Rational Concept Analysis

Lucas Carr

A thesis submitted for the degree of
Master of Computer Science
Department of Computer Science
University of Cape Town

March 2025

Acknowledgements

Abstract

Contents

Introduction

Part I

Foundations

Chapter 1

Mathematical Preliminaries

This chapter is intended to serve as a reference point, clarifying ideas and notation for the more fundamental concepts that will be used throughout the remainder of this dissertation. In the first section we discuss order and lattice theory. The third and final section introduces propositional logic, as well more general ideas in logic.

1.1 Order and Lattice Theory

This section refers extensively to the fundamental text by Davey and Priestley [1], as well as Ganter and Wille [2].

1.1.1 Orders

A *binary relation* R over the sets X and Y is a set of ordered pairs (x, y) where $x \in X$ and $y \in Y$. We may choose to express $(x, y) \in R$ using infix notation and write xRy, which tells us that R relates x to y.

Certain binary relations, satisfying specific properties, occur frequently enough to warrant their own denomination. One such relation, which will be used in almost every section of this dissertation, is called a *partial order*.

Definition 1. A *partial-order* is a binary relation $\preceq \subseteq X \times X$ that satisfies the following properties:

$$\text{(Reflexivity)} \quad x \preceq x \tag{1.1}$$

$$\text{(Antisymmetry)} \quad x \preceq y \text{ and } y \preceq x \text{ implies } x = y \tag{1.2}$$

$$\text{(Transitivity)} \quad x \preceq y \text{ and } y \preceq z \text{ implies } x \preceq z \tag{1.3}$$

for all $x, y, z \in X$.

Frequently, 'preference' is used as a metonymy for an order, and so in this context "element x is preferred to y" should be taken to mean that $(x, y) \in \preceq$, or simply $x \preceq y$.

We write $x \npreceq y$ to indicate that (x, y) is not in the relation, and $x \prec y$ for the case where $x \preceq y$ and $x \neq y$. In the scenario where $x \npreceq y$ and $y \npreceq x$—i.e., that x and y are incomparable—we write $x \| y$. From a partial-order we can quite easily induce the notion of a *strict partial-order*.

Definition 2. A *strict partial-order* is a binary relation $\prec \subseteq X \times X$ that satisfies:

$$\text{(Irreflexivity)} \quad x \nprec x \tag{1.4}$$

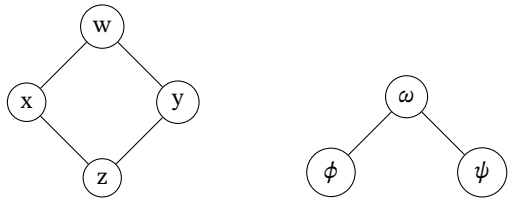
$$\text{(Asymmetry)} \quad x \prec y \text{ implies } y \nprec x \tag{1.5}$$

$$\text{(Transitivity)} \quad x \prec y \text{ and } y \prec z \text{ implies } x \prec z \tag{1.6}$$

for all $x, y, z \in X$.

An *ordered set* is a pair $(X, \preceq)$ with X being a set and $\preceq$ being an ordering on the elements of X. We make notation easier, and use $\mathbf{X}$ to denote the pair; moreover, the order relation associated with $\mathbf{X}$ may be written $\preceq_X$ in settings where ambiguity arises. If $\mathbf{Y}$ is a subset of $\mathbf{X}$, then $\mathbf{Y}$ inherits the order relation from $\mathbf{X}$; and so, for $x, y \in \mathbf{Y}$, $x \preceq_Y y$ if and only if $x \preceq_X y$.

We can visualize ordered sets through the use of *Hasse* diagrams.



(a) $\mathbf{A}$ (b) $\mathbf{B}$

Figure 1.1: Hasse diagrams of two partially ordered sets

As an illustrative example, from the ordered set in Sub-figure 1.1a we read that $z \preceq x$, as there is a (strictly) upward path from z to x. In fact, it is clear that $z \preceq w, x, y, z$, or that "z is preferred to every other element in $\mathbf{A}$". We say such an element is *minimal*.

More formally, an element $x \in \mathbf{X}$ is *minimal* with respect to the ordering if there exists no distinct element $y \in \mathbf{X}$ such that $y \preceq x$. Conversely, we say x is *maximal* if there exists no distinct $y \in \mathbf{X}$ where $x \preceq y$. Then x is the *minimum* element if $x \preceq y$ for all $y \in \mathbf{X}$; the dual notion of a *maximum* is defined as we might expect.

Definition 3. Let $\mathbf{X}$ and $\mathbf{Y}$ be ordered sets with a mapping $\varphi : \mathbf{X} \to \mathbf{Y}$. We call φ an *order-preserving* (or, isotone) map if $x \preceq_X y$ implies $\varphi(x) \preceq_Y \varphi(y)$. It is an *order-embedding* if it is injective, and $x \preceq_X y$ if and only if $\varphi(x) \preceq_Y \varphi(y)$ for all $x, y \in X$. Finally, φ is an *order-isomorphism* if it is an order-embedding that is also *surjective*. The dual notion to an order-preserving map is an *order-reversing* (or, antitone) map.

1.1.2 Lattice Theory

Lattice theory studies partially ordered sets that behave well with respect to upper and lower bounds. Given an ordered set $\mathbf{X}$ and a subset $\mathbf{Y} \subseteq \mathbf{X}$, the set of upper bounds of $\mathbf{Y}$ is defined as

$$\mathbf{Y}^u := \{x \in \mathbf{X} \mid \forall y \in \mathbf{Y} : y \preceq x\},$$

and the set of lower bounds, $\mathbf{Y}^l$, is defined dually. If $\mathbf{Y}^u$ has a minimum element, then we call that element the *supremum* of $\mathbf{Y}$; dually, if $\mathbf{Y}^l$ has a maximum element, then we call that element the *infimum* of $\mathbf{Y}$. Instead of talking about the supremum of two elements $x, y \in \mathbf{X}$, we opt for the term *join* and write $x \vee y$, or $\bigvee \mathbf{Y}$ where $\mathbf{Y} \subseteq \mathbf{X}$. Instead of infimum, we say *meet* and write $x \wedge y$, or $\bigwedge \mathbf{Y}$.

A *lattice* is a non-empty partially ordered set $\mathbf{L}$ in which every pair of elements $x, y \in \mathbf{L}$ for which $x \wedge y$ and $x \vee y$ exists. If $\bigwedge \mathbf{M}$ and $\bigvee \mathbf{M}$ exist for any subset $\mathbf{M} \subseteq \mathbf{L}$ then $\mathbf{L}$ is a *complete lattice*.

1.1.2.1 Lattices as algebraic structures

From another perspective, a lattice can be seen as an algebraic structure where meet and join are binary operations on a set. Then, $\langle L, \vee, \wedge \rangle$ formally represents a lattice. Often, we simplify notation to just L when the context clearly indicates the operations.

If the set L is equipped with only one of these operations, the structure is called a *semilattice*. Specifically, the structure $\langle L, \vee \rangle$ (resp. $\langle L, \wedge \rangle$) is called a *join semilattice* (resp. a *meet semilattice*). Such a structure is a reduct of a lattice, satisfying the following properties for all $x, y, z \in L$:

$$\text{(associative)} \quad (x \vee y) \vee z = x \vee (y \vee z) \qquad \text{(resp.)} \quad (x \wedge y) \wedge z = x \wedge (y \wedge z) \tag{1.7}$$

$$\text{(commutative)} \quad x \vee y = y \vee x \qquad \text{(resp.)} \quad x \wedge y = y \wedge x \tag{1.8}$$

$$\text{(idempotent)} \quad x \vee x = x \qquad \text{(resp.)} \quad x \wedge x = x \tag{1.9}$$

Finally, when considering both operations simultaneously, the interaction between join and meet operations within a lattice structure is governed by the absorption laws:

$$\text{(absorption)} \quad x \vee (x \wedge y) = x \qquad \text{(resp.)} \quad x \wedge (x \vee y) = x \tag{1.10}$$

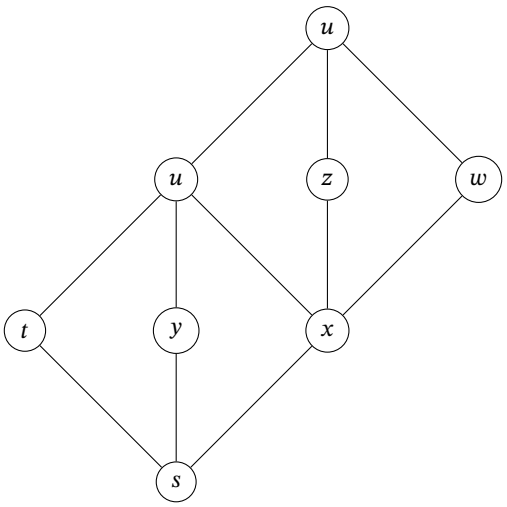


Figure 1.2: A complete lattice

Figure 1.2 illustrates a simple complete lattice; verifying that arbitrary meets and joins exist for all subsets. There is an obvious similarity between this representation of a lattice and the Hasse diagrams in Figure 1.1 used to describe partially ordered sets. This is perhaps obvious, given the opening remarks where we described a lattice as an ordered set, satisfying certain properties on upper and lower bounds. However, Lemma 1 makes the connection between the order and algebraic views of lattices concrete.

Lemma 1 (The Connecting Lemma)**.** If $\mathbf{L}$ is a lattice with $x, y \in \mathbf{L}$, then the following are equivalent:

1. $x \preceq y$;
2. $x \vee y = y$;
3. $x \wedge y = x$.

Moving forward, we will no longer keep the order-theoretic and algebraic perspectives on lattices strictly separate; rather, we will tacitly adopt whichever viewpoint best suits the surrounding context. Lattices are central to Formal Concept Analysis and will be revisited in Chapter 2.

1.2 Propositional Logic

Propositional logic is a system for abstracting reasoning away from natural language. A *propositional statement* is a claim like "Tralfamadorians have one eye"; that is, something that can be assigned a value of *true* or *false* [3, p. 7]. More complex statements can be recursively constructed from simpler ones; our interest is then about which propositions follow from others.

1.2.1 Syntax

Propositional atoms are the construction blocks of a propositional language. They are indivisible statements that can either be true or false, and maybe be combined with the boolean connectives $\{\neg, \vee, \wedge, \rightarrow\}$ to construct more complex statements, or *formulae*. We denote propositional atoms with lower-case letters p, q, r, s, and t, and formulae with lower-case Greek letters $\alpha, \beta, \gamma, \phi, \psi$, and φ.

Not all combinations of atoms and boolean connectives result in meaningful expressions (for example, '$\wedge \rightarrow \phi$' has no discernible meaning) and so we make a distinction between any formulae and *well-formed formulae*. The latter being constructions that are valid with respect to the rules of the grammar defined in Backus-Naur form below. We are entirely uninterested in formulae that are not well-formed formulae, and so we drop the 'well-formed' suffix with the recognition that we shall never again mention the alternative [4, p. 33].

$$\phi ::= p \mid (\neg\phi) \mid (\phi_1 \vee \phi_2) \mid (\phi_1 \wedge \phi_2) \mid (\phi_1 \rightarrow \phi_2) \mid (\phi_1 \leftrightarrow \phi_2) \tag{1.11}$$

Any formula accepted by this grammar is said to be in the language $\mathcal{L}^{\mathcal{P}}$, although we will drop the superscript where possible. To make reading this dissertation slightly more enjoyable, we may construct examples where we denote propositional atoms by monospaced text, enabling the expression of formulae like `tralfamadorians` $\rightarrow \neg$`human`, which should be interpreted as meaning "Tralfamadorians are not human".

1.2.2 Semantics

In the previous section we described the syntax of propositional logic and how the boolean connectives enable the construction of arbitrary formulae from atoms. The semantics of propositional logic are concerned with how meaning can be ascribed to these formulae. The aim is to provide a method of answer questions like "when is this formula true?" or, "if ϕ, ψ, φ are true, what else must be true?".

Propositional atoms were described as indivisible statements that can be assigned values of *true* or *false*. We now define a function that assigns a truth value to each proposition in the set $\mathcal{P}$ of propositions, sometimes called the *universe of discourse*.

Definition 4. A *valuation* is a function $u : \mathcal{P} \rightarrow \{\, \textit{true}, \textit{false} \,\}$ that assigns a truth value to each propositional atom.

Given a set of atoms $\mathcal{P} = \{p, q, r\}$, we write $\mathcal{U}$ to denote the set of all possible valuations. A valuation $u \in \mathcal{U}$ *satisfies* an atom $p \in \mathcal{P}$ if u maps p to *true*, then we write $u \Vdash p$. Otherwise, we write $u \nVdash p$ to indicate that u does not satisfy p (in this context, meaning u maps p to *false*) [3, p. 16]. We might represent the valuation that maps p and q to true but r to false by $\{p, q, \overline{r}\}$.

This satisfaction relation can be extended beyond propositional atoms to include more complex formulae, as described in § 1.2.1. For any $\phi, \psi \in \mathcal{L}$

- $u \Vdash \neg\phi$ if and only if $u \nVdash \phi$
- $u \Vdash \phi \vee \psi$ if and only if $u \Vdash \phi$ or $u \Vdash \psi$
- $u \Vdash \phi \wedge \psi$ if and only if $u \Vdash \phi$ and $u \Vdash \psi$
- $u \Vdash \phi \rightarrow \psi$ if and only if $u \Vdash \neg\phi$ or $u \Vdash \psi$
- $u \Vdash \phi \leftrightarrow \psi$ if and only if $u \Vdash \phi \rightarrow \psi$ and $u \Vdash \psi \rightarrow \phi$

Definition 5. For a valuation $u \in \mathcal{U}$ and formula $\phi \in \mathcal{L}$ we call u a *model* of ϕ if and only if u satisfies ϕ. The set of models of ϕ is constructed by $\hat{\phi} := \{u \in \mathcal{U} \mid u \Vdash \phi\}$.

Satisfiability can be extended to sets of of formulae so that a valuation $u \in \mathcal{U}$ *satisfies* the set $\Phi := \{\phi_0, \dots, \phi_n\}$ when $u \Vdash \phi_i$ for all $0 \leq i \leq n$, and we write $u \Vdash \Phi$. If no such valuation exists then Φ is unsatisfiable [3, p. 31].

1.2.3 Logical Consequence

The introduction of models at the end of § 1.2 leads quite naturally into a discussion on the matter of *logical consequence*, which provides an answer to the question of when it is appropriate to infer one formula from another [5, p. 408].

For example, if we know that "*Billy lived in Slaughterhouse 5*" and that "*The inhabitants of Slaughterhouse 5 survived the bombing of Dresden*". We may then hold the view that, as a consequence of these two pieces of knowledge, it would be sensible to infer that "*Billy survived the bombing of Dresden*". Of course, this inference being *sensible*—under some common concept of consequence—gives little insight into how logical consequence may be appropriately be formalised.

We place a brief moratorium on this discussion to (formally) introduce the notions of the *object* and *metalanguage*. In the scenario we have just described, the italicised sentences from a part of the object language: the language we use to model the world and represent information. The metalanguage facilitates reasoning about elements in the object language; that is, we can use the metalanguage to describe one sentence being a consequence of another (where these sentences are elements in the object language). Here, *consequence* is an element of the metalanguage [3, p 22].

In this case, both the object and metalanguage are comprised of English which can certainly lead to confusion. It is more common to use a language like the one described in § 1.2.1 as the object language, and a metalanguage comprising of symbols like $\vDash$ and $\vdash$, which we introduce below.

Definition 6. Let Γ be a set of formulae and φ a formula in the language $\mathcal{L}$. We say that φ is a *logical consequence* of Γ, and write $\Gamma \vDash \varphi$, if and only if every model of Γ is a model of φ, or equivalently if $\hat{\Gamma} \subseteq \hat{\varphi}$.

Logical consequence can be used to define an operator $\mathcal{C}n$ such that if Γ is a set of formulae in $\mathcal{L}$, then $\mathcal{C}n(\Gamma) := \{\alpha \in \mathcal{L} \mid \Gamma \vDash \alpha\}$

Example 1. Let us model the earlier scenario in propositional logic, and so we now initialise *Billy* with $\mathtt{b}$, *slaughterhouse 5* with $\mathtt{h}$, and *survived* with $\mathtt{s}$. The "sensible inference" we posed was to determine whether $\mathtt{b} \to \mathtt{s}$ should—under some reasonable account of consequence—follow from the set $\{\mathtt{b} \to \mathtt{h},\ \mathtt{h} \to \mathtt{s}\}$.

1.2.4 Deductive Systems

Logical consequence, described in § 1.2.3, offers a purely semantic account of how it might be inferred that one formula follows (logically) from another set thereof. In contrast, deductive systems answer this question syntactically by describing a system of *axiomata* and *rules of inference*.

Definition 7. A *deductive system* is a collection of axiomata and rules of inference. A *proof* in such a system is a sequence of formulae where each formula is either an axiom, or has been inferred by application of an inference rule to previous formulae in the sequence. The final formula in the sequence, ϕ, is called the *theorem* and is *provable*, and so we write $\vdash \phi$.

Deductive systems offer some advantages over their semantic counterparts; particularly, when reasoning over large—possibly infinite—domains, logical consequence can become difficult. Moreover, semantic consequence provides little insight into the relationships between pieces of information that lead to the inferences we make; while the sequential nature of deduction systems trace a path describing this relationship [3, p. 55].

1.2.4.1 Hilbert Systems

A propositional Hilbert system $\mathcal{H}$ is characterised by three axiom schemata,

$$\text{(Axiom 1)} \quad \vdash (\phi \to (\psi \to \phi)), \tag{1.12}$$

$$\text{(Axiom 2)} \quad \vdash (\phi \to (\psi \to \gamma)) \to ((\phi \to \psi) \to (\phi \to \gamma)), \tag{1.13}$$

$$\text{(Axiom 3)} \quad \vdash (\neg\phi \to \neg\psi) \to (\phi \to \psi), \tag{1.14}$$

and a single rule of inference:

$$\text{(Modus Ponens)} \quad \frac{\vdash \phi, \qquad \vdash \phi \to \psi}{\vdash \psi}. \tag{1.15}$$

The axiom schemata themselves are not axioms, but rather patterns containing meta-variables that, when uniformly substituted for formulae, result in an axiom. The turnstile symbol ($\vdash$) is the syntactic counterpart to the double-turnstile ($\models$) used for logical consequence. We frequently express that ϕ is a theorem by writing $\vdash \phi$ [3, p. 55].

As it stands, constructing proofs from instances of axiom schema and applications of modus ponens is a challenging ordeal. *Derived rules* are introduced as a means to make it easier to spot the next step in a proof sequence. Of particular importance is the so called *deduction theorem*; which allows the construction of proofs that are conditioned on a hypothesis, without requiring that the hypothesis be an axiom.

$$\text{(Deduction Theorem)} \quad \frac{\Delta \cup \phi \vdash \psi}{\Delta \vdash \phi \to \psi} \tag{1.16}$$

Derived rules must be sound with respect to what can be proved by the three axioms and applications of modus ponens. That is, a derived rule should not enable one to make an inference that would not be possible without such a rule.

As an illustration, we provide the following Hilbert-style proof for the inference made in Example 1.

Theorem 1. $b \to h,\ h \to s \vdash\ b \to s$

Proof.

1. $\vdash (\mathtt{b} \to \mathtt{h})$	(Premise)
2. $\vdash (\mathtt{h} \to \mathtt{s})$	(Premise)
3. $\vdash (\mathtt{h} \to \mathtt{s}) \to (\mathtt{b} \to (\mathtt{h} \to \mathtt{s}))$	(Axiom 1)
4. $\vdash (\mathtt{b} \to (\mathtt{h} \to \mathtt{s}))$	(MP 2,3)
5. $\vdash (\mathtt{b} \to (\mathtt{h} \to \mathtt{s})) \to ((\mathtt{b} \to \mathtt{h}) \to (\mathtt{b} \to \mathtt{s}))$	(Axiom 2)
6. $\vdash (\mathtt{b} \to \mathtt{h}) \to (\mathtt{b} \to \mathtt{s})$	(MP 4,5)
7. $\vdash (\mathtt{b} \to \mathtt{s})$	(MP 1,6)

□

1.2.4.2 Gentzen Systems

1.2.5 Soundness and Completeness

1.2.6 Consequence Relations

Chapter 2

Formal Concept Analysis

Formal Concept Analysis (FCA) provides a simple, and yet mathematically rigorous, framework for identifying and reasoning about "concepts" and their corresponding hierarchies in data [2, 6]. The central view of concepts as a dual between *extension*—what one refers to as instances of a concept—and *intension*—what meaning is ascribed to a concept—is supported by a rich philosophical backing.

2.1 Basic Notions

The universe of discourse in FCA is made-up of sets of *objects* and *attributes*. Objects are extensional, they are the things pointed to as instances of some more general concept. In turn, attributes construct the intensional component of a concept.

We collect these extensional and intensional building blocks in a structure called a *formal context*, which includes a binary relation that allows traversal from extension to intension, and vice verse.

Definition 8. A *formal context* $\mathbb{K} = (G, M, I)$ is a triple comprised of a set of objects G, a set of attributes M, and a binary relation $I \subseteq G \times M$ referred to as an 'incidence' relation. For an object-attribute pair $(g, m) \in I$ we might say that "object g *has* the attribute m".

A formal context in some sense describes an open-world interpretation, and so $(g, m) \notin I$ is not usually interpreted as saying that "object g has the negation of the attribute m".

Example 2. Finite formal contexts of a reasonable size can be described entirely by a tabular representation. Each object corresponds to a row, and each attribute to a column.

	closure	associativity	identity	divisibility	commutativity
magma	×				
semigroup	×	×			
monoid	×	×	×		
group	×	×	×	×	
abelian group	×	×	×	×	×
loop	×		×	×	
quasigroup	×			×	
groupoid		×	×	×	
category		×	×		
semicategory		×			

Figure 2.1: A formal context showing necessary properties of group-like structures.

The tabular representation of a formal context allows for easy identification of the set of attributes that a given object satisfies: one need only scan across the respective row in the table and note where an '×' symbol appears. This set of attributes is called the *object intent*. The dual notion of an *attribute extent* can similarly be found by scanning down the column of a given attribute. The utility of this visual metaphor diminishes when, as is often the case, we consider larger sets of objects or attributes. Instead, we opt for a more formal approach to determining the intents and extents for (sets of) objects and attributes, respectively.

Definition 9. Given a formal context (G, M, I), the *derivation operators* are two order-reversing maps $(\cdot)^{\uparrow} : 2^G \to 2^M$ and $(\cdot)^{\downarrow} : 2^M \to 2^G$ where the order is given by subset inclusion. Then, for any subsets $A \subseteq G$ and $B \subseteq M$,

$$A^{\uparrow} := \{m \in M \mid \forall g \in A,\ \langle g, m\rangle \in I\}$$
$$B^{\downarrow} := \{g \in G \mid \forall m \in B,\ \langle g, m\rangle \in I\}$$

The derivation operators provide a clear way of describing, for a given set $A \subseteq G$ of objects, the set of attributes which every object in A satisfies, denoted $A^{\uparrow}$. As an illustration, given the set of objects {`semigroup, monoid`} from Figure 2.1, its derivation would be {`closure, associativity`}. It is quite easy to spot that this is just the intersection of the object intents of `semigroup` and `monoid`.

Of course, these two functions can be composed; and so, $A^{\uparrow\downarrow}$ would yield the set of objects {`semigroup, monoid, group, abelian group`}, which can rather cumbersomely be described as "the set of all objects that satisfy all the attributes satisfied by `semigroup` and `monoid`".

In fact, this double-application of derivation operators satisfies very specific properties,

$$\text{(monotonicity)} \qquad A \subseteq A_1 \textit{ implies } A^{\uparrow\downarrow} \subseteq A_1^{\uparrow\downarrow} \tag{2.1}$$

$$\text{(extensivity)} \qquad A \subseteq A^{\uparrow\downarrow} \tag{2.2}$$

$$\text{(idempotency)} \qquad A^{\uparrow\downarrow} = (A^{\uparrow\downarrow})^{\uparrow\downarrow} \tag{2.3}$$

for all $A, A_1 \subseteq G$. Thus, $(\cdot)^{\uparrow\downarrow}$ describes a closure operator on 2^G. The dual notion holds for attributes, and $(\cdot)^{\downarrow\uparrow}$ describes a closure operator on 2^M.

Proposition 1. Let (G, M, I) be a formal context with subsets $A_0, A_1, A_2 \subseteq G$ and $B_0, B_1, B_2 \subseteq M$ of attributes. Then,

1. $A_0 \subseteq A_1 \Rightarrow A_1^{\uparrow} \subseteq A_0^{\uparrow}$
2. $A_0 \subseteq A_0^{\uparrow\downarrow}$
3. $A_0^{\uparrow} = A_0^{\uparrow\downarrow\uparrow}$

1. $B_0 \subseteq B_1 \Rightarrow B_1^{\downarrow} \subseteq B_0^{\downarrow}$
2. $B_0 \subseteq B_0^{\downarrow\uparrow}$
3. $B_0^{\downarrow} = B_0^{\downarrow\uparrow\downarrow}$

Definition 10. A *formal concept* of a formal context (G, M, I) is a pair (A, B) of subsets $A \subseteq G$ and $B \subseteq M$ that satisfies $A^{\uparrow} = B$ and $B^{\downarrow} = A$. Then, A is the concept *extent* and B is the *intent*. We write $\mathfrak{B}(G, M, I)$ to denote the set of all concepts of (G, M, I).

Now explain why derivation of set of objects is always a concept intent by proposition 1. And the dual. and Galois connections

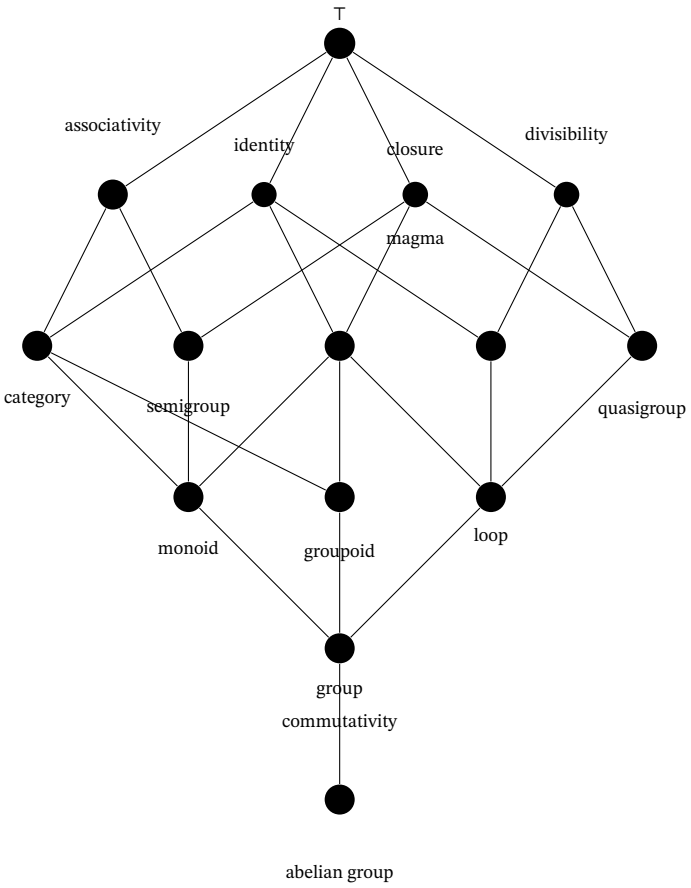


Figure 2.2: The concept lattice associated with the formal context in Figure 2.1

Chapter 3

Non-Monotonic Reasoning

Hello there my dog A wfaew

Part II

Rational Concept Analysis

Chapter 4

Defeasible Reasoning in Formal Concept Analysis

Chapter 5

Rational Concepts

Bibliography

[1] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*. Cambridge University Press, 2002.

[2] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*. Springer Berlin, Heidelberg, 1999.

[3] M. Ben-Ari, *Mathematical logic for computer science.* Prentice Hall, 1993, vol. 13, pp. I–XI, 1–305.

[4] M. Huth and M. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*, 2nd ed. Cambridge University Press, 2004.

[5] A. R. Turquette, A. Tarski, and J. H. Woodger, *Logic, Semantics, Metamathematics*. Oxford University Press (OUP), 1957, vol. 8, p. 87.

[6] B. Ganter and S. Obiedkov, *Conceptual Exploration*. Springer Berlin Heidelberg, 2016, pp. 1–315.

Index