Suppose $\mathbf{f} \in \Bbbk^{\mathbf{L}}$. The absorption identities (Ch imply

$$\begin{aligned}(T_x^{\vee} T_x^{\wedge} \mathbf{f})_y = && (T_x^{\wedge} \mathbf{f})_{x \vee y} \\ = && f_{(x \vee y) \wedge x} \\ = && f_x.\end{aligned}$$

Clearly, $(T_x^{\wedge} T_x^{\vee} \mathbf{f})_y = f_x$ as well. □

Corollary 8.1. $T_x^{\wedge} T_x^{\vee}$ and $T_x^{\vee} T_x^{\wedge}$ are idempotent.

Proof. Notice $T_x^{\wedge} \mathbf{1} = \mathbf{1}$ and $T_x^{\vee} \mathbf{1} = \mathbf{1}$. Then, by Proposition 8.2,

$$\begin{aligned}T_x^{\vee} T_x^{\wedge} T_x^{\vee} T_x^{\wedge} \mathbf{f} = && T_x^{\vee} T_x^{\wedge} f_x \mathbf{1} \\ = && f_x \mathbf{1} \\ = && T_x^{\vee} T_x^{\wedge} \mathbf{f}\end{aligned}$$

□

We will now introduce some of Fourier analysis presented in the original paper [132]. Suppose $\mathbf{L}$ is a bounded lattice. Let

$$1_{\{y \preccurlyeq x\}} = \begin{cases} 1 & y \preccurlyeq x \\ 0 & y \not\preccurlyeq x \end{cases}$$

be the indicator function on the boolean variable $\{y \preccurlyeq x\}$. Define a signals $\mathbf{f}^{\preccurlyeq y} = (1_{\{y \preccurlyeq x\}})_{x \in \mathbf{L}}$ and $\mathbf{f}^{\succcurlyeq y} = (1_{\{y \succcurlyeq x\}})_{x \in \mathbf{L}}$.

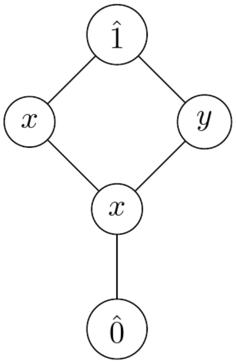
Lemma 8.3 (Theorem 1 [132])**.** Suppose $\mathbf{L}$ is a lattice with shifts $\{T_x^{\wedge}, T_x^{\vee}\}_{x \in \mathbf{L}}$. Then, $T_x^{\wedge} \mathbf{f}^{\preccurlyeq y} = 1_{y \preccurlyeq x} \mathbf{f}^{\preccurlyeq y}$ and $T_x^{\vee} \mathbf{f}^{\succcurlyeq y} = 1_{y \succcurlyeq x} \mathbf{f}^{\succcurlyeq y}$. Furthermore, $\{\mathbf{f}^{\preccurlyeq y}\}_{y \in \mathbf{L}}$ is an eigenbasis of $\Bbbk^{\mathbf{L}}$ with respect to $T_x^{\wedge}$. Dually, $\{\mathbf{f}^{\succcurlyeq y}\}_{y \in \mathbf{L}}$ is an eigenbasis of $\Bbbk^{\mathbf{L}}$ with respect to $T_x^{\vee}$.

Proof. See [132] for the $T_x^{\wedge}$ case. The $T_x^{\vee}$ cases follows by the duality principle. □

Every basis vector $\mathbf{f}^{\preccurlyeq y}$ (dually, $\mathbf{f}^{\succcurlyeq y}$) is an eigenvector of $T_x^{\wedge}$ with eigenvalue $1_{\{x \preccurlyeq y\}}$. Similarly, for $T_x^{\vee}$. Hence, the only eigenvalues of $T_x^{\wedge}$ and $T_x^{\vee}$ are 0 and 1. Define a change-of-basis transformation $\theta : \Bbbk^{\mathbf{L}} \to \Bbbk^{\mathbf{L}}$ by sending each $\mathbf{f}^{\preccurlyeq y} \in \{\mathbf{f}^{\preccurlyeq y}_{y \in \mathbf{L}}$ to $\mathbf{f}^{\succcurlyeq y} \in \{\mathbf{f}^{\succcurlyeq y}_{y \in \mathbf{L}}$. For the remainder of this section, suppose $\mathbf{L}$ is finite. Then, we can place a total order on the elements $\mathbf{L}$ (e.g. topological sort), labeling elements $\mathbf{L} = \{y_1, y_2, \ldots, y_m\}$. Now, in the finite case, the linear transformation θ is represented by an invertible matrix $\Theta = [\theta_{ij}]$ uniquely determined by writing each element of the basis $\{\mathbf{f}^{\succcurlyeq y}\}_{y \in \mathbf{L}}$ as a linear combination of basis elements $\{\mathbf{f}\}^{\preccurlyeq y}_{y \in \mathbf{L}}$.

$$\mathbf{f}^{\succcurlyeq y_i} = \qquad \sum_{i=1}^{m} \theta_{i,j} \mathbf{f}^{\preccurlyeq y_j}.$$

Example 8.10. Suppose $\mathbf{L}$ is the lattice



We order the elements of $\mathbf{L} = \{\hat{0}, z, x, y, \hat{1}\}$. We write $B_\wedge$ for the matrix whose columns are the meet-basis vectors

$$B_\wedge = \begin{bmatrix} \mathbf{f}^{\preccurlyeq \hat{0}} & \mathbf{f}^{\preccurlyeq z} & \mathbf{f}^{\preccurlyeq x} & \mathbf{f}^{\preccurlyeq y} & \mathbf{f}^{\preccurlyeq \hat{1}} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

Clearly, $B_\vee = B_\wedge^\top$ because $[B_\wedge]_{i,j} = 1_{\{x_i \preccurlyeq x_j\}}$ Then, $\Theta B_\wedge = B_\vee$ implies $\Theta = B_\vee B_\wedge^{-1}$. In our case, θ sends the signal

$$\mathbf{f} = \left(f_{\hat{0}}, f_z, f_x, f_y, f_{\hat{1}}\right)$$

to the signal

$$\theta(\mathbf{f}) = \left(f_{\hat{0}} - f_z, f_{\hat{0}} - f_x - f_y + f_{\hat{1}}, f_{\hat{0}} - f_y, f_{\hat{0}} - f_x, f_{\hat{0}}\right).$$

We now present an “impossibility result” demonstrating an isomorphism between the representations $\rho^{\wedge}$ and $\rho^{\vee}$ implies the underlying lattice is the trivial lattice $\mathbf{L} = \{0 = 1\}$.

Theorem 8.1. Suppose $\mathbf{L}$ is a lattice. Then,

$$\theta\left(T_x^{\wedge}\mathbf{f}\right) = T_x^{\vee}\theta(\mathbf{f}) \tag{8.5}$$

if and only if $\mathbf{L} = \{0 = 1\}$.

Proof. One direction is trivial. Suppose (8.5) holds. Then, by Lemma 8.3, we can write $\mathbf{f} = \sum_{y\in\mathbf{L}} h_y \mathbf{f}^{\preccurlyeq y}$ with $h_y \in \Bbbk$. On one hand,

$$\begin{aligned}
\theta\left(T_x^{\wedge}\mathbf{f}\right) &= \theta T_x^{\wedge}\left(\sum_{y\in\mathbf{L}} h_y \mathbf{f}^{\preccurlyeq y}\right) \\
&= \sum_{y\in\mathbf{L}} h_y \theta T_x^{\wedge y} \mathbf{f}^{\preccurlyeq y} \\
&= \sum_{y\in\mathbf{L}} h_y 1_{y\preccurlyeq x} \theta(\mathbf{f}^{\preccurlyeq y}) \\
&= \sum_{y\in\mathbf{L}} h_y 1_{y\preccurlyeq x} \mathbf{f}^{\succcurlyeq y}
\end{aligned}$$

On another hand,

$$\begin{aligned}
T_x^{\vee}\theta(\mathbf{f}) = & \quad T_x^{\vee}\theta\left(\sum_{y\in\mathbf{L}} h_y\mathbf{f}^{\preccurlyeq y}\right) \\
= & \quad \sum_{y\in\mathbf{L}} h_y T_x^{\vee}\theta(\mathbf{f}^{\preccurlyeq y}) \\
= & \quad \sum_{y\in\mathbf{L}} h_y T_x^{\vee}\mathbf{f}^{\succcurlyeq y} \\
= & \quad \sum_{y\in\mathbf{L}} h_y 1_{y\succcurlyeq x}\mathbf{f}^{\succcurlyeq y}
\end{aligned}$$

Therefore, it must hold that, given a fixed $x \in \mathbf{L}$, $y \preccurlyeq x$ if and only if $y \succcurlyeq x$. This is only possible if $\mathbf{L} = \{\hat{0} = \hat{1}\}$ is a one-element lattice. □

We provide two examples of signals over lattices and their convolutions.

Example 8.11 (Encodings)**.** Given an element of a lattice $x \in \mathbf{L}$ we can encode it as a signal $\mathbf{1}_x : \mathbf{L} \to \mathbb{R}$ with

$$\mathbf{1}_x(y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}.$$

In the signal-processing literature, this is called a *one-hot* encoding since exactly one component of the signal is "hot," while the rest are "cold." In practice, noise is sometimes added to one-hot encodings for numerical stability. The one-hot encoding is useful for applying continuous information processing techniques to lattices (algebraic structures). For instance, if $\mathbf{L}$ is finite, $\mathbf{L} = \{\hat{0} = y_1, y_2, \ldots, y_m = \hat{1}\}$ is an total ordering of the elements of $\mathbf{L}$, and $\mathcal{F}$ is an approximation of the constant sheaf $\underline{\mathbf{L}}$ over G, then the Tarski Laplacian can be identifies as a map $\mathbf{L} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$. In general, the Tarski Laplacian of a lattice-valued sheaf whose stalks are finite lattices can be expressed as a map

$$L : \mathbb{R}^{\sum_{i\in\mathcal{V}_G}|\mathcal{F}(i)|} \to \mathbb{R}^{\sum_{i\in\mathcal{V}_G}|\mathcal{F}(i)|}$$

More efficient encodings of lattice elements is a subject of interest. For instance, if a lattice is finite, then every element is a join of join irreducible elements. Hence, you could represent an element of a finite lattice with a few-hot encoding, however, not uniquely.

Example 8.12 (Hypergraph Convolution)**.** With mimicking graph signal processing [168] as one approach to filtering signals on hypergraphs, lattice convolution, as was suggested in the introductory paper [132], is another approach. Suppose $H = (\mathcal{V}_H, \mathcal{H}_H)$ is a hypergraph. Recall, a hypergraph signal is a map $\mathbf{f} : \mathcal{V}_H \to \mathbb{R}$. Suppose $\mathcal{I}$ is the incidence relation of H. Then, the Galois lattice is the following

$$\mathrm{Gal}(\mathcal{I}) = \qquad \{(\sigma, \tau) \subseteq \mathcal{V}_H \times \mathcal{H}_H \mid \cap_{i\in\sigma} i^{\uparrow} = \tau, \sigma = \cap_{e\in\tau} e^{\downarrow}\},$$

which we label $\mathrm{Gal}(H)$ in this context. Recall, $\mathrm{Gal}(H)$ is isomorphic to the following fixed-point lattices

$$\begin{aligned} \mathrm{Cl}(\wp(\mathcal{V}_H)) = &\qquad \{\sigma \subseteq \mathcal{V}_H \mid \sigma^{\uparrow\downarrow} = \sigma\} \subseteq &\qquad \wp(\mathcal{V}_H), \\ \mathrm{Cl}(\wp(\mathcal{H}_H)) = &\qquad \{\tau \subseteq \mathcal{H}_H \mid \tau^{\downarrow\uparrow} = \tau\} \subseteq &\qquad \wp(\mathcal{H}_H). \end{aligned}$$

Signals on $H = (\mathcal{V}_H, \mathcal{H}_H)$ are ported to signals on $\mathrm{Gal}(H)$ via aggregations. We hypothesize maximum is a good choice, because the hypergraph diffusion energy functional [33] is of the form

$$Q(\mathbf{f}) = \qquad \sum_{e\in\mathcal{H}_H} \max_{i,j\in e} (f_i - f_j)^2.$$

8.2.3. Signals on posets

If Ω is a finite poset P and $\mathcal{C}$ is again the real or complex number, then one may port the signal processing framework on lattices to a new framework on finite posets under Birkhoff

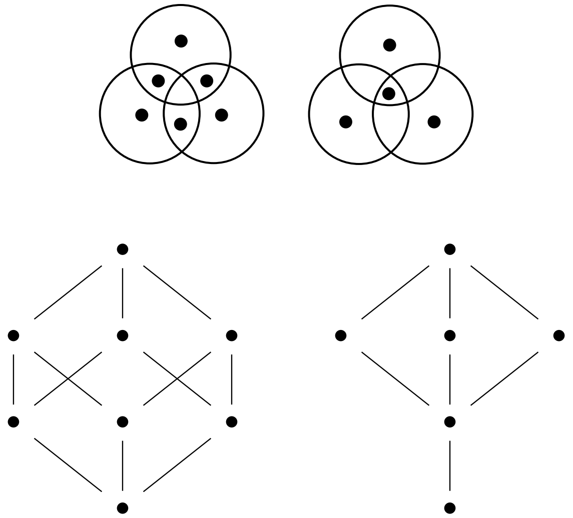


Figure 8.1: Hypergraphs and their Galois lattices.

Duality (Theorem 2.3). In one direction, given a finite poset P there is an isomorphism

$$\begin{aligned} \eta : \quad & \mathrm{P} \xrightarrow{\cong} \mathcal{J}\left(\mathcal{D}\left(\mathrm{P}\right)\right) \\ \eta(p) = \quad & \downarrow p \end{aligned} \tag{8.6}$$

which implies the join-irreducibles of the distributive lattice $\mathcal{D}\left(\mathrm{P}\right)$ are in one-to-one correspondence with the elements of P. Hence, each $p \in \mathrm{P}$ corresponds to a unique join-irreducible $\downarrow p$. This implies the monoid algebra $\Bbbk[\mathcal{D}\left(\mathrm{P}\right)_{\vee}]$ has a generating set $\mathcal{G} = \{[\downarrow p]\}_{p \in \mathrm{P}}$. Hence, the shift operators in the representation framework are given by the images of $g \in \mathcal{G}$ under the representation

$$\rho_{\vee} : \quad \Bbbk\left[\mathcal{G}\right] \to \mathrm{End}\left(\Bbbk^{\mathcal{D}(\mathrm{P})}\right).$$

Recent work, takes a similar approach and generalizes (semi)lattice signal processing to signal processing on directed acyclic graphs (DAGs) [149]. The authors of this work do not, however, construct convolutions via Birkhoff Duality as we suggest.

8.2.4. Lattice-valued signals

On the other hand, suppose Ω is an abelian group (e.g. $\mathbb{Z}$ or $\mathbb{Z}/n\mathbb{Z}$) and $\mathcal{C}$ is a lattice. A lattice-valued signal, then, is a map $\mathbf{f} : A \to \mathbf{L}$. Lattice-valued signals are a natural taxum of signal to study. For instance, if $\mathcal{C}$ is the lattice of partitions of the nodes of a dynamic graph, a signal $f \in \mathcal{X}(\mathbb{Z}, \Pi(\mathcal{V}_G))$ would model the merging and splitting of connected components of a network as connections between nodes are established/disconnected [92].

Definition 8.5 (Lattice-Valued Convolution [110])**.** Suppose $(\mathbb{L}, \wedge, \vee, 0, 1, \star, e, \to)$ is a residuated lattice and $(A, +, 0)$ an abelian group. Define the following *sup-convolution*, $f, h \in \mathcal{X}(A, \mathbf{L})$,

$$(h \circledast f)(x) = \bigvee_{s \in A} h(s) \star f(t-s). \tag{8.7}$$

Here is an example leading to nonlinear filters in the time domain.

Example 8.13 (1-D Max-Plus Convolution)**.** Suppose Ω is $\mathbb{Z}$ and $\mathcal{C}$ is the lattice $\mathbb{R}_{\max}$. If $h, f : \mathbb{Z} \to \mathbb{R}_{\max}$ are signals, then

$$(h \circledast f)(t) = \max_{s \in \mathbb{Z}} h(s) + f(t-s).$$

Of course, $\mathcal{C}$ could have multiple channels (i.e. $\mathcal{C} = \mathbb{R}^d_{\max}$) in which case each channel can be processed in parallel.

Let $\alpha \in \mathbb{L}$ be an element of a residuated lattice. If $f \in \mathcal{X}(\Omega, \mathbb{L})$ is a residuated-lattice -valued signal over some domain Ω, we define a *scalar multiplication* of a signal f by an $a \in \mathbf{L}$

$$(a \star f)(t) = a \star f(t) \quad \forall t \in \Omega.$$

If $f : \mathbb{Z} \to \mathcal{C}$ is a signal, then its domain translation is $T_s f : \mathbb{Z} \to \mathcal{C}$ given by $T_s f(t) = f(t-s)$.

We say an operator $D : \mathbf{L}^{\mathbb{Z}} \to \mathbf{L}^{\mathbb{Z}}$ is *dilation-translation invariat* if the following conditions are satisfied

$$\begin{aligned} D\left(\bigvee_{i \in I} f_i\right) &= \bigvee_{i \in I} D(f_i) \\ D(a \star f) &= a \star D(f) \\ T_s D(f) &= D(T_s f) \end{aligned}$$

for all $f \in \mathbb{L}^{\mathbb{Z}}$, $a \in \mathbb{L}$, $s \in \mathbb{Z}$. The first requirement states that D is a join-preserving map with respect the lattice structure on $\mathbb{L}^{\mathbb{Z}}$. The second and third specify an equivariance property.

The following is the closest to an "algebraic result" for this flavor of signal processing.

Theorem 8.2 (Theorem 3 [109])**.** An operator $D : \mathbf{L}^{\mathbb{Z}} \to \mathbf{L}^{\mathbb{Z}}$ is DTI dilation-translation invariant if and only if every signal $f \in \mathbb{L}^{\mathbb{Z}}$ is expressed as

$$f(t) = \bigvee_{s \in \mathbb{Z}} f(s) \star D(q)(t - s)$$

where $q : \mathbb{Z} \to \mathbf{L}$ is the signal called the *impulse* defined

$$q(t) = \begin{cases} e & t = 0 \\ 0 & t \neq 0 \end{cases}$$

Our contribution here is the following definition.

Definition 8.6 (Püschel-Maragos Convolution)**.** Suppose P is a finite distributed lattice and $\mathbb{L}$ is a residuated lattice. Then, the Püschel-Maragos convolution of $h, f : \mathrm{P} \to \mathbb{L}$ is the map

$$\begin{aligned} h \circledast f : && \mathrm{P} \to \mathbb{L} \\ (h \circledast f)(x) = && \bigvee_{y \in \mathbf{K}} h(y) \star f(x \wedge y) \end{aligned}$$

Theorem 8.3. Suppose $f : \mathrm{P} \to \mathbb{L}$ is join-preserving. Then, $h \circledast f : \mathrm{P} \to \mathbb{L}$ is join-preserving, also. Furthermore, the map

$$\begin{aligned} - \circledast f : && \mathbb{L}^{\mathrm{P}} \to \mathbb{L}^{\mathrm{P}}, \\ (h \circledast f)(x) = && \bigvee_{y \in \mathrm{P}} f(x \wedge y) * h(y) \end{aligned}$$

has a right adjoint

$$\begin{aligned} - \circledast' f : && \mathbb{L}^{\mathrm{P}} \to \mathbb{L}^{\mathrm{P}}, \\ (h \circledast' f)(y) = && \bigwedge_{x \in \mathrm{P}} f(x \wedge y) \to h(y). \end{aligned}$$

Proof. Follows from Theorem 3.4. □

8.3. Convolutional Neural Networks

Convolution neural networks (neural networks in general) consist of a series of layers. Each layer ℓ has an input signal $\mathbf{f}^{(\ell-1)}$ and an output signal $\mathbf{f}^{(\ell)}$ so that the output to layer ℓ is the input of layer $\ell + 1$. The input layer is $\mathbf{x} = \mathbf{f}^{(0)}$; the output layer is denoted $\mathbf{y}$. In each layer, the following operations may be performed.

1. Filter of the input $\mathbf{h}^{\ell}$ signal with a bank of filters.

2. Aggregate filtered signals.

3. Apply a nonlinearity σ_{ℓ} (e.g. agregation,ReLU or sigmoid).

4. (Optional) Pooling.

In our view, convolutional neural networks fall under the guise of signal processing.

8.3.1. Lattice neural networks

Convolutional layers in a lattice convolutional neural network (L-CNN) have two flavors. Meet- and join- convolutional layers. Fix a finite lattice $\mathbf{L}$, and choose an (arbitrary) total order on the elements $\mathbf{L} = \{\hat{0} = y_1, x_2, \ldots, x_m = \hat{1}\}$. Hence, we can identify a lattice signal $f : \mathbf{L} \to \mathbb{R}$ as a vector $\mathbf{f} \in \mathbb{R}^m$. For layer ℓ, let F_ℓ denote the number of features in that layer. Each feature has an associated lattice signal, stored in a matrix

$$\mathbf{F}^{(\ell)} = \{\mathbf{f}_i\}_{i=1}^{F_{\ell-1}} \in \mathbb{R}^{m \times F_\ell}.$$

Similarly, filters are stored in a tensor

$$\mathbf{H}^{(\ell)} = \{\mathbf{h}_{i,j}^{(\ell)}\}_{\substack{i=1,\ldots,F_{\ell-1} \\ j=1,\ldots,F_\ell}} \in \mathbb{R}^{m \times F_{\ell-1} \times F_\ell}.$$

Then, a meet-convolution layer is

$$\begin{aligned} \mathbf{F}_j^\ell = && \sigma_\ell \left(\sum_{i=1}^{F_{\ell-1}} \mathbf{h}_{i,j}^{(\ell)} \circledast_\wedge \mathbf{f}_j \right) \\ j \in && \{1, 2, \ldots, F_\ell\}, \end{aligned}$$

while a join-convolutional layer is

$$\begin{aligned} \mathbf{F}_j^\ell = && \sigma_\ell \left(\sum_{i=1}^{F_{\ell-1}} \mathbf{h}_{i,j}^{(\ell)} \circledast_\vee \mathbf{f}_j \right) \\ j \in && \{1, 2, \ldots, F_\ell\}. \end{aligned}$$

Either choice consists of lattice convolution by filters $\mathbf{h}_i^j$, aggregation (sum), followed by nonlinearity σ_ℓ. A note on implementation, since signals in every layer are supported on the same lattice, a Fourier basis $\{f^{\leqslant y_i}\}_{i=1,\ldots,m}$ suffices to write every convolution operation as a matrix multiplication. In practice, you may implement a L-CNN with a few fully-connected

(MLP [68]) layers, for example, in a classifier.

Example 8.14 (Point Cloud Classification)**.** A point cloud $\mathbb{Y}$ is a finite subset of $\mathbb{R}^d$. For instance, point clouds can be collected from LiDAR (Light Detection and Ranging) systems. There is a well-known pipeline to "approximate" $\mathbb{Y}$ as a simplicial complex. Given resolution $\varepsilon > 0$, define the *Vietoris-Rips complex*

$$\mathrm{VR}_\varepsilon(\mathbb{Y}) = \qquad \{\sigma \in \wp(\mathbb{Y}) \mid B_\varepsilon(x) \cap B_\varepsilon(y) \neq \emptyset \quad \forall x, y \in \sigma\}.$$

This leads to a filtration $\mathbb{X}_\varepsilon = \mathrm{VR}_\varepsilon$. It has been argued that, while this filtration is robust to some kinds of noise, it is not robust to outliers. It was originally suggest by Carlsson and Zomorodian [29] to filter $\mathbb{Y}$ also by density ρ (e.g. estimated with k-nearest neighbors). The intuition is that certain topological features can be captured at different sampling resolutions. Let

$$\mathrm{VR}_\varepsilon(\mathbb{Y}^{\geqslant \rho})$$

denote the Vietoris-Rips complex of the sub- point cloud of point with density greater than or equal to ρ. Then, $\mathbb{X}_{(\varepsilon,\rho)} = \mathrm{VR}_\varepsilon(\mathbb{Y}^{\geqslant \rho})$ clearly forms a bifiltration of simplicial complexes. Bi-filtrations have several known topological invariants: the Hilbert function and bi-graded Betti numbers [101]. Upon discretizing and rescaling the domain of ε and ρ, each invariant is seen as a signal on the grid lattice $\mathbf{L} = [m] \times [n]$ where $[n] = \{0 < 1 < \cdots < n\}$. To this end, we trained a L-CNN [136] in order to classify point clouds in the ModelNet10 dataset [167] strictly using topological invariants produced with the RIVET software package [101] as features. The results are summarized in Appendix B.

8.3.2. Towards graph neural networks (GNNs)

Graph Neural Networks (GNNs) are a powerful models offering more efficient and generalizable solutions to node classification, network classification, and representation learning tasks. It is widely argued that graph diffusion is the driving force behind graph neural networks, the impetus for our study of foundational models of graph neural networks based upon the Tarski Laplacian (as well as other linear sheaf Laplacians [20, 12, 74]).

We will review three types of GNNs, each of which brings to light different aspects of neural information processing on graphs. First we establish some common notation. Suppose $G = (\mathcal{V}_G, \mathcal{E}_G, \omega)$ is a weighted graph with $n = |\mathcal{V}_G|$ nodes, (normalized) adjacency matrix $A \in \mathbb{R}^{n\times n}$, degree matrix $D \in \mathbb{R}^{n\times n}$, and (normalized) graph Laplacian $L \in \mathbb{R}^{n\times n}$. Suppose each layer ℓ of the GNN has F_ℓ features so that $\mathbf{X}^{(\ell)} \in \mathbb{R}^{n\times \ell}$ is a feature matrix of layer ℓ. Each layer is associated another matrix $\mathbf{H}^{(\ell)} \in \mathbb{R}^{F_\ell \times F_{\ell+1}}$ called the *weight matrix* or *channel-mixing* matrix. The entries of each $\mathbf{H}^{(\ell)}$ are learnable parameters. Finally, each layer has a nonlinear function σ_ℓ that is either pointwise (e.g. ReLU, sigmoid) or a localized nonlinear function such as local maximum or median [143].

1. Graph Convolutional Networks (GCNs) [93] are deep neural networks that incorporate elements of graph diffusion and mutli-layer perceptrons. Layers are of the form

$$\mathbf{X}^{(\ell+1)} = \sigma_\ell\left(L\mathbf{X}^{(\ell)}\mathbf{H}^{(\ell)}\right), \tag{8.8}$$

or, locally,

$$x_{i,g}^{(\ell+1)} = \sigma_\ell\left(\sum_{f=1}^{F_\ell} h_{f,g}^{(\ell)}\left(L\mathbf{x}_f^{(\ell)}\right)_i\right), \quad \forall i \in \mathcal{V}_G,\ g \in \{1, 2, \ldots, F_{\ell+1}\}.$$

Only one diffusion step takes place in each layer. Graph convolution takes place, but each graph filter only has one non-zero parameter.

2. Message-Passing Neural Networks (MPNNs) [65] take quite a general form.

   (a) Each node $i \in \mathcal{V}_G$ receives messages from neighboring nodes consisting via a function $\psi_{i,j}^{(\ell)}\left(\mathbf{f}_i^{(\ell)}, \mathbf{f}_j^{(\ell)}, a_{ij}\right)$ (learned parameters possible).

(b) Node $i \in G$ aggregates the messages received by neighbors with a map

$$\square : \mathbb{R}^{\mathcal{N}_i} \to \mathbb{R},$$

for instance, sum, maximum, or mean.

(c) Then, a node updates its state with an update function $\varphi^{(\ell)}$ (learned parameters possible) depending on the previous state $\mathbf{f}_i^{(\ell)}$ and the aggregated message $\mathbf{m}_i^{(\ell+1)}$.

In summary,

$$\begin{aligned} \mathbf{m}_i^{(\ell+1)} &= \mathop{\square}_{w \in \mathcal{N}_i} \psi_{i,j}^{(\ell)} \left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(\ell)}, a_{ij} \right), \\ \mathbf{x}_i^{(\ell+1)} &= \varphi_i^{(\ell)} \left(\mathbf{x}_i^{(\ell)}, \mathbf{m}_i^{(\ell+1)} \right). \end{aligned} \tag{8.9}$$

3. Graph Convolutional Neural Networks (GCNNs) [58] epitomize a perspective that *GNNs are build from filters* and emphasize the spectral domain. Each layer can have multiple diffusions take place, aggregating information in a k-hop neighorhood, as opposed to a 1-hop neighborhood.

$$x_{i,g}^{(\ell+1)} = \sigma_\ell \left(\sum_{f=1}^{F_\ell} \sum_{k=0}^{K_\ell} h_{f,g,k}^{(\ell)} \left(L^k \mathbf{x}_f^{(\ell)} \right)_i \right), \quad \forall i \in \mathcal{V}_G, \; g \in \{1, 2, \ldots, F_{\ell+1}\}.$$

An iteration of heat flow 7.13 can be viewed as a layer in a message-passing neural network as follows. Suppose $\underline{\mathcal{F}}$ is a Tarski sheaf over G. Then, with the following,

$$\begin{aligned} \psi_{i,j}^{(\ell)}(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(\ell)}, a_{ij}) &= P_{j \to i}^{\mathcal{F}}(\mathbf{x}_j^{(\ell)}) \\ \mathop{\square}_{j \in \mathcal{N}_i} \mathbf{x}_j &= \bigwedge_{j \in \mathcal{N}_i} \mathbf{x}_j \\ \varphi_i^{(\ell)}(\mathbf{x}_i^{(\ell)}, \mathbf{m}_i^{(\ell+1)}) &= \mathbf{x}_i^{(\ell)} \wedge \mathbf{m}_i^{(\ell+1)}, \end{aligned}$$

heat flow is a message-passing neural network.

An open question remains whether the Tarski Laplacian inspires convolutional neural networks. With respect to an underlying lattice-valued sheaf over G, we hypothesize that signals on lattice $\mathcal{F}(i)$ are correlated related to signals on neighboring lattices $\{\mathcal{F}(j)\}_{j \in \mathcal{N}_i}$. This leads to a possible notion of signal convolution relative to sheaf.

Definition 8.7 (Convolution Relative to a Sheaf)**.** Suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a graph and $\mathcal{F}$ is a Tarski sheaf over G. Suppose $\mathbf{F} = \{\mathbf{f}_i\}_{i \in \mathcal{V}_G}$ is a collection of signals over lattices indexed by $\mathcal{V}_G$, suppose $\mathbf{H} = \{\mathbf{h}^{i,j}\}_{i,j \in \mathcal{V}_H}$ in a filter bank, and suppose $L : C^0(G; \underline{\mathcal{F}}) \to C^0(G; \underline{\mathcal{F}})$ is the Traski Laplacian. Then, *convolution of* $\mathbf{F}$ *by a filterbank* $\mathbf{H}$ *relative to* $\mathcal{F}$ is the operation

$$(\mathbf{H} \star_{\mathcal{F}} \mathbf{F})_i = \sum_{j \in \mathcal{V}_G} \sum_{\mathbf{y} \in C^0(G; \underline{\mathcal{F}})} h^{i,j}_{y_i} T^{\wedge}_{(L\mathbf{y})_i} \mathbf{f}_i \tag{8.10}$$

CHAPTER 9

Semantics

Our second application domain can be loosely described as *semantics.* In this chapter, we explore semantics in two different contexts: temporal logic and multi-agent logic.

In logic, syntax consists of strung-together symbols, while semantics is found "under the hood." Semantics also has a linguistic connotation as something like "subsurface layers of meaning," and, of course, meaning is highly dependent on context. Thus, semantics is synonymous with a formal context which we call a *model* and others call a *structure.* A model consists of a set of *states* and rules for verifying whether a given state satisfies a logic formula. Model checking [9] is a subfield of formal verification in computer science dedicated to solving the following problems.

Problem 9.1 (Model Checking)**.** Given a model, a particular state s, and a formula φ (in an appropriate language), decide whether s satisfies φ, written $(M, s) \models \varphi$.

Problem 9.2 (Validity)**.** Given a model M and a formula φ, decide whether every state satisfies a given formula or, equivalently, check every s satisfies $\neg\varphi$. If φ is valid, it is common to write $M \models \varphi$.

Problem 9.3 (Planning)**.** Given a M and a formula φ, produce a state s with $(M, s) \models \varphi$ (if such a state exists).

For models whose rules are specified by satisfying a given relation, we observe that Galois connections induced by relations (Theorem 9.3) transform semantic content. Thus, we argue, as others have done [96], that order theory is practical in semantics. To date, we see formal theories of semantics having been applied to single- and multi-agent autonomous systems as well as artificial intelligence. It is in the area of autonomy that we hope our work can make an impact.

9.1. Temporal Logic

Temporal logic is a modal logic with much success in interpreting abstract systems by specifying logical constraints a system must satisfy as well as the time or duration. Two critical properties that temporal logic model checking address are *saftey*, something “bad” will never happen, and *liveness*, something “good” will always happen, at least eventually. More recently, temporal logic control systems have been utilized in path-planning [154, 53] and multi-agent autonomous systems [88].

9.1.1. Transition systems

The logic of time, temporal logic, is powerful for modeling systems. We utilize a popular method of abstracting away a system. We begin with a set of *atomic propositions*, denoted Φ. Elements of Φ are labeled p_1, p_2 etc. In the most elementary setting, these propositions have two possible truth values $\{\texttt{false}, \texttt{true}\}$ and typically designate facts about a system, for instance, whether an agent is located in particular region of a map, or whether there is a communication link between a pair of agents. In temporal logic, the truth values of atomic propositions are time-variant.

Definition 9.1. A (labeled nondeterministic) *transition system* is a tuple $\mathbf{TS} = (Q, Q_0, \Sigma, \delta, \Phi, \pi)$ with

1. Q, a set of states,
2. $Q_0 \subseteq Q$, a set of initial states,
3. Σ, a set of control inputs,
4. δ, a transition map sending a states $q \in Q$ and a control input $\sigma \in \Sigma$ to a subset of possible next states

$$\delta : Q \times \Sigma \to \wp(Q), \tag{9.1}$$

5. Φ, a set of atomic propositions (true or false),

6. π, a map sending a state $q \in Q$ to a set of atomic propositions true at q

$$\pi : Q \to \wp(\Phi).$$

Notice, we could have write the transition map as a labeled relation

$$\to \subseteq \; Q \times \Sigma \times Q$$

by currying δ. Thus, transition systems are conveniently drawn as directed graphs with arrows labeled by subsets of Σ. In an *deterministic transition system*, (9.1) is replaced by

$$\delta : Q \times \Sigma \to Q.$$

and we assume $Q_0 = \{Q_0\}$. In an *unlabeled transition system*, we ignore Σ and replace (9.1) by

$$\delta : Q \to \wp(Q), \tag{9.2}$$

equivalently, a relation $\to \subseteq Q \times Q$. If Q, Σ and Φ are finite, we say the transition system is *finite*.

Remark 9.1. In a transition system, uncertainty is not (ordinarily) modeled with probability, but agnostic to the likelihood of a given state q transitioning to any number or states $q' \in \delta(q, \sigma)$. We can form an unlabeled transition system from a labeled one via

$$\tilde{\delta}(q) = \bigcup_{\sigma \in \Sigma} \delta(q, \sigma).$$

An unlabeled transition system is equivalent to a binary relation $\to \subseteq \; Q \times Q$.

Example 9.1 (Control Systems)**.** Consider the discrete time-invariant linear system

$$\mathbf{x}[t+1] = A\mathbf{x}[t] + B\mathbf{u}[t] \tag{9.3}$$

$$\mathbf{y}[t+1] = C\mathbf{x}[t+1] \tag{9.4}$$

with $A \in \mathbb{R}^{n\times n}$, $\mathbf{x} \in \mathbb{R}^{n}$, $B \in \mathbb{R}^{n\times p}$, $\mathbf{u} \in \mathbb{R}^{p}$, $C \in \mathbb{R}^{m\times n}$, $\mathbf{y} \in \mathbb{R}^{m}$. Linear systems are ubiquitous in control theory, for instance, model-predictive control [28]. The system (9.3) is rewritten as the following transition system:

$$\begin{aligned}
Q &= \mathbb{R}^{n} \\
\Sigma &= \mathbb{R}^{p} \\
\Phi &= \mathbb{R}^{m} \\
\delta(\mathbf{x},\mathbf{u}) &= \{A\mathbf{x} + B\mathbf{u}\} \\
\pi(x) &= \{C\mathbf{y}\}.
\end{aligned}$$

Let Q^{ω} denote the set of infinite sequences of elements in Q

$$Q^{\omega} = \{\alpha : \mathbb{N} \to Q\}$$

which we call traces. If $\tau \in Q^{\omega}$, the *suffix* of τ is the trace $\tau[t_0]$ starting at $t_0 \geqslant 0$. Suppose $\sigma_{in} \in \Sigma^{\omega}$ is a sequence of control inputs, for instance, driving directions: *turn left at the light, go straight at the stop sign.* Then,

$$\begin{aligned}
\tau(t) &= \delta(\sigma(t), \tau(t-1)) \\
\tau(0) &\in Q_0
\end{aligned}$$

defines a *trace* $\tau \in \wp(Q)^{\omega}$. Finally, a trace specifies a sequence of subsets of Φ via

$$\sigma_{out}(t) = \pi\left(\tau(t)\right).$$

Before we describe how to subject an output σ_{out} to temporal logical constraints, consider the following critical example.

Example 9.2 (Network Connectivity [135])**.** Suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ is an undirected graph.[19] A *dynamic graph* is map

$$G_t : \mathbb{N} \to \text{Subgraph}\,(G) \tag{9.5}$$

is a dynamic graph with edges $\mathcal{E}_{Gt} \subseteq \mathcal{E}_G$ collecting edges of the subgraph G_t at time t. At each discrete time instant, nodes in the edge $ij \in \mathcal{E}_G$ toggle their status from on to off or from off to on or are on standby. These actions are represented by the following control inputs

$$\Sigma = \prod_{ij \in \mathcal{E}_G} \{\sigma_{switch}, \quad \sigma_{sleep}\}.$$

States are represented

$$\mathbf{Q} = \{-1, +1\}^{|\mathcal{E}_G|}.$$

State transitions are governed by the function

$$\delta(\mathbf{s}, \boldsymbol{\sigma})_{ij} = \begin{cases} -q_{ij} & \sigma^{ij} = \sigma_{switch} \\ q_{ij} & \sigma^{ij} = \sigma_{sleep} \end{cases}$$

[19] Without loss of generality, we can think of G as the complete graph on vertices $\mathcal{V}_G$

given $\mathbf{q} \in Q$, $\boldsymbol{\sigma} \in \Sigma$. Let

$$\Phi = \qquad \{p^{ij}\}_{ij \in \mathcal{E}_G}$$

where p^{ij} is the proposition

> *There is a connection between node i and node j.*

Then, of course,

$$\pi(\mathbf{s}) = \qquad \bigcup_{q_{ij}>0} \pi^{ij}.$$

Hence, a control input $\boldsymbol{\sigma}_{in} \in \Sigma^\omega$ determines an output $\boldsymbol{\sigma}_{out} \in \wp(\Phi)^\omega$ specifying a subset of edges $\mathcal{E}_{Gt}$ that are connected at time t. So we see the defined transitions system $\mathbf{TS} = (\mathbf{Q}, \mathbf{q}_0, \Sigma, \Phi, \pi)$ is equivalent to a dynamic graph (9.5). Shortly, we will find it helpful to be able to reason about dynamic graphs with transition systems.

9.1.2. Linear temporal logic

Linear temporal logic (LTL) has the following syntax

$$\texttt{true} \quad | \quad p \quad | \quad \varphi \wedge \psi \quad | \quad \neg\varphi \quad | \quad \bigcirc\varphi \quad | \quad \varphi\mathcal{U}\psi \quad | \quad \Diamond\varphi \quad | \quad \Box\varphi.$$

There needs explanation. If p is an atomic proposition in Φ, then p is a formula. If φ and ψ are formulas, then all of the operations of propositional calculus also constitute formulas. The other symbols are interpreted: $\bigcirc\varphi$ *next time*, $\varphi\mathcal{U}\psi$ *until*, $\Diamond\varphi$ *eventually*, $\Box\varphi$ *always*. Derived operators include $\Box\Diamond\varphi$ *infinitely often* and $\Diamond\Box\varphi$ *at some point forever*. The collection of all formulas formed with this syntax is $\mathcal{L}^{LTL}(\Phi)$.

The semantics of LTL is the following. Let $\sigma \in \wp(\Phi)^\omega$ be an input word, $p \in \Phi$, and

$\varphi, \psi \in \mathcal{L}^{LTL}(\Phi)$. Then,

$$\begin{aligned}
\sigma \models \quad & p \Leftrightarrow & p \in \sigma(0), \\
\sigma \models \quad & \varphi \wedge \psi \Leftrightarrow & \sigma \models \varphi,\ \sigma \models \psi, \\
\sigma \models \quad & \neg\varphi \Leftrightarrow & \sigma \not\models \varphi, \\
\sigma \models \quad & \bigcirc\varphi \Leftrightarrow & \sigma[1] \models \varphi, \\
\sigma \models \quad & \varphi\mathcal{U}\psi \Leftrightarrow & \exists\ t_0 \geqslant 0 \quad \text{such that} \quad \sigma[t_0] \models \psi \quad \text{and} \quad \sigma[t'] \models \varphi \quad \forall 0 \leqslant t' < t_0, \\
\sigma \models \quad & \lozenge\varphi \Leftrightarrow & \exists t_0 \geqslant 0 \quad \text{such that} \quad \sigma[t_0] \models \varphi, \\
\sigma \models \quad & \square\varphi \Leftrightarrow & \sigma[t'] \models \varphi \quad \forall t' \geqslant 0.
\end{aligned}$$

For a trace $\tau \in \wp(Q)^\omega$, we write $\tau \models \varphi$ if for the cooresponding output work $\sigma_{in} \models \varphi$.

Going back to syntax, it is useful to have a notion of which words satisfy a given proposition. Given φ, the *intension* of φ is

$$[\![\varphi]\!] = \qquad \{\tau \in \wp(Q)^\omega \mid \tau \models \varphi\}$$

and the *words* satisfying φ is collected in the set $\mathrm{Words}(\varphi) = \pi\left([\![\varphi]\!]\right)$.

Given a transition system **TS** and a proposition φ there is a computation pipeline to verify that φ is valid in **TS** [9]. The process involves "translating" a formula φ into a Büchi automaton.

Definition 9.2. A *nondeterministic Büchi automaton (NBA)* is a tuple $\mathbf{B} = (S, S_0, \Sigma, \delta, F)$ with

1. S, a set of states,

2. S_0, a set of initial states,

3. Σ, a set of control inputs,

4. δ, a transition map

$$\delta : S \times \Sigma \to \wp(S),$$

5. $F \subseteq S$, a set of accepting states.

An *input word* $\sigma \in \Sigma^\omega$ is *accepted* if there is at least one corresponding sequence $\mathbf{s} = s(0)s(1)s(2)\ldots$ such $s(0) \in S_0$ and the set $\{t \mid s(t) \in F\}$ is countably infinite. The *language* of $\mathbf{B}$, denoted $\mathcal{L}(\mathbf{B})$ consists of the set of input words σ that are accepted by $\mathbf{B}$.

Theorem 9.1 (Theorem 5.37 [9]). For any LTL formula $\varphi \in \mathcal{L}^{LTL}(\Phi)$, there is a (generalized[20]) nondeterministic Büchi automaton $\mathbf{G}_\varphi$ with $\Sigma = \wp(\Phi)$ such that

$$\text{Words}(\varphi) = \mathcal{L}(\mathbf{G}_\varphi).$$

Once we have φ translated into $\mathbf{B}_\varphi$, we construct the *product Büchi automaton* (PBA) $\mathbf{P} = \mathbf{TS} \otimes \mathbf{B}_\varphi$ [9, p. 200] which can be viewed as a directed graph via (9.2). After taking a quotient of the digraph of $\mathbf{P}$ by strongly connected components, use Djistra's algorithm [38] to search for paths beginning at an initial state $(q_0, s_0) \in Q \times S$ that visit an accepting state in $Q \times F$ infinitely often. A projection of this path in the product automaton $\mathbf{P}$ onto the original transition system $\mathbf{TS}$ yields a trace $\tau \models \varphi$. We say $\mathbf{TS} \models \varphi$ if there does not exist a trace τ such that $\tau \models \neg\varphi$.

Proposition 9.1 (Complexity [9]). The complexity of the model checking problem $\mathbf{TS} \models \varphi$ is

$$\mathcal{O}\left(|\mathbf{TS}| 2^{|\varphi|}\right)$$

where $|\mathbf{TS}|$ is the number of states and transitions in $\mathbf{TS}$ and $|\varphi|$ is (roughly) the number

[20] A generalized Büchi nondeterministic automaton (GNBA) is lightly more general than a plain NBA. However, GNBA can be transformed into an NBA [9, Theorem 4.56].

of symbols in φ.

9.1.3. Distributed model checking

In the following, we consider a network $G = (\mathcal{V}_G, \mathcal{E}_G)$ of agents and a number of atomic propositions indexed by the agents $i \in \mathcal{V}_G$. In stead of model-checking a formula φ for the entire system, a strategy is to model-check *local formulas* φ_i.

Problem 9.4 (Distributed LTL Model-Checking)**.** Suppose $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a graph and $\mathbf{TS} = (Q, Q_0, \Sigma, \delta, \Phi)$ is a transition system such that Φ is indexed by $\mathcal{V}_G$ and a (possibly empty) set $\mathcal{A}$ not dependent on the graph G.

$$\Phi = \qquad \{p^i_A\}_{i \in \mathcal{V}_G, A \in \mathcal{A}}.$$

Let $\Phi_i = \{p^j_A\}_{j \in \mathcal{N}_i \cup i, A \in \mathcal{A}}$ be the set of atomic propositions supported on the the neighborhood of $i \in \mathcal{V}_G$. Let

$$\varphi = \qquad \bigwedge_{i \in \mathcal{V}_G} \varphi_i$$

where $\varphi_i \in \mathcal{L}(\Phi_i)$. Given $\sigma \in \Sigma^\omega$, decide whether $\sigma \models \varphi$.

We include some examples of distributive model-checking problems.

Example 9.3 (Intermittent Connectivity)**.** The following example is taken from [88]. Suppose robots with limited communication (e.g. submarine robots) each travel in between two rendezvous locations $\mathbf{v}_i$ and $\mathbf{v}_j$ in order to commuicate with other agents. Then, let $G = (\mathcal{V}_G, \mathcal{E}_G)$ be the graph with nodes indexing rendevous locations $\mathbf{v}_{ij}$ and edges indexing robots r_{ij}. This intermittent connectivity problem is a distributed model-checking problem with

$$\varphi_i = \qquad \Box\Diamond \bigwedge_{j \in \mathcal{N}_i} p^{ij}_{\mathbf{v}_i}$$

where $p^{ij}_{\mathbf{v}_i}$ is the proposition:

> *Robot r_{ij} is located in the region $\mathcal{R}_i = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_{ij}\| \leqslant \varepsilon\}$.*

The local formulas φ_i specify:

> *All robot assigned to the rendevous point $\mathbf{v}_i$ are simultaneously located in the region $\mathcal{R}_i$ infinitely often.*

Example 9.4 (Epidemics)**.** Let $\mathcal{A} = \{S, I, R\}$ represent the status *susceptible*, *infected*, and *recovered*. Suppose $\Phi = \{p^i_{A \in \mathcal{A}}\}$. For instance, p^i_I is the proposition *agent i is infected*. The popular susceptible-infected-recovered (SIR) model can be viewed as a distributed model-checking problem with

$$\varphi_i = \qquad \Box \left(\bigvee_{j \in \mathcal{N}_i} p^i_S \Rightarrow \Diamond p^j_I \right) \wedge \Box \left(p^i_I \Rightarrow \Diamond p^i_R \right).$$

We interpret φ_i:

> *If at least one neighbor of agent i is infected, then agent i will eventually be infected; if agent i is infected, she will eventually recover.*

Of course, there are similar models such as susceptible-infected-susceptible (SIS), susceptible-infected-recovered-diseased (SIRD). Note that epidemic models can be used for modeling rumors as well as physical diseases.

Example 9.5 (Non-Interfering Connectivity [135])**.** Consider an arbitrary undirected graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ and the transition system **TS** defined in Example 9.2 modeling network connectivity. Recall, $p^{ij}, ij \in \mathcal{E}_G$ is the proposition

> *Communication link ij is active.*

Consider the local formula

$$\varphi_i = \bigwedge_{j \in \mathcal{N}_i} \varphi_{ij}$$

where φ_{ij} is the formula

$$\varphi_{ij} = \Box \left(p^{ij} \Rightarrow \bigvee_{i'j' \in \mathcal{N}_{ij}} \neg p^{i'j'} \right) \wedge \Box \Diamond p^{ij}. \tag{9.6}$$

φ_{ij} can be interpreted:

> *If communication link ij is active, then every adjacent communication link must be off. Moreover, every communication link must become active infinitely often.*

We proposed [135] a partial solution to the distributed model-checking problem, which applies to the above examples.

1. Select a subset $\mathcal{V}_C \subseteq \mathcal{V}_G$ of nodes which we call *command nodes.*

2. After selecting a global proximity radius $R \in \mathbb{N}$ and a local proximity radius $r \leqslant \lfloor R/2 \rfloor$, form the *command graph* with nodes $\mathcal{V}_C$ and $ij \in \mathcal{E}_C$ if and only if i and j are connected by a shortest path of length no greater than r in G. Construct a family of subgraphs $G_i, i \in \mathcal{V}_C$ induced by the node sets

   $$\mathcal{V}_{G_i} = \{j \in \mathcal{V}_G \mid d(i,j) \leqslant r\};$$

   G_i is the smallest subgraph of G containing $\mathcal{V}_{G_i}$.

For $i \in \mathcal{V}_C$, let $\mathbf{TS}_i$ be the subtransaction system of $\mathbf{TS}$ (Problem 9.4) restricted to G_i. Model checking each $\mathbf{TS}_i \models \bigwedge_{j \in \mathcal{V}_{G_i}} \varphi_j,\ i \in \mathcal{V}_C$ is now substantially less expensive than model-checking $\mathbf{TS} \models \varphi$.

9.2. Muli-Agent Logic

For more details, we suggest the reader consult the standard reference [52]. For the following, suppose a system has agents labeled $\mathcal{V}_G = \{1, 2, \dots, n\}$. In multi-agent logic, we develop a mathematical language to reason about information held by individuals and groups of individuals in the system. Information, in our setting, consists of formulas and their truth values, contrasting the perspective of Chapter 8 which views information as signals, boxes of numbers. In temporal logic, there are two modalities $\Box$ and $\Diamond$. In multi-agent logic, we have a modality for every agent $K_1, K_2, \dots, K_n$.

Definition 9.3 (Kripke Model)**.** An *Kripke frame* consists of a set S of *states* with binary relations

$$\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_n.$$

A *Kripke model*, denoted M, is a frame $(S, \mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_n)$ together with an *evaluation*

$$\pi : S \to \wp(\Phi).$$

It is obvious, by currying, that evaluations are in bijection with relations $\models$ between S and Φ which we call *interpretations*; we liberally go back and forth between evaluations and interpretations

$$(M, s) \models p \quad \Leftrightarrow \qquad\qquad p \in \pi(s).$$

Multi-agent logic distinguishes itself from ordinary propositional logic with the set of relations $\mathcal{K}_i$ that encode the semantics for syntactic modal operators written K_i. Together with the connectives of propositional logic, the entire syntax of *multi-agent logic* is given by the following

$$\texttt{true} \quad | \quad p \quad | \quad \varphi \wedge \psi \quad | \quad \neg\varphi \quad | \quad i \quad | \quad K_i\varphi.$$

The set of all sentences formed via the above syntax over the atomic propositions Φ is called the language $\mathcal{L}_n(\Phi)$. The set of all n-agent Kripke models over Φ is denoted $\mathcal{M}_n(\Phi)$. We can *interpret* a $\varphi \in \mathcal{L}_n(\Phi)$ with the following inductive rules:

$$(M, s) \models \quad p \quad \Leftrightarrow \quad p \in \pi(s) \tag{9.7}$$

$$(M, s) \models \quad \varphi \wedge \psi \quad \Leftrightarrow \quad (M, s) \models \varphi, (M, s) \models \psi \tag{9.8}$$

$$(M, s) \models \quad \neg\varphi \quad \Leftrightarrow \quad (M, s) \not\models \varphi \tag{9.9}$$

$$(M, s) \models \quad K_i\varphi \quad \Leftrightarrow \quad (M, t) \models \varphi \quad \forall t \in S \quad \text{such that} \quad s\mathcal{K}_i t \tag{9.10}$$

It is the last rule that gives a Kripke model the nickname, a “possible worlds” model because $K_i\varphi$ is satisfied at a state if every neighboring state t with $s\ \mathcal{K}_i t$ satisfies φ. As suggested by notation, the recommended mental model for the formula $K_i\varphi$ is *agent i knows φ*. We will see shortly why this is just a “mental model.” The modal logic $\mathcal{L}_n(\Phi)$ could then be coined a *epistemic logic*. We write $M \models \varphi$ if $(M, s) \models \varphi$ for all $s \in S$. In this case, we would say φ is *valid* in M.

Derived from the basic syntax and semantics, are additional logical operators. Suppose $A \subseteq \mathcal{V}_G$ is a subset of agents. Then,

$$E_A\varphi = \bigwedge_{i \in A} K_i\varphi$$

which in our mental model specifies that *everyone in A knows φ*. Define

$$C_A\varphi = \bigwedge_{k \geqslant 1} E_A^k\varphi$$

which in our mental model specifies that φ is *common knowledge*, or, in other words, everyone knows that everyone knows that everyone knows φ *ad infinitum.* Note that E_G is not in the language $\mathcal{L}_n(\Phi)$. To include common knowledge, we will need to augment our language to include the symbols C_G. Finally, let D_A be the operator with semantics

$$(M, s) \models \qquad D_A\varphi \quad \Leftrightarrow \qquad (M, t) \models \varphi \quad (\forall t \in S) \quad (s, t) \in \bigcap_{i \in A} \mathcal{K}_i.$$

In our mental mode, D_A supplies a notion of *distibutive knowledge* with the intuition that the "combined" knowledge of all the agents in the group $A \subseteq \mathcal{V}_G$ implies φ. We write $\mathcal{L}_n^C$, $\mathcal{L}_n^D$, $\mathcal{L}_n^{CD}$ for the augmented languages with common knowledge, distributive knowledge, and both common and distributive knowledge, respectively.

For the time being, we will narrow our focus on $\mathcal{L}_n$. Let us briefly describe this logic axiomatically. An axiom system consists of axioms and inference rules. An axiom is a valid formula while an inference rule is a reduction of a set of valid formulas in a context to another formula. The axiom system $\mathsf{K}_n(\Phi)$ consists of the following rules and axioms:[21]

(A1) Tautologies of propositional calculus.

(A2) $(K_i\varphi \wedge K_i(\varphi \wedge \psi)) \Rightarrow K_i\psi \quad \forall i \in \mathcal{V}_G$ (Distribution Axiom)

(R1) From φ and $\varphi \Rightarrow \psi$ infer ψ (Modus Potens)

(R2) From φ infer $K_i\varphi$ (Knowledge Generalization)

The following theorem guarantees, given Φ, every formula provable in K_n is valid in $\mathcal{M}_n$ and conversely every formula valid in $\mathcal{M}_n$ is provable in K_n.

Theorem 9.2 (Theorem 3.1.3 [52])**.** The axiom system $\mathsf{K}_n(\Phi)$ is sound and complete axiomatization of the language $\mathcal{L}_n(\Phi)$ with respect to $\mathcal{M}_n(\Phi)$.

Additional axioms distinguishing knowledge over belief, result in alternative axiom systems

[21] We label axioms and inference rules per Fagin et. al. for convenience [52].

which correspond semantically to subclasses of $\mathcal{M}_n$ [52].

(A3) $K_i\varphi \Rightarrow \varphi$ (Knowledge Axiom)

(A4) $K_i\varphi \Rightarrow K_iK_i\varphi$ (Positive Introspection)

(A5) $\neg K_i\varphi \Rightarrow K_i\neg K_i\varphi$ (Negative Introspection)

(A6) $\neg K_i(\texttt{false})$ (Consistency)

(A7) $\varphi \Rightarrow K_i\neg K_i\neg\varphi$ (Monadicity)

We will not say much more about the resulting correspondences between classes of Kripke models and axiom systems except that axiom systems with more axioms than K_n restrict the class of allowable Kripke relation $\mathcal{K}_i$. For instance, the axiom system called $\mathsf{S5}_n$ (K_5 plus A3, A4, A5) restricts $\mathcal{M}_n$ to the class of models $\mathcal{M}_n^{rst}$ whose Kripke relations $\mathcal{K}_i$ are equivalence relations. A case can be made, thus, that $\mathcal{M}_n^{rst}$ models knowledge. Hence, we interpret the formula $K_i\varphi$ as *agent i knows φ*.

In another instance, the axiom system called $\mathsf{KD45}_n$ (K_5 plus A4, A5, A6) restricts $\mathcal{M}_n$ to the class of models $\mathcal{M}_n^{elt}$ whose Kripke relations $\mathcal{K}_i$ are Euclidean, serial and transitive. $\mathsf{KD45}_n$ is reasonably interpreted as modeling belief with K_i interpreted as *agent i believes* φ and, of special note, $\neg K_i\neg\varphi$ as *agent i doesn't disbelieve φ*. With this interpretation in mind, notice the absence of A3 in $\mathsf{KD45}_n$; believing something does not make it true.

9.2.1. Kripke-Galois connections

Recall from our discussion of temporal logic, intension allows you to pass from syntax and semantics. In Kripke semantics, the *intension* of a formula describes the semantic content of that formula, the subset of states for which a given formula is true. In general, we refer to a subset $e \subseteq \wp(S)$ as an *event*.

Definition 9.4 (Intension)**.** Suppose M is a Kripke model and $\varphi \in \mathcal{L}_n(\Phi)$ a formula. Then,

$$[\![\varphi]\!] = \{s \in S \mid (M, s) \models \varphi\}$$

is the *intension* of φ. We say φ has more *semantic content* than ψ if $[\![\varphi]\!] \supseteq [\![\psi]\!]$. We say two formulae $\varphi, \psi \in \mathcal{L}_n(\Phi)$ are *semantically equivalent* if $[\![\varphi]\!] = [\![\psi]\!]$ and write $\varphi \equiv \psi$.

Proposition 9.2. Given a model M, semantic equivalence is an equivalence relation on $\mathcal{L}_n(\Phi)$.

Let $\mathcal{L}_n(\Phi)/\simeq$ denote the set of equivalence classes of formulas where individual classes are denoted $[\varphi]$. Semantic equivalence is not to say that two formulas φ and ψ are the same, but that they are the same in the context M. We now begin an investigation of how syntax and semantics interact in multi-agent logic. Let $(-)^c$ denote complement in the lattice $\wp(S)$.

Lemma 9.1. Suppose $\varphi, \psi \in \mathcal{L}_n(\Phi)$ and $M = (S, K_1, \ldots, K_n, \Phi, \pi)$. Then,

$$\begin{aligned} [\![\varphi \wedge \psi]\!] &= [\![\varphi]\!] \cap [\![\psi]\!] \\ [\![\neg\varphi]\!] &= [\![\varphi]\!]^c \\ [\![\texttt{true}]\!] &= S \end{aligned}$$

Proof. Exercise. □

Of course, other identities relating the syntax of propositional logic and the structure of the lattice $\wp(S)$ are easily shown. However, lattice-theoretic interpretations of Kripke modalities are much less trivial. In the following theorem, we establish a Galois connection on $\wp(S)$ which "respects" semantic equivalence.

Theorem 9.3 (Kripke-Galois Connection)**.** Suppose M is a Kripke model $M = S, \mathcal{K}_1, \ldots, \mathcal{K}_n, \Phi, \pi)$, Then, there is a family of Galois connections indexed by $i \in \{1, 2, \ldots, n\}$

$$\begin{tikzcd}
\wp(S) \arrow[r, bend left, "\mathcal{K}_i^{\exists}"] & \wp(S) \arrow[l, bend left, "\mathcal{K}_i^{\forall}"]
\end{tikzcd}$$

given by

$$\begin{aligned}
\mathcal{K}_i^{\exists}(e) = & \quad \{t \in S \mid (\exists s)\,(s \in e \wedge s\mathcal{K}_i t)\} \qquad (9.11)\\
\mathcal{K}_i^{\forall}(e) = & \quad \{s \in S \mid (\forall t)\,(s\mathcal{K}_i t \Rightarrow t \in e)\}
\end{aligned}$$

such that

$$\begin{aligned}
\mathcal{K}_i^{\forall}(\llbracket\varphi\rrbracket) = & \quad \llbracket K_i\varphi\rrbracket \qquad (9.12)\\
\mathcal{K}_i^{\exists}(\llbracket\varphi\rrbracket) = & \quad \llbracket\neg K_i\neg\varphi\rrbracket.
\end{aligned}$$

for all $\varphi \in \mathcal{L}_n(\Phi)$.

Proof. By Theorem 3.2, the pair $\left(\mathcal{K}_i^{\exists}, \mathcal{K}_i^{\forall}\right)$ is a Galois connection. It remains to show (9.12). First,

$$\begin{aligned}
\mathcal{K}_i^{\forall}(\llbracket\varphi\rrbracket) = & \\
= & \quad \{s \in S \mid (\forall t) \quad s\mathcal{K}_i t \Rightarrow t \models \varphi\}\\
= & \quad \llbracket K_i\varphi\rrbracket.
\end{aligned}$$

Second,

$$
\begin{aligned}
\mathcal{K}_i^{\exists}([\![\varphi]\!]) = & \\
= & \quad \{t \in S \mid (\exists s) \quad s \models \varphi \ \wedge \ s\mathcal{K}_i t\} \\
= & \quad \{t \in S \mid \neg\left((\forall s) \quad s \not\models \varphi \vee \neg(s\mathcal{K}_i t)\right)\} \\
= & \quad \{t \in S \mid \neg\left((\forall s) \quad s\mathcal{K}_i t \Rightarrow s \not\models \varphi\right)\} \\
= & \quad \{t \in S \mid \neg\left((\forall s) \quad s\mathcal{K}_i t \Rightarrow s \in \neg\varphi\right)\} \\
= & \quad S - \{t \in S \mid (\forall s) \quad s\mathcal{K}_i t \Rightarrow s \in \neg\varphi\} \\
= & \quad S - [\![K_i\neg\varphi]\!] \\
= & \quad [\![\neg K_i\neg\varphi]\!]
\end{aligned}
$$

where the last equality is by Lemma 9.1. □

Corollary 9.1. The semantic content of $K_i\neg K_i\neg\varphi$ is greater than the semantic content of φ, whereas, the semantic content of $\neg K_i\neg K_i\varphi$ is less than the semantic content of φ.

Proof. By Proposition 3.1,

$$
\begin{aligned}
\mathcal{K}_i^{\forall}\mathcal{K}_i^{\exists}([\![\varphi]\!]) \supseteq & \quad [\![\varphi]\!] \\
\mathcal{K}_i^{\exists}\mathcal{K}_i^{\forall}([\![\varphi]\!]) \subseteq & \quad [\![\varphi]\!].
\end{aligned}
$$

Furthermore,

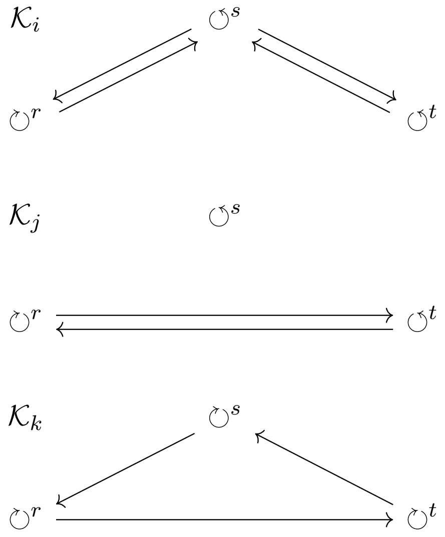
$$
\begin{aligned}
\mathcal{K}_i^{\forall}\mathcal{K}_i^{\exists}([\![\varphi]\!]) = & \quad [\![K_i\neg K_i\neg\varphi]\!] \\
\mathcal{K}_i^{\exists}\mathcal{K}_i^{\forall}([\![\varphi]\!]) = & \quad [\![\neg K_i\neg K_i\varphi]\!]
\end{aligned}
$$

by Lemma 9.1. □

Thus, applying $\mathcal{K}_i^{\forall}\mathcal{K}_i^{\exists}$ or $\mathcal{K}_i^{\exists}\mathcal{K}_i^{\forall}$ increases or decreases the information content, respectively.

It may not be obvious, but the reader may check for herself than $\mathcal{K}_i^{\forall}\mathcal{K}_j^{\exists}$ in general neither increases or decreases semantic content.

Example 9.6 (Alice, Bob & Eve)**.** Consider a network G with three agents (Alice, Bob and Eve) represented by nodes i, j, k respectively. Suppose Alice and Bob can communicate $ij \in \mathcal{E}_G$, Bob and Eve can communicate $jk \in \mathcal{E}_G$, but not Alice and Eve $ik \notin \mathcal{E}_G$. There are three states in our system $S = \{r, s, t\}$. The Kripke relations $\mathcal{K}_i, \mathcal{K}_j, \mathcal{K}_k$ for Alice, Bob and Eve are depicted by the following directed graphs.



Each relation induces a Galois connection, summarized in the following table.

e	$\mathcal{K}_i^{\exists}(e)$	$\mathcal{K}_i^{\forall}(e)$	$\mathcal{K}_j^{\exists}(e)$	$\mathcal{K}_j^{\forall}(e)$	$\mathcal{K}_k^{\exists}(e)$	$\mathcal{K}_k^{\forall}(e)$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\{r\}$	$\{r, s\}$	$\emptyset$	$\{r, t\}$	$\emptyset$	$\{r, t\}$	$\emptyset$
$\{s\}$	$\{r, s, t\}$	$\emptyset$	$\{s\}$	$\{s\}$	$\{r, s\}$	$\emptyset$
$\{t\}$	$\{s, t\}$	$\emptyset$	$\{r, t\}$	$\emptyset$	$\{s, t\}$	$\emptyset$
$\{r, s\}$	$\{r, s, t\}$	$\{r\}$	$\{r, s, t\}$	$\{s\}$	$\{r, s, t\}$	$\{s\}$
$\{r, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, t\}$	$\{r, t\}$	$\{r, s, t\}$	$\{r\}$
$\{s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{t\}$
$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$

9.2.2. Kripke sheaves

We now introduce a class of lattice-valued network sheaves and cosheaves whose stalks are powersets representing the semantic content of logical propositions and whose restriction and corestriction maps are Kripke-Galois connections.

Definition 9.5 (Kripke Sheaves)**.** Suppose G is a graph with nodes labeled $\mathcal{V}_G = \{1, 2, \ldots, n\}$. Suppose $M = (S, \mathcal{K}_1, \ldots, \mathcal{K}_n, \Phi, \pi)$ is a Kripke model. A *Kripke bisheaf* is a Tarski sheaf $\mathcal{F}_M$ characterized by the following.

1. Each stalk $\mathcal{F}_M(i)$ is the powerset $\wp(S)$.

2. Each stalk $\mathcal{F}_M(ij)$ is the powerset $\wp(S)$.

3. $\underline{\mathcal{F}}_M(i \trianglelefteq ij) = \mathcal{K}_i^{\exists}$ for all $i \in \mathcal{V}_G, j \in \mathcal{N}_i$.

4. $\overline{\mathcal{F}}_M(i \trianglelefteq ij) = \mathcal{K}_i^{\forall}$ for all $i \in \mathcal{V}_G, j \in \mathcal{N}_i$.

The 0-cochains of $\underline{\mathcal{F}}_M$ are the product $C^0(G; \underline{\mathcal{F}}_M) = \wp(S)^n$. For cochains, thereby passing from syntax to semantics, let

$$[\![\boldsymbol{\varphi}]\!] = \qquad\qquad ([\![\varphi_i]\!])_{i \in \mathcal{V}_G} \in \wp(S)^n$$

for a tuple of formulas $\boldsymbol{\varphi} \in \mathcal{L}_n(\Phi)^n$. Let's consider some special cases.

- Suppose $\mathcal{K}_i$ is the identity relation on S; $s\mathcal{K}_i s$ for all $s \in S$, no other elements are in $\mathcal{K}_i$. Then, $(\mathcal{K}_i^{\exists}, \mathcal{K}_i^{\forall}) = (\mathrm{id}, \mathrm{id})$ and $\underline{\mathcal{F}}_M$ is the constant sheaf $\underline{\wp(S)}$. Note, in this case, $[\![\varphi]\!] = [\![K_i\varphi]\!]$ so that the constant sheaf reduces a network sheaf model of the semantics for ordinary propositional logic.

- Suppose $\mathcal{K}_i$ is a (possibly different) equivalence relation for each i, so that $\mathcal{M}_n^{rst}$ is a class of models for $\mathsf{S5}_n$. The restriction map $\mathcal{K}_i^{\exists}$ sends an event e to the union of equivalence classes of elements of e. In particular, $\mathcal{K}_{\exists}$ sends equivalence classes to equivalence classes. The corestriction map $\mathcal{K}_i^{\forall}$ sends an event e to the set of elements of S whose equivalence class is contained in e.

As our notation suggests, 0-cochains model tuples of formulas through their intension. We investigate the sections of this class of sheaves.

Proposition 9.3. Suppose $\mathcal{F}_M$ is a Kripke sheaf with the above data. Suppose $\boldsymbol{\varphi} \in \mathcal{L}_n(\Phi)^n$ is a tuple of formulas indexed by the nodes of G. Then, $[\![\boldsymbol{\varphi}]\!] \in H^0(G; \underline{\mathcal{F}}_M)$ if and only if

$$K_i \neg \varphi_i \equiv K_j \neg \varphi_j \quad \forall ij \in \mathcal{E}_G.$$

Proof. Recall, by the global sections criterion, $\mathbf{e} \in \Gamma\,(G; \underline{\mathcal{F}}_M)$ if and only

$$\underline{\mathcal{F}}_M(i \trianglelefteq ij)(e_i) = \underline{\mathcal{F}}_M(j \trianglelefteq ij) \quad \forall ij \in \mathcal{E}_G.$$

Here, with e_i defined to equal $[\![\varphi_i]\!]$ for some formula $\varphi_i \in \mathcal{L}_n(\Phi)$, so that

$$\mathcal{K}_i^{\exists}([\![\varphi_i]\!]) = \mathcal{K}_j^{\exists}([\![\varphi_j]\!]) \quad \forall ij \in \mathcal{E}_G.$$

By Lemma 9.1 and Theorem 9.3,

$$\begin{aligned} \mathcal{K}_i^{\exists}([\![\varphi_i]\!]) &= [\![\neg K_i \neg \varphi_i]\!] \\ &= [\![K_i \neg \varphi_i]\!]^c \\ &= \mathcal{K}_i^{\forall}\left([\![\neg \varphi_i]\!]\right)^c \\ &= \mathcal{K}_i^{\forall}\left([\![\varphi_i]\!]^c\right)^c \end{aligned}$$

This implies,

$$\mathcal{K}_i^{\forall}\left([\![\varphi_i]\!]^c\right)^c = \mathcal{K}_j^{\forall}\left([\![\varphi_j]\!]^c\right)^c,$$

which, taking complements of both sides and applying $\llbracket\varphi\rrbracket^c = \neg\llbracket\varphi\rrbracket$ again, yields

$$\mathcal{K}_i^{\forall}\left(\llbracket\neg\varphi_i\rrbracket\right) = \qquad \mathcal{K}_j^{\forall}\left(\llbracket\neg\varphi_j\rrbracket\right).$$

Hence,

$$\llbracket K_i\neg\varphi_i\rrbracket = \qquad \llbracket K_j\neg\varphi_j\rrbracket.$$

□

The proof of Proposition 9.3 suggest that under certain conditions, $K_i\varphi_i$ is semantically equivalent to $K_j\varphi_j$ as well.

Proposition 9.4. Suppose, in addition to the above assumptions, that $\mathcal{K}_i$ is serial for all $i \in \mathcal{V}_G$. Then, $\llbracket\varphi\rrbracket \in H^0(G;\underline{\mathcal{F}}_M)$ if and only if

$$K_i\varphi_i \equiv K_j\varphi_j.$$

We will need a lemma.

Lemma 9.2. Suppose $\mathcal{K}_i$ is serial. Then, $\mathcal{K}_i^{\forall}(\emptyset) = \emptyset$.

Proof of Lemma 9.2. By Proposition 3.1 and Theorem 9.3,

$$\mathcal{K}_{\forall}(\emptyset) = \qquad \bigcup\{e \subseteq S \mid \mathcal{K}_i^{\exists}(e) = \emptyset\}.$$

Because $\mathcal{K}_i$ is serial, for every $s \in S$, there is at least one $t \in S$ with $s\mathcal{K}_i t$. Hence, $\mathcal{K}_i^{\exists}(e) = \emptyset$ if and only if $e = \emptyset$. Thus, $\mathcal{K}_{\forall}(\emptyset) = \emptyset$. □

Proof of Proposition 9.4. By Proposition 9.3, $\llbracket\boldsymbol{\varphi}\rrbracket$ is a section if and only if $\llbracket K_i\neg\varphi_i\rrbracket = \llbracket K_j\neg\varphi_j\rrbracket$ for all $ij \in \mathcal{E}_G$, which holds if and only if $\llbracket K_i\varphi_i\rrbracket = \llbracket K_j\varphi_j\rrbracket$ for all $ij \in \mathcal{E}_G$ by the following.

Claim. $\mathcal{K}_i^{\forall}(\llbracket\varphi_i\rrbracket^c) = \mathcal{K}_i^{\forall}(\llbracket\varphi\rrbracket)^c$.

By Proposition 3.1 (again) and Theorem 9.3,

$$\mathcal{K}_i^{\forall}\left(\llbracket\varphi_i\rrbracket \cap \llbracket\varphi_i\rrbracket^c\right) = \mathcal{K}_i^{\forall}\left(\llbracket\varphi_i\rrbracket\right) \cap \mathcal{K}_i^{\forall}\left(\llbracket\varphi_i\rrbracket^c\right).$$

On the other hand,

$$\begin{aligned} \mathcal{K}_i^{\forall}\left(\llbracket\varphi_i\rrbracket \cap \llbracket\varphi_i\rrbracket^c\right) &= \mathcal{K}_i^{\forall}(\emptyset) \\ &= \emptyset \end{aligned}$$

Hence,

$$\mathcal{K}_i^{\forall}\left(\llbracket\varphi_i\rrbracket^c\right) = \mathcal{K}_i^{\forall}\left(\llbracket\varphi_i\rrbracket\right)^c,$$

or,

$$\llbracket K_i \neg\varphi_i\rrbracket = \llbracket K_i\varphi_i\rrbracket^c. \tag{9.13}$$

Finally, $\llbracket K_i\neg\varphi_i\rrbracket = \llbracket K_j\neg\varphi_j\rrbracket$ is equivalent to $\llbracket K_i\varphi_i\rrbracket^c = \llbracket K_j\varphi_j\rrbracket^c$ which is of course equivalent to $\llbracket K_i\varphi_i\rrbracket = \llbracket K_j\varphi_j\rrbracket$ by taking set complements of both sides. □

Proposition 9.3 and Proposiotin 9.4 suggest we can interpret global sections of Kripke sheaves as local *knowledge consensus*. We say $\mathcal{F}$ is *star-shaped* if for all $i \in \mathcal{V}_G$ and for all $j, j' \in \mathcal{N}_i$, $\underline{\mathcal{F}}(i \trianglelefteq ij) = \underline{\mathcal{F}}(i \trianglelefteq ij')$. However, by the following Lemma, local knowledge consensus necessarily extends to global knowledge consensus.

Lemma 9.3. Suppose $\mathcal{F}$ is a star-shaped Tarski sheaf over a G, and suppose $\mathbf{x} \in \Gamma\left(G; \underline{\mathcal{F}}\right)$. Then, $x_{ij} = x_{i'j'}$ for every pair of edges $ij, i'j' \in \mathcal{E}_G$ in the same connected component of G.

Proof. By the star-shaped property,

$$\underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) = \qquad \underline{\mathcal{F}}(i \trianglelefteq ij')(x_i)$$

for all $i \in \mathcal{V}_G$, $j, j' \in \mathcal{N}_i$. Yet, because $\mathbf{x} \in \Gamma\left(G; \underline{\mathcal{F}}\right)$,

$$\begin{aligned} \underline{\mathcal{F}}(j \trianglelefteq ij)(x_j) &= \qquad \underline{\mathcal{F}}(i \trianglelefteq ij)(x_i) \\ \underline{\mathcal{F}}(j' \trianglelefteq ij')(x_{j'}) &= \qquad \underline{\mathcal{F}}(i \trianglelefteq ij')(x_i) \end{aligned}$$

which implies $x_{ij} = x_{ij'}$. By connectivity, this implies sections of $\underline{\mathcal{F}}$ are constant when projected onto $C^1(G; \underline{\mathcal{F}})$. □

All the preceding results lead up to the following theorem.

Theorem 9.4. Suppose $M = (S, \mathcal{K}_1, \ldots, \mathcal{K}_n, \Phi, \pi)$ is a Kripke model supported on a graph $G = (\mathcal{V}_V, \mathcal{E}_E)$ such that every $\mathcal{K}_i$ is serial and $\mathcal{F}_M$ its corresponding Kripke bisheaf. Then, $[\![\boldsymbol{\varphi}]\!] \in H^0(G; \underline{\mathcal{F}}_M)$ if and only if for every $i, j \in \mathcal{V}_G$ in the same connected component

$$K_i \varphi_i \equiv K_j \varphi_j.$$

Corollary 9.2. In particular, if $\boldsymbol{\varphi}$ is constant with value φ, then

$$[\![K_i \varphi]\!] = \qquad [\![K_j \varphi]\!]$$

for all $i, j \in \mathcal{V}_G$ in the same connected component.

9.2.3. Kripke Laplacian

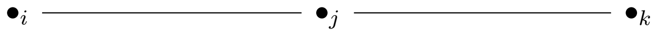
Our focus will now shift to semantic dynamics over a network. Diffusion dynamics on the 0-cochains $\wp(S)^n$ are defined by means of the Tarski Laplacian of $\underline{\mathcal{F}}_M$.

Definition 9.6 (Kripke Laplacian)**.** Suppose $M = (S, \mathcal{K}_1, \ldots, \mathcal{K}_n, \Phi, \pi)$ is Kripke model and G is a graph with nodes $\mathcal{V}_G = \{1, 2, \ldots, n\}$ as before. Then, the *Kripke Laplacian* is the operator

$$\begin{aligned} L : \qquad & \wp(S)^n \to \wp(S)^n \\ (L\mathbf{e})_i = \qquad & \bigcap_{j \in \mathcal{N}_i} \mathcal{K}_i^{\forall} \mathcal{K}_i^{\exists}(e_i) \end{aligned} \tag{9.14}$$

Example 9.7 (Alice, Bob & Eve)**.** Let us revisit the network G, frame $(S, \mathcal{K}_i, \mathcal{K}_j, \mathcal{K}_j)$, and Galois connections $(\mathcal{K}_i^{\exists}, \mathcal{K}_i^{\forall}), (\mathcal{K}_j^{\exists}, \mathcal{K}_j^{\forall}), (\mathcal{K}_k^{\exists}, \mathcal{K}_k^{\forall})$ from Example 9.6. Then, the semantic sheaf $\mathcal{F}_M$ is depicted by the following diagram

$$\begin{tikzcd}
\wp(S) \arrow[r, shift left, "\mathcal{K}_i^{\exists}"] & \wp(S) \arrow[l, shift left, "\mathcal{K}_i^{\forall}"] \arrow[r, shift right, "\mathcal{K}_j^{\forall}"'] & \wp(S) \arrow[l, shift right, "\mathcal{K}_j^{\exists}"'] \arrow[r, shift left, "\mathcal{K}_j^{\exists}"] & \wp(S) \arrow[l, shift left, "\mathcal{K}_j^{\forall}"] \arrow[r, shift right, "\mathcal{K}_k^{\forall}"'] & \wp(S) \arrow[l, shift right, "\mathcal{K}_k^{\exists}"']
\end{tikzcd}$$



The Kripke Laplacian is the operator

$$\begin{aligned} & L(e_i, e_j, e_k) \\ = \qquad & \left(\mathcal{K}_i^{\forall}\mathcal{K}_j^{\exists}(e_j), \mathcal{K}_j^{\forall}\mathcal{K}_i^{\exists}(e_i) \cap \mathcal{K}_j^{\forall}\mathcal{K}_k^{\exists}(e_k), \mathcal{K}_k^{\forall}\mathcal{K}_j^{\exists}(e_j) \right) \end{aligned}$$

We compute the value of the Kripke Laplacian for a few initial 0-cochains.

e_i	e_j	e_k	$(L\mathbf{e})_i$	$(L\mathbf{e})_j$	$(L\mathbf{e})_k$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\{r\}$	$\{s\}$	$\{t\}$	$\emptyset$	$\{s\}$	$\emptyset$
$\{s\}$	$\{r\}$	$\{t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r\}$
$\{t\}$	$\{t\}$	$\{t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r\}$
$\{s\}$	$\{s\}$	$\{s\}$	$\emptyset$	$\{s\}$	$\emptyset$
$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$	$\{r, s, t\}$

We can now apply the theory built up in Chapter 6 and 7.

Theorem 9.5. Given the data M and G, the global sections $\Gamma\left(G;\underline{\mathcal{F}_M}\right)$ form a complete quasi-sublattice of $\wp(S)^n$.

Proof. Apply Theorem 6.2. □

We argue that the lattice $\Gamma\left(G;\underline{\mathcal{F}_M}\right)$ is an algebraic model for the pair (M,G). Suppose $\mathcal{K}_i$ is the identity relation for every $i \in \mathcal{V}_G$. Then, $\Gamma\left(G;\underline{\mathcal{F}_M}\right)$ is isomorphic to $\wp(S)$. This makes sense because in this case, modalities $\mathcal{K}_i$ act as the identity, which reduces the semantics to propositional calculus, a Boolean algebra.

Theorem 9.6. Suppose L is the Kripke Laplacian of $(M,)$. Then,

$$\mathrm{Post}(L) = \qquad\qquad \Gamma\left(G;\underline{\mathcal{F}_M}\right).$$

Proof. Apply Theorem 7.1. □

Theorem 9.7. The global sections coincide with the global time-invariant solutions of

$$e_i[t+1] = \qquad\qquad (\boldsymbol{L}\mathbf{e})_i \cap e_i \quad \forall i \in \mathcal{V}_G.$$

Proof. Powersets satisfy the descending chain condition. Apply Propositon 7.5. □

In Appendix B, we run an experiment by randomly generating a network and local Kripke relations. We compute heat flow using the gossip algorithm (Algorithm 3).

9.2.4. Diffusive knowledge

As before, suppose G is a graph and Φ is a set of atomic propositions.

Definition 9.7. Suppose $\varphi \in \mathcal{L}_n(\Phi)$. Then, for $i \in \mathcal{V}_V$, the *syntactic Laplacian* is the formula in $\mathcal{L}_n(\Phi)$ defined

$$\left(\tilde{L}\boldsymbol{\varphi}\right)_i = \bigwedge_{j\in\mathcal{N}_i} K_j \neg K_i \neg \varphi.$$

Lemma 9.4. Suppose $\boldsymbol{\varphi} \in \mathcal{L}_n(\Phi)^n$ and $M \in \mathcal{M}_n(\Phi)$. Then,

$$[\![(L\varphi)_i]\!] = (L[\![\boldsymbol{\varphi}]\!])_i\,.$$

Proof. By Lemma 9.1 and Theorem 9.3,

$$\begin{aligned}
[\![L_i\varphi_i]\!] &= \\
&= [\![\bigwedge_{j\in\mathcal{N}_i} K_i\neg K_j\neg\varphi_j]\!] \\
&= \bigwedge_{j\in\mathcal{N}_i} [\![K_i\neg K_j\neg\varphi_j]\!] \\
&= \bigwedge_{j\in\mathcal{N}_i} \mathcal{K}_i^{\forall}\left([\![\neg K_j\neg\varphi_j]\!]\right) \\
&= \bigwedge_{j\in\mathcal{N}_i} \mathcal{K}_i^{\forall}\mathcal{K}_j^{\forall}[\![\varphi_j]\!].
\end{aligned}$$

□

Suppose $\boldsymbol{\varphi}$ is constant with value φ. Then, it makes sense to define $L_i\varphi$ to be the formula

$$\bigwedge_{j\in\mathcal{N}_j} \neg K_j\neg\varphi.$$

In $\mathsf{KD45}_n$, modeling belief, $L_i\varphi$ could be interpreted

> *If Bob is a neighbor of Alice, then Alice believes that Bob doesn't disbelieve φ.*

Similarly, in $\mathsf{S5}_n$, modeling knowledge, $L_i\varphi$ could be interpreted

> *If Bob is a neighbor of Alice, then Alice knows that Bob doesn't not know φ.*

In both cases, and generally in our mental model, we could call $L_i\varphi$ *diffusive knowledge* of φ.

By mimiking the common knowledge operators $E_A\varphi$ and $C_A\varphi$, we can both extend the notion of diffusive knowledge to an arbitrary subset of nodes and integrate higher order information by iterating the L_i operator. Suppose $A \subseteq \mathcal{V}_G$ is a group of agents. Then, define

$$\begin{aligned} L_A\varphi = & \bigwedge_{i\in A} L_i\varphi \\ L_A^{\omega}\varphi = & \bigwedge_{k\geqslant 1} L_A^k\varphi. \end{aligned}$$

Many questions remain. For instance, can you infer $E_{\mathcal{V}_G}\varphi$, $C_{\mathcal{V}_G}\varphi$ or $D_{\mathcal{V}_G}\varphi$ from $L_{\mathcal{V}_G}$ or $L^{\omega}\varphi$? What happens in K_n or other axiom systems? Does it depend on the structure of G?

APPENDIX A

Sheaf Theory

In Chapter 5 and 6 we were introduced to network sheaves and, more specifically, lattice valued network sheaves. However, those with some degree of familiarity with sheaves, may not even recognize that a functor $\underline{\mathcal{F}} : \mathrm{Ind}(G) \to \mathcal{S}up$ is properly a sheaf. We may also wonder what lattice-valued sheaf theory looks like in the general case.

Recall, a topological space is the data $(\mathbb{X}, \mathrm{Open}(\mathbb{X}))$ with a poset of subsets called *open sets*[22] satisfying the following conditions

1. $\emptyset \in \mathrm{Open}(\mathbb{X})$.

2. $\mathbb{X} \in \mathrm{Open}(\mathbb{X})$.

3. If $\{U\}_{i \in I} \subseteq \mathrm{Open}(\mathbb{X})$ is an arbitrary collection of opens, then their union $\bigcup_{i \in I} U_i \in \mathrm{Open}(\mathbb{X})$ is also open.

Definition A.1 (Presheaf)**.** Suppose $\mathbb{X}$ is a topologcal space and $\mathcal{D}$ is a category. Then, a *presheaf over* $\mathbb{X}$ *valued in* $\mathcal{D}$ is a contravariant functor

$$\underline{\mathcal{F}} : \mathrm{Open}(\mathbb{X})^{\mathrm{op}} \to \mathcal{D}$$

Specifically, for every open set U, a presheaf is the data

$$\begin{aligned} \underline{\mathcal{F}}(U) &\in && \mathrm{Ob}(\mathcal{D}) \\ \underline{\mathcal{F}}(V \subseteq U) &\in && \mathrm{Hom}_{\mathcal{D}}(\underline{\mathcal{F}}(V), \underline{\mathcal{F}}(U)) \end{aligned}$$

[22] The notion of an open set is quite general, but it coincides with familiar intuition. For instance, if $(\mathbb{M}, d)$ is a metric space, then a subset $U \subseteq \mathbb{M}$ is open in the *metric space topoloy* if and only for every $x \in U$ there is an $\varepsilon > 0$ such that the (open) ball $B_\varepsilon(x) = \{y \in \mathbb{M} \mid d(x, y) < \varepsilon\}$ is contained in U. The topology formed by arbitrary unions and finite intersections of balls in$\mathbb{M}$ is called the *metric space topology.*

for $V \subseteq U$ open. The architypical example of a presheaf with $\mathcal{D} = \mathcal{S}et$ over a space $\mathbb{X}$ is the following. For $U \in \text{Open}(\mathbb{X})$, assign the set of $\mathbb{R}$-valued continuous functions to U and assign the restriction of a particular map on U to the open subset $V \subseteq U$

$$\begin{aligned} \underline{\mathcal{F}}(U) &= & \{f : U \to \mathbb{R}\} \\ \underline{\mathcal{F}}(V \subseteq U) &= & f_{|V} \end{aligned}$$

Suppose $\mathcal{A}$ and $\mathcal{B}$ are categories. A functor $\mathcal{A} \to \mathcal{B}$ is *faithful* if for every pair of objects $x, y \in \text{Ob}(\mathcal{A})$, the map $\hom_{\mathcal{A}}(x, y) \hookrightarrow \hom_{\mathcal{B}}(T(x), T(y))$ is injective. Then, a *concrete category* $\mathcal{D}$ is a category that admits a faithful functor $\mathcal{D} \to \mathcal{S}et$.

Example A.1 (Suplattices)**.** The category $\mathcal{S}up$ is concrete. The functor $\mathcal{S}up \to \mathcal{S}et$ sends a complete lattice $\mathbf{L}$ to its underlying set L and a join-preserving map $f : \mathbf{K} \to \mathbf{L}$ to the ordinary function $f : K \to L$. Faithfulness is trivial because morphisms in $\mathcal{S}up$ are maps between sets that preserve structure; if two functions are equal after forgetting the join-preserving structure, then, of course, the structure-preserving maps were equal.

We define sheaves with arbitrary algebraic strucutre by first defining sheaves valued in sets. An *open cover* of $U \subseteq \mathbb{X}$ consists of a collection of open sets $\{U_i\}_{i\in I}$ such that $\bigcup_{i\in I} U_i \supseteq U$. Let U_{ij} denote the open set $U_i \cap U_j$. The following definition is taken from [107], while not standard, is one of a handful of equivalent definitions of a sheaf.

Definition A.2 (Sheaf)**.** Suppose $\underline{\mathcal{F}}$ is a presheaf of sets over X. Then, $\underline{\mathcal{F}}$ is a *sheaf* if for every open set $U \in \text{Open}(\mathbb{X})$ and open cover $\mathcal{U} = \{U_i\}_{i\in I}$ of U, the following is an equalizer

$$\begin{tikzcd} \underline{\mathcal{F}}(U) \arrow[r, dashed] & \prod_{i\in I} \underline{\mathcal{F}}(U_i) \arrow[r, shift left, "\prod_{j\in I} \underline{\mathcal{F}}(U_{ij}\subseteq U_i)"] \arrow[r, shift right, "\prod_{i\in I} \underline{\mathcal{F}}(U_{ij}\subseteq U_j)"'] & \prod_{i,j\in I} \underline{\mathcal{F}}(U_{ij}) \end{tikzcd}$$

Then, a presheaf $\underline{\mathcal{F}}$ valued in a concrete category $\mathcal{D}$ is *sheaf* if the composite presheaf in the following diagram

$$\begin{tikzcd}
\mathrm{Open}(\mathbb{X})^{\mathrm{op}} \arrow[r, "\underline{\mathcal{F}}"] \arrow[dr, dashed] & \mathcal{D} \arrow[d] \\
& \mathcal{S}et
\end{tikzcd}$$

is a sheaf.

Global sections of a sheaf over a space are simply the object $\underline{\mathcal{F}}(\mathbb{X})$. However, this definition is not very useful because it is not constructable. In certain situations, $\mathbb{X}$ is presented with a cell decomposition or, in real-life examples, is approximated with as a graph or simplicial complex.

Definition A.3. Suppose $(\mathrm{P}, \trianglelefteq)$ is a poset. Then, the *Alexandrov topology* on P, denoted $\mathrm{Alex}(\mathrm{P})$, consists of up-sets: subsets $U \subseteq \mathrm{P}$ with $x \in U$, $y \trianglerighteq u$ implies $y \in U$.

Recall, a category is *complete* or *cocomplete* if it has all (small) limits and colimits, respectively.[23]

Theorem A.1. The category $\mathcal{S}up$ of complete lattices and join-preserving maps is complete and cocomplete.

Proof. See Joyal [87, Chapter 1] for a detailed explanation. Alternatively, $\mathcal{S}up$ has products, equalizers, coproducts, and coequalizers, making $\mathcal{S}up$ complete and cocomplete [133, Theorem 3.4.12]. □

We now state a result by Curry which adresses the common (invalid) critique that functors out of posets (e.g. Tarski sheaves) are not really sheaves. The follwing theorem requires the notion of a Kan extension [133] and is stated in a form dual to the original result.

Theorem A.2 (Theorem 4.8 [42])**.** Suppose P is a poset and $\mathcal{C}$ is a complete category. Suppose $\underline{\mathcal{F}} : \mathrm{P} \to \mathcal{C}$ is a functor (e.g. network sheaf), and suppose $\iota : \mathrm{P} \to \mathrm{Alex}(\mathrm{P})^{\mathrm{op}}$ is the

[23] We will not review limits and colimits here. For a good introduction, see [133, Chpater 3].

functor associating an element $x \in \mathrm{P}$ to the open set $\uparrow x$. Then, the right Kan extension $\mathrm{Ran}_\iota(\underline{\mathcal{F}})$ of $\underline{\mathcal{F}}$ along ι is a sheaf over the Alexandrov topology $\mathrm{Alex}(\mathrm{P})$.

Corollary A.1. A Tarski sheaf $\mathcal{F} : \mathrm{P}_G \to \mathcal{L}tc$ is (naturally isomorphic to) a sheaf over the space $\mathrm{Alex}(\mathrm{P}_G)$.

Proof. Theorem 6.1, Theorem A.1 and Theorem A.2. □

APPENDIX B

Experiments

B.1. Gossip

In the following example, we demonstrate the validity of the gossip algorithm introduced in Chapter 7 for an example sheaf introduced in Chapter 9. Suppose $\mathcal{F}$ is a Tarski sheaf, such that for every $ij \in \mathcal{E}_G$, $\underline{\mathcal{F}}(ij)$ is not only a complete lattice but also a metric space $(\underline{\mathcal{F}}(ij), d_{ij})$. We measure "how close" an $\mathbf{x} \in C^0(G; \underline{\mathcal{F}})$ is to being a section with the following notion of *Dirichlet energy* or *total variation.*

$$V(\mathbf{x}) = \sum_{ij \in \mathcal{E}_G} d_{ij}\left(\underline{\mathcal{F}}(i \trianglelefteqslant ij)(x_i), \underline{\mathcal{F}}(j \trianglelefteqslant ij)(x_j)\right).$$

The following is evident

Fact. Suppose $\mathcal{F}$ is a Tarski sheaf over G and $\mathbf{x} \in C^0(G; \underline{\mathcal{F}})$. Then, $V(\mathbf{x}) = 0$ if and only if $\mathbf{x} \in H^0(G; \underline{\mathcal{F}})$.

Recall the definition of the time-varying Tarki Laplacian (Definition 7.8). In the following experiment, we generate random geometric graphs $G = (\mathcal{V}_G, \mathcal{E}_G)$ with parameters $|\mathcal{V}_G| = 40$ and $r = 0.08$. Specifically, locations $y_i \in \mathbb{R}^2$ of each $i \in \mathcal{V}_G$ are sampled uniformly at random from $[0,1]^2$; $ij \in \mathcal{E}_G$ if and only if $\|x_i - x_j\| \leqslant r$.

We construct a Kripke sheaf $\mathcal{F}_M$ (Definition 9.5) by generating a Kripke model M (really, just a frame) with randomized Kripke relations $\mathcal{K}_i$, $i \in \mathcal{V}_G$. In our experiment, we choose a fixed number of states $S = \{1, 2, \dots, 10\}$ and wire each $(a, b) \in \mathcal{K}_i$ randomly (with probability $p = 0.9$ if $a = b$, $p = 0.1$ otherwise). The relations $\mathcal{K}_i$ are generated independently. We select a random firing sequence $\tau : \mathbb{N} \to \wp(\mathcal{V}_G)$ which serves the purpose of determining which nodes broadcast to their neighbors at time t. In our experiment, we randomly select a single node to broadcast each round. For a series of random initial assignments $\mathbf{x}[0]$, we run the heat flow dynamics using the asynchronous Kripke Laplacian (Definition 9.6). The results are summarized below (Figure B.1).

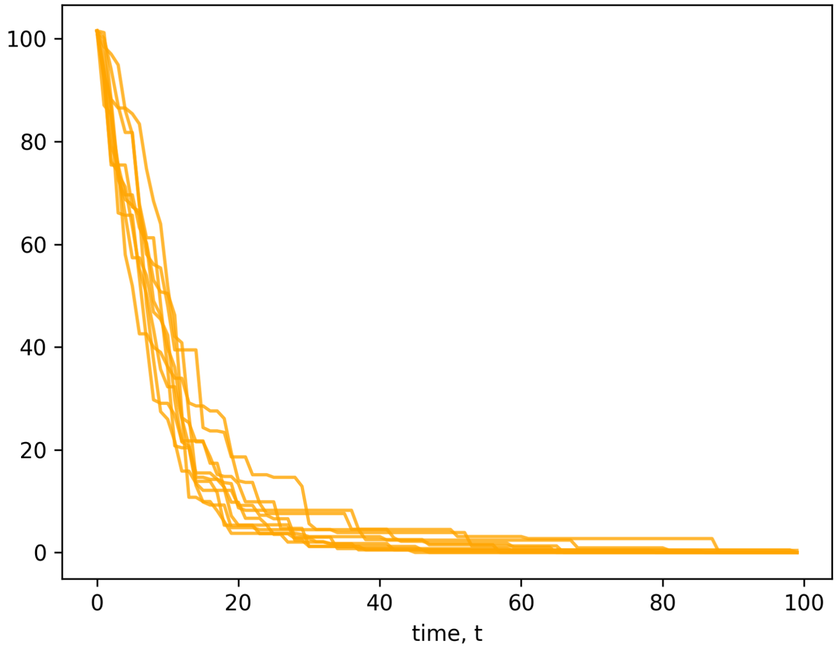


Figure B.1: Dirichlet energy of heat flow for the gossip algorithm on a geometric random graph with a randomized Kripke sheaf for 10 randomized initial conditions.

B.2. A Lattice Convolutional Neural Network (L-CNN)

We use a portion of the ModelNet10 dataset [167] as a source of point clouds. This dataset consists of 100s of 3D CAD models which represent objects from 1 of 10 classes. We sample points from the 3D models to produce finite metric spaces embedded in $\mathbb{R}^3$, then compute the corresponding multidimensional persistence modules [101], from which we produce features used as an input to a convolutional neural network classifier.

The pipline thus begins with a 3D polyhedral model, of which 3000 vertices are sampled to produce a point cloud in $\mathbb{R}^3$. This point cloud then produces a bifiltered simplicial complex, whose degree-0 persistent homology we calculate using RIVET [101], sampled at a discrete grid of 40×40 points, producing lattice-indexed signals given by the Hilbert function and the multi-graded Betti numbers ξ_0, ξ_1, ξ_2; four features in total. These are then passed to the classifier, which produces a class prediction, in this case one of 10 possible objects. As the filter function on these data sets, we use a codensity function that is the inverse sum distance from a point x to the k-nearest neighbors ($k = 100$). The name codensity is appropriate because the points in the densest regions of $\mathcal{M}$ appear earlier in the filtration.

We compare the performance of two convolutional networks on this classification task. One uses is a L-CNN and the other is a standard CNN. Each has three convolutional layers followed by two fully connected layers. The lattice-based convolution layers are of the form $\alpha \cdot \text{MeetConv}(x) + (1 - \alpha) \cdot \text{JoinConv}(x)$ for a hyperparameter $\alpha \in [0, 1]$ ($\alpha = 0.5$) Thus, *lattice layers* are a hybrid of join- and meet- convolutions. All convolution kernels have dimension 4×4, hidden convolution layers have 16 features, and the final convolution layer has 8 features. The first two convolution layers are followed by max-pooling layers with a 2×2 kernel. For the lattice convolutional layers, the support of the kernel lay in an evenly spaced 4×4 grid of points in $[m] \times [n]$, while the standard convolutional layers had a standard kernel. The inner fully connected layer has 32 features. We use a cross-entropy loss function with a softmax in the final layer. The lattice-based convolutional architecture is summarized in Figure B.2. The networks are trained with Adam gradient algorithm with learning rate 2×10^{-4} for a total of 300 epochs. We hold out 10% of the data for testing. Results are shown in Figure B.2.

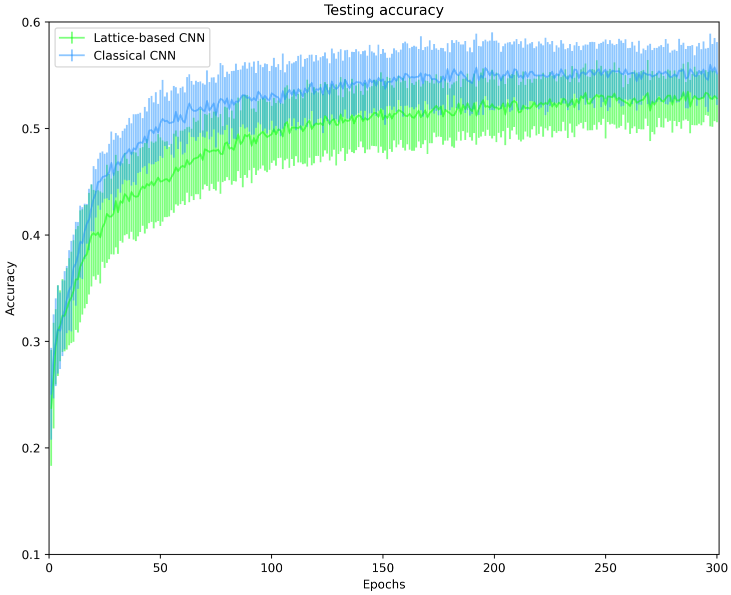


Figure B.2: Testing/training accuracy of L-CC (Lattice Convolutional Neural Network) compared to a classical CNN.

BIBLIOGRAPHY

[1] Mohamed Abdelkader, Samet Güler, Hassan Jaleel, and Jeff S Shamma. Aerial swarms: Recent applications and challenges. *Current Robotics Reports*, 2(3):309–320, 2021.

[2] Samson Abramsky and Adam Brandenburger. The sheaf-theoretic structure of nonlocality and contextuality. *New Journal of Physics*, 13(11):113036, 2011.

[3] Arman Adibi, Aryan Mokhtari, and Hamed Hassani. Submodular meta-learning. *Advances in Neural Information Processing Systems*, 33:3821–3832, 2020.

[4] Eitan Altman and Zwi Altman. S-modular games and power control in wireless networks. *IEEE Transactions on Automatic Control*, 48(5):839–842, 2003.

[5] Kenneth J Arrow. *Social choice and individual values*, volume 12. Yale university press, 2012.

[6] Alain Aspect, Philippe Grangier, and Gérard Roger. Experimental tests of realistic local theories via bell's theorem. *Physical review letters*, 47(7):460, 1981.

[7] M Atiyah, R Bott, and VK Patodi. On the heat equation and the index theorem, 1973.

[8] John C Baez and James Dolan. Categorification. *arXiv preprint arXiv:9802029*, 1998.

[9] Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT press, 2008.

[10] Ruzena Bajcsy. Computer identification of visual surfaces. *Computer Graphics and Image Processing*, 2(2):118–130, 1973.

[11] Afonso S Bandeira, Amit Singer, and Daniel A Spielman. A cheeger inequality for the graph connection laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34 (4):1611–1630, 2013.

[12] Federico Barbero, Cristian Bodnar, Haitz Saez de Ocariz Borde, Michael Bronstein, Petar Velickovic, and Pietro Lio. Sheaf neural networks with connection laplacians. *arXiv preprint arXiv:2206.08702*, 2022.

[13] Jean-Pierre Barthélemy and Melvin F. Janowitz. A formal theory of consensus. *SIAM Journal on Discrete Mathematics*, 4(3):305–322, 1991.

[14] Ulrich Bauer and Herbert Edelsbrunner. The morse theory of čech and delaunay complexes. *Transactions of the American Mathematical Society*, 369(5):3741–3762, 2017.

[15] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[16] Radim Bêlohlávek. Fuzzy galois connections. *Mathematical Logic Quarterly*, 45(4): 497–504, 1999.

[17] Štěpánka Bilová. Lattice theory–its birth and life. *Mathematics throughout the ages*, pages 250–257, 2001.

[18] Garrett Birkhoff. *Lattice theory*, volume 25. American Mathematical Soc., 1940.

[19] Garrett Birkhoff and John Von Neumann. The logic of quantum mechanics. *Annals of mathematics*, pages 823–843, 1936.

[20] Cristian Bodnar, Francesco Di Giovanni, Benjamin Paul Chamberlain, Pietro Lio, and Michael M Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *arXiv preprint arXiv:2202.04579*, 2022.

[21] George Boole. *The mathematical analysis of logic*. Philosophical Library, 1847.

[22] Theodor Borsche and Sid Ahmed Attia. On leader election in multi-agent control systems. In *2010 Chinese Control and Decision Conference*, pages 102–107, 2010.

[23] Michal Botur, Radomír Halaš, Radko Mesiar, and Jozef Pócs. On generating of idempotent aggregation functions on finite lattices. *Information Sciences*, 430:39–45, 2018.

[24] Glen E Bredon. *Sheaf theory*, volume 170. Springer, 2012.

[25] L Brickman and PA Fillmore. The invariant subspace lattice of a linear transformation. *Canadian Journal of Mathematics*, 19:810–822, 1967.

[26] David Bryant. A classification of consensus methods for phylogenetics. In *Bioconsensus: DIMACS Working Group Meetings on Bioconsensus: October 25-26, 2000 and October 2-5, 2001, DIMACS Center*, volume 61, page 163. American Mathematical Soc., 2003.

[27] Peter Bubenik and Nikola Milićević. Homotopy, homology and persistent homology using Čech's closure spaces. *arXiv preprint arXiv:2104.10206*, 2021.

[28] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer, 2013.

[29] Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009.

[30] Gunnar E Carlsson, Facundo Mémoli, et al. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.*, 11(Apr):1425–1470, 2010.

[31] Christos G Cassandras and Stéphane Lafortune. *Introduction to discrete event systems.* Springer, 2008.

[32] Michael J Catanzaro, Justin M Curry, Brittany Terese Fasy, Jānis Lazovskis, Greg Malen, Hans Riess, Bei Wang, and Matthew Zabka. Moduli spaces of morse functions for persistence. *Journal of Applied and Computational Topology*, 4(3):353–385, 2020.

[33] T-H Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral properties of hypergraph laplacian and approximation algorithms. *Journal of the ACM (JACM)*, 65(3):1–48, 2018.

[34] Chi-Tsong Chen. *Linear system theory and design.* Saunders college publishing, 1984.

[35] Nicholas A Christakis and James H Fowler. *Connected: The surprising power of our social networks and how they shape our lives.* Little, Brown Spark, 2009.

[36] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[37] Bob Coecke and Aleks Kissinger. Picturing quantum processes. In *International Conference on Theory and Application of Diagrams*, pages 28–31. Springer, 2018.

[38] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms.* MIT press, 2022.

[39] Jorge Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44(3):726–737, 2008.

[40] Jorge Cortés, Sonia Martínez, and Francesco Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2006.

[41] Raymond A Cuninghame-Green and Peter Butkovic. The equation $A \otimes x = B \otimes y$ over $(\max, +)$. *Theoretical Computer Science*, 293(1):3–12, 2003.

[42] Justin Curry. Functors on posets left kan extend to cosheaves: an erratum. *arXiv preprint arXiv:1907.09416*, 2019.

[43] Justin Michael Curry. *Sheaves, cosheaves and applications.* University of Pennsylvania, 2014.

[44] Justin Michael Curry. Dualities between cellular sheaves and cosheaves. *Journal of Pure and Applied Algebra*, 222(4):966–993, 2018.

[45] Brian A Davey and Hilary A Priestley. *Introduction to lattices and order*. Cambridge University Press, 2002.

[46] Vin De Silva, Elizabeth Munch, and Amit Patel. Categorified reeb graphs. *Discrete & Computational Geometry*, 55(4):854–906, 2016.

[47] Petar Djuric and Cédric Richard. *Cooperative and graph signal processing: principles and applications*. Academic Press, 2018.

[48] Beno Eckmann. Harmonische funktionen und randwertaufgaben in einem komplex. *Commentarii Mathematici Helvetici*, 17(1):240–255, 1944.

[49] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.

[50] Francis Y Edgeworth. The pure theory of monopoly. *Papers relating to political economy*, 1(111):42, 1925.

[51] Pavel I Etingof, Oleg Golberg, Sebastian Hensel, Tiankai Liu, Alex Schwendner, Dmitry Vaintrob, and Elena Yudovina. *Introduction to representation theory*, volume 59. American Mathematical Soc., 2011.

[52] Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Vardi. *Reasoning about knowledge*. MIT press, 2004.

[53] Georgios E Fainekos, Hadas Kress-Gazit, and George J Pappas. Hybrid controllers for path planning: A temporal logic approach. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4885–4890. IEEE, 2005.

[54] J Alexander Fax and Richard M Murray. Information flow and cooperative control of vehicle formations. *IEEE transactions on automatic control*, 49(9):1465–1476, 2004.

[55] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2): 374–382, 1985.

[56] Brendan Fong and David I Spivak. *An invitation to applied category theory: seven sketches in compositionality*. Cambridge University Press, 2019.

[57] Peter J Freyd. *Abelian categories*, volume 1964. Harper & Row New York, 1964.

[58] Fernando Gama, Antonio G Marques, Geert Leus, and Alejandro Ribeiro. Convolutional neural network architectures for signals supported on graphs. *IEEE Transactions on Signal Processing*, 67(4):1034–1049, 2018.

[59] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[60] Robert Ghrist. Laplacians and network sheaves. Public lecture, 2021.

[61] Robert Ghrist. *Applied Dynamical Systems.* Agenbyte Press, 2022.

[62] Robert Ghrist and Sanjeevi Krishnan. Positive alexander duality for pursuit and evasion. *SIAM Journal on Applied Algebra and Geometry*, pages 308–327, 2017.

[63] Robert Ghrist and Hans Riess. Cellular sheaves of lattices and the tarski laplacian. *Homology, Homotopy and Applications*, 24(1):325–345, 2022.

[64] Robert W Ghrist. *Elementary applied topology*, volume 1. Createspace Seattle, 2014.

[65] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[66] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

[67] Joseph A Goguen. Sheaf semantics for concurrent interacting objects. *Mathematical Structures in Computer Science*, 2(2):159–191, 1992.

[68] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[69] Marco Grandis. *Homological Algebra: In Strongly Non-Abelian Settings.* World scientific, 2013.

[70] George Grätzer. *General lattice theory.* Springer, 2002.

[71] Javier Gutierrez Garcia, Iraide Mardones-Pérez, María Angeles de Prada Vicente, and Dexue Zhang. Fuzzy galois connections categorically. *Mathematical Logic Quarterly*, 56(2):131–147, 2010.

[72] Jakob Hansen. From connections to relationships with cellular sheaves. In *SIAM Workshop on Network Science*, 2019.

[73] Jakob Hansen. *Laplacians of Cellular Sheaves: Theory and Applications.* PhD thesis, University of Pennsylvania, 2020.

[74] Jakob Hansen and Thomas Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

[75] Jakob Hansen and Robert Ghrist. Distributed optimization with sheaf homological constraints. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 565–571. IEEE, 2019.

[76] Jakob Hansen and Robert Ghrist. Learning sheaf laplacians from smooth signals. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5446–5450. IEEE, 2019.

[77] Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

[78] Jakob Hansen and Robert Ghrist. Opinion dynamics on discourse sheaves. *SIAM Journal on Applied Mathematics*, 81(5):2033–2060, 2021.

[79] Robert M Haralick, Stanley R Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, (4):532–550, 1987.

[80] Keith A Hardie, Klaus Heiner Kamps, and Rudger W Kieboom. A homotopy 2-groupoid of a hausdorff space. *Applied Categorical Structures*, 8(1):209–234, 2000.

[81] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[82] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[83] Jianwei Huang, Randall A Berry, and Michael L Honig. Distributed interference compensation for wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(5):1074–1084, 2006.

[84] Weiyu Huang, Antonio G Marques, and Alejandro R Ribeiro. Rating prediction via graph signal processing. *IEEE Transactions on Signal Processing*, 66(19):5066–5081, 2018.

[85] Melvin F Janowitz and Sándor Radeleczki. Aggregation on a finite lattice. *Order*, 33 (3):371–388, 2016.

[86] Jürgen Jost. *Riemannian geometry and geometric analysis*, volume 42005. Springer, 2008.

[87] André Joyal and Myles Tierney. *An extension of the Galois theory of Grothendieck*, volume 309. American Mathematical Soc., 1984.

[88] Yiannis Kantaros and Michael M Zavlanos. Distributed intermittent connectivity control of mobile robot networks. *IEEE Transactions on Automatic Control*, 62(7): 3109–3121, 2016.

[89] Masaki Kashiwara and Pierre Schapira. Persistent homology and microlocal sheaf theory. *Journal of Applied and Computational Topology*, 2(1):83–113, 2018.

[90] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE, 2003.

[91] Bahador Khaleghi, Alaa Khamis, Fakhreddine O Karray, and Saiedeh N Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information fusion*, 14(1):28–44, 2013.

[92] Woojin Kim and Facundo Mémoli. Formigrams: Clustering summaries of dynamic data. In *CCCG*, pages 180–188, 2018.

[93] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[94] MS Knebelman. Spaces of relative parallelism. *Annals of Mathematics*, pages 387–399, 1951.

[95] Sanjeevi Krishnan and Crichton Ogle. Invertibility in category representations. *arXiv preprint arXiv:2010.11276*, 2020.

[96] Fred Landman. *Structures for semantics*, volume 45. Springer, 2012.

[97] Bruno Leclerc and Bernard Monjardet. Latticial theory of conensus. In W.A. Barnett, editor, *Social Choice, Welfare, and Ethics: Proceedings of the Eighth International Symposium in Economic Theory and Econometrics*, International Symposia in Economic Theory and Econometrics. Cambridge University Press, 1995.

[98] Bruno Leclerc and Bernard Monjardet. Aggregation and residuation. *Order*, 30(1): 261–268, 2013.

[99] John M Lee. Smooth manifolds. In *Introduction to smooth manifolds*, pages 1–31. Springer, 2013.

[100] John M Lee. *Introduction to Riemannian manifolds*, volume 176. Springer, 2018.

[101] Michael Lesnick and Matthew Wright. Interactive visualization of 2-d persistence modules. *arXiv preprint arXiv:1512.00180*, 2015.

[102] Hua Li and Edwin KP Chong. On a connection between information and group lattices. *Entropy*, 13(3):683–708, 2011.

[103] Wuchen Li. Transport information geometry: Riemannian calculus on probability simplex. *Information Geometry*, pages 1–47, 2021.

[104] Lek-Heng Lim. Hodge laplacians on graphs. *Siam Review*, 62(3):685–715, 2020.

[105] Gongliang Liu, Ruisong Wang, Haijun Zhang, Wenjing Kang, Theodoros A Tsiftsis, and Victor CM Leung. Super-modular game-based user scheduling and power allocation for energy-efficient noma network. *IEEE Transactions on Wireless Communications*, 17(6):3877–3888, 2018.

[106] Gabriel AD Lopes, Bart Kersbergen, Ton JJ van den Boom, Bart De Schutter, and Robert Babuška. Modeling and control of legged locomotion via switching max-plus models. *IEEE Transactions on Robotics*, 30(3):652–665, 2014.

[107] Saunders MacLane and Ieke Moerdijk. *Sheaves in geometry and logic: A first introduction to topos theory.* Springer, 2012.

[108] Robert MacPherson and Amit Patel. Persistent local systems. *Advances in Mathematics*, 386:107795, 2021.

[109] Petros Maragos. Morphological filtering. In *The essential guide to image processing*, pages 293–321. Elsevier, 2009.

[110] Petros Maragos. Dynamical systems on weighted lattices: General theory. *Mathematics of Control, Signals, and Systems*, 29(4):1–49, 2017.

[111] Petros Maragos, Vasileios Charisopoulos, and Emmanouil Theodosis. Tropical geometry and machine learning. *Proceedings of the IEEE*, 109(5):728–755, 2021.

[112] Yukio Matsumoto. *An introduction to Morse theory*, volume 208. American Mathematical Soc., 2002.

[113] Alexander McCleary and Amit Patel. Edit distance and persistence diagrams over lattices. *SIAM Journal on Applied Algebra and Geometry*, 6(2):134–155, 2022.

[114] Bernard Monjardet and Vololonirina Raderanirina. Lattices of choice functions and consensus problems. *Social Choice and Welfare*, 23(3):349–382, 2004.

[115] Sidiney B Montanhano. Characterization of contextuality with semi-module Čech cohomology and its relation with cohomology of effect algebras. *arXiv preprint arXiv:2104.11411*, 2021.

[116] Michael Moy, Robert Cardona, Robert Green, Jacob Cleveland, Alan Hylton, and Robert Short. Path optimization sheaves. *arXiv preprint arXiv:2012.05974*, 2020.

[117] Abubakr Muhammad and Magnus Egerstedt. Control using higher order laplacians in network topologies. In *Proc. of 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 1024–1038. Citeseer, 2006.

[118] Michael C Munger. *Choosing in groups: Analytical politics revisited.* Cambridge University Press, 2015.

[119] So Nakashima and Takanori Maehara. Subspace selection via dr-submodular maximization on lattices. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4618–4625, 2019.

[120] Vidit Nanda and Amit Patel. Canonical stratifications along bisheaves. In *Topological Data Analysis*, pages 391–403. Springer, 2020.

[121] Fabrícia F Nascimento, Mario dos Reis, and Ziheng Yang. A biologist's guide to bayesian phylogenetic analysis. *Nature ecology & evolution*, 1(10):1446–1454, 2017.

[122] Behrang Monajemi Nejad, Sid Ahmed Attia, and Jorg Raisch. Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies. In *2009 XXII International Symposium on Information, Communication and Automation Technologies*, pages 1–7. IEEE, 2009.

[123] Dean A Neumann and Victor T Norton. On lattice consensus methods. *Journal of Classification*, 3(2):225–255, 1986.

[124] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[125] Oystein Ore. Galois connexions. *Transactions of the American Mathematical Society*, 55(3):493–513, 1944.

[126] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

[127] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6:1–38, 2017.

[128] Steve Y Oudot. *Persistence theory: from quiver representations to data analysis*, volume 209. American Mathematical Soc., 2017.

[129] Alejandro Parada-Mayorga and Alejandro Ribeiro. Algebraic neural networks: Stability to deformations. *IEEE Transactions on Signal Processing*, 69:3351–3366, 2021.

[130] Alejandro Parada-Mayorga, Hans Riess, Alejandro Ribeiro, and Robert Ghrist. Quiver signal processing. *arXiv preprint arXiv:2010.11525*, 2020.

[131] Markus Puschel. A discrete signal processing framework for meet/join lattices with applications to hypergraphs and trees. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5371–5375. IEEE, 2019.

[132] Markus Püschel, Bastian Seifert, and Chris Wendler. Discrete signal processing on meet/join lattices. *IEEE Transactions on Signal Processing*, 69:3571–3584, 2021.

[133] Emily Riehl. *Category theory in context.* Courier Dover Publications, 2017.

[134] Hans Riess and Robert Ghrist. Diffusion of information on networked lattices by gossip. In *2022 Proceedings on the Conference on Decision and Control.* IEEE, 2022.

[135] Hans Riess, Yiannis Kantaros, George Pappas, and Robert Ghrist. A temporal logic-based hierarchical network connectivity controller. In *2021 Proceedings of the Conference on Control and its Applications*, pages 17–24. SIAM, 2021.

[136] Hans Matthew Riess and Jakob Hansen. Multidimensional persistence module classification via lattice-theoretic convolutions. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020.

[137] Michael Robinson. *Topological signal processing*, volume 81. Springer, 2014.

[138] Michael Robinson. Sheaves are the canonical data structure for sensor integration. *Information Fusion*, 36:208–224, 2017.

[139] Michael Robinson. Assignments to sheaves of pseudometric spaces. *Compositionality*, 2, 2020.

[140] Steven Roman. *Lattices and ordered sets.* Springer, 2008.

[141] Gian-Carlo Rota. On the foundations of combinatorial theory i. theory of möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 2(4):340–368, 1964.

[142] Gian-Carlo Rota. The many lives of lattice theory. *Notices of the AMS*, 44(11): 1440–1445, 1997.

[143] Luana Ruiz, Fernando Gama, Antonio Garcia Marques, and Alejandro Ribeiro.

Invariance-preserving localized activation functions for graph neural networks. *IEEE Transactions on Signal Processing*, 68:127–141, 2019.

[144] Hanamantagouda P Sankappanavar and Stanley Burris. A course in universal algebra. *Graduate Texts Math*, 78:56, 1981.

[145] Edward R Scheinerman and John C Wierman. On circle containment orders. *Order*, 4(4):315–318, 1988.

[146] Jürgen Schmidt. Beiträge zur filtertheorie. ii. *Mathematische Nachrichten*, 10(3-4): 197–232, 1953.

[147] Urs Schreiber and Konrad Waldorf. Parallel transport and functors. *Journal of Homotopy and Related Structures*, 4(1):187–244, 2009.

[148] Bastian Seifert, Chris Wendler, and Markus Püschel. Wiener filter on meet/join lattices. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5355–5359. IEEE, 2021.

[149] Bastian Seifert, Chris Wendler, and Markus Püschel. Causal fourier analysis on directed acyclic graphs and posets. *arXiv preprint arXiv:2209.07970*, 2022.

[150] Jean-Pierre Serre et al. *Linear representations of finite groups*, volume 42. Springer, 1977.

[151] Claude Shannon. The lattice theory of information. *Transactions of the IRE Professional Group on Information Theory*, 1(1):105–107, 1953.

[152] Allen Dudley Shepard. *A cellular description of the derived category of a stratified space*. Brown University, 1985.

[153] Amit Singer and H-T Wu. Vector diffusion maps and the connection laplacian. *Communications on pure and applied mathematics*, 65(8):1067–1144, 2012.

[154] Stephen L Smith, Jana Tumova, Calin Belta, and Daniela Rus. Optimal path planning for surveillance with temporal-logic constraints. *The International Journal of Robotics Research*, 30(14):1695–1708, 2011.

[155] Benjamin Steinberg et al. *Representation theory of finite monoids*. Springer, 2016.

[156] Alireza Tahbaz-Salehi and Ali Jadbabaie. A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 4664–4669. IEEE, 2006.

[157] Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic control*, 52(5):863–868, 2007.

[158] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific journal of Mathematics*, 5(2):285–309, 1955.

[159] Donald M Topkis. Minimizing a submodular function on a lattice. *Operations research*, 26(2):305–321, 1978.

[160] Rakesh V Vohra. *Advanced mathematical economics.* Routledge, 2004.

[161] John Von Neumann. *Mathematical foundations of quantum mechanics: New edition.* Princeton university press, 2018.

[162] Duncan J Watts. The" new" science of networks. *Annual review of sociology*, pages 243–270, 2004.

[163] Duncan J Watts. *Six degrees: The science of a connected age.* WW Norton & Company, 2004.

[164] Chris Wendler and Markus Püschel. Sampling signals on meet/join lattices. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE, 2019.

[165] Rudolf Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In *Ordered Sets*, pages 445–470. Springer, 1982.

[166] John Wright and Yi Ma. *High-dimensional data analysis with low-dimensional models: Principles, computation, and applications.* Cambridge University Press, 2022.

[167] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[168] Songyang Zhang, Zhi Ding, and Shuguang Cui. Introducing hypergraph signal processing: Theoretical foundation and practical applications. *IEEE Internet of Things Journal*, 7(1):639–660, 2019.

[169] Xiong Zheng and Vijay Garg. Byzantine lattice agreement in synchronous message passing systems. In *34th International Symposium on Distributed Computing (DISC 2020).* Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[170] Xiong Zheng and Vijay Garg. Byzantine lattice agreement in asynchronous systems. In *24th International Conference on Principles of Distributed Systems*, 2021.

[171] Xiong Zheng, Changyong Hu, and Vijay K Garg. Lattice agreement in message passing systems. In *32nd International Symposium on Distributed Computing (DISC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[172] Lin Zhou. The set of nash equilibria of a supermodular game is a complete lattice. *Games and economic behavior*, 7(2):295–300, 1994.