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

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 that $(x, y) \in R$ 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, in this context writing "element x is preferred to y" should be interpreted to mean that $(x, y) \in \preceq$, or simply $x \preceq y$. The use of the metaphor will become more apparent in Chapter 3.

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 may 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:

$$\text{(Irreflexivity)} \quad x \not\prec x \tag{1.4}$$

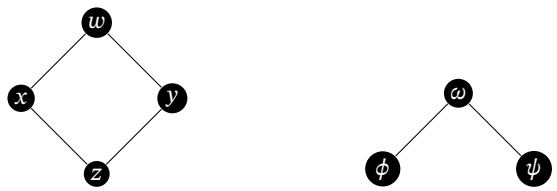
$$\text{(Asymmetry)} \quad x \prec y \text{ implies } y \not\prec x \tag{1.5}$$

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

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, we may reference the order relation associated with $\mathbf{X}$ by writing $\preceq_X$ in settings where there is ambiguity. 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 visually describe 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 certain properties involving 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 $\mathbf{Y}$ (the supremum and infinum are also referred to as the *least upper bound* and *greatest lower bound*, respectively). 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 $\bigvee \mathbf{Y}$. Instead of infimum, we say *meet* and write $x \wedge y$, or $\bigwedge \mathbf{Y}$ [1, p. 33].

With these definitions of meets and joins in mind, we are able to define a lattice as

Definition 4. Given a partially ordered set $\mathbf{L}$, we say that $\mathbf{L}$ is a *lattice* if for every pair $x, y \in \mathbf{L}$ the join $x \vee y$ and meet $x \wedge y$ exist, and are unique. Then, we call $\mathbf{L}$ a *complete lattice* if for every subset $\mathbf{M} \subseteq \mathbf{L}$ both $\bigvee \mathbf{M}$ and $\bigwedge \mathbf{M}$ exist in $\mathbf{L}$.

We say that a lattice $\mathbf{L}$ is *bounded* if there exists an element $x \in \mathbf{L}$ such that $x \vee y = x$ for all $y \in \mathbf{L}$, and there exists an element $z \in \mathbf{L}$ with $z \wedge y = z$ for all $y \in \mathbf{L}$. Frequently we refer to such an x and y as a top and bottom element and denote them by $\top$ and $\bot$, respectively. It is not difficult to spot that every complete lattice is bounded—in fact this is true by definition of a complete lattice—as a corollary every finite lattice is also bounded.

The lattice $(\mathbb{Z}, \leq)$ constructed by imposing the natural order over the integers is an example of an lattice that is not bounded [1, p. 36].

Example 1. The set of natural numbers $\mathbb{N}$ forms a complete lattice when ordered by divisibility sometimes called the *division lattice*. In the division lattice, the join operation corresponds to the *least common multiple* and the meet operation to the greatest common divisor. The bottom element of this lattice is 1 as it divides every other natural, while the top element is 0 since it is divisible by all other naturals. Although the division lattice is infinite, it is indeed a complete lattice.

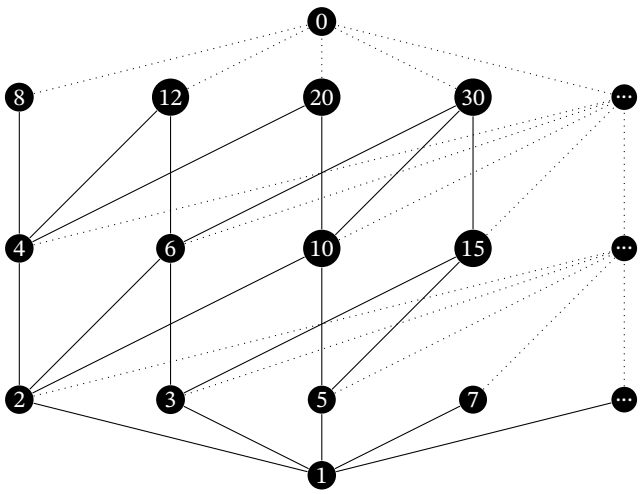


Figure 1.2: The division lattice $(\mathbb{N}, \leq)$

1.1.2.1 Lattices as algebraic structures

From another perspective, a lattice can be viewed 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 this perspective.

If the set L is equipped with just one of these operations, the structure is called a *semilattice*. Specifically, the structures $\langle L, \vee \rangle$ and $\langle L, \wedge \rangle$ are called a *join semilattice* and *meet semilattice*, respectively. Such structures are reducts 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:

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

Talk about the duality principle of proving things for $\vee$ and $\wedge$. Talk about distributive lattices

The connection between partially ordered sets and lattices is quite straightforward; however, that the order-theoretic and algebraic views of lattices do in fact coincide is shown by the lemma below.

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 maintain a rigid distinction between these two perspectives on lattices; 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 sentence like "Tralfamadorians have one eye". That is, something that can be assigned a value of *true* or *false* [3, p. 7]. More complex propositions may be constructed recursively from simpler ones. The main interest is to then discover which true propositions follow—under some agreeable sense of what it means to 'follow'—from others.

1.2.1 Syntax

Propositional atoms are the fundamental building blocks of a propositional language. They are indivisible statements that can either be true or false, and may be be combined with the boolean connectives $\{\neg, \vee, \wedge, \rightarrow\}$ to construct more complex statements, or *formulae*. We denote propositional atoms with lower-case Latin letters p, q, r, s, and t, and the set of all atoms will usually be denoted by $\mathcal{P}$. Lower-case Greek letters $\alpha, \beta, \gamma, \phi, \psi$, and φ will be used to denote formulae, and corresponding upper-case Greek letters will usually represent sets of formulae.

Not all combinations of atoms and boolean connectives result in meaningful expressions. For example, '$\wedge \rightarrow \phi$', which can be parsed as "conjunction materially implies ϕ", has no discernible meaning; so we make a distinction between any formulae and so-called *well-formed formulae*. The latter being constructions that are valid with respect to the rules of the grammar defined in Backus-Naur form below [4, p. 33].

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

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]. Any formula accepted by this grammar is said to be in the language $\mathcal{L}^{\mathcal{P}}$, although we may also 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 suggesting a more interesting domain, such as $\texttt{tralfamadorians} \to \neg\texttt{human}$, which should be interpreted as proposing that "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 may 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 atom in the set $\mathcal{P}$ of atoms, sometimes called the *universe of discourse*.

Definition 5. A *valuation* is a function $u : \mathcal{P} \to \{\, \textit{true}, \textit{false} \,\}$ 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*, and so 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 Sub-section 1.2.1. For any $\phi, \psi \in \mathcal{L}$

- $u \Vdash \neg\phi$ if and only if $u \nVdash \phi$ (negation)
- $u \Vdash \phi \vee \psi$ if and only if $u \Vdash \phi$ or $u \Vdash \psi$ (disjunction)
- $u \Vdash \phi \wedge \psi$ if and only if $u \Vdash \phi$ and $u \Vdash \psi$ (conjunction)
- $u \Vdash \phi \to \psi$ if and only if $u \Vdash \neg\phi$ or $u \Vdash \psi$ (material implication)
- $u \Vdash \phi \leftrightarrow \psi$ if and only if $u \Vdash \phi \to \psi$ and $u \Vdash \psi \to \phi$ (material equivalence)

Definition 6. 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 all 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—in terms of the semantics—to the question of when it is appropriate to infer one formula from another [5, p. 408].

For example, if we know that "*Billy Pilgrim 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 Pilgrim 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. Informally, it might help ones' intuition to try imagine a world where the first propositions are true while the final one false.

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 form 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. The distinction is clearer in Propositional logic: constructions derived from combinations of atoms and the boolean operators result in elements of the object language; while the metalanguage uses symbols like $\vDash$ and $\vdash$, which are introduced below.

Definition 7. 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 $\hat{\Gamma} \subseteq \hat{\varphi}$.

This semantic account of consequence, due to Tarski [5, p. 417], makes implicit the view that if φ is in fact a consequence of Γ, then it should not be possible for all the sentences (formulae) in Γ to be true while φ be false.

Example 2. If we modelled the earlier example in propositional logic, we might initialise *Billy Pilgrim* with $\mathtt{b}$, *slaughterhouse 5* with $\mathtt{h}$, and *survived* with $\mathtt{s}$. It is then our aim to determine whether $\{\mathtt{b} \to \mathtt{h},\ \mathtt{h} \to \mathtt{s}\} \vDash \mathtt{b} \to \mathtt{s}$. From the satisfaction relation in Sub-section 1.2.2 it can be derived that the models of $\{\mathtt{b} \to \mathtt{h},\ \mathtt{h} \to \mathtt{s}\}$ are precisely

$$\{\{\overline{\mathtt{b}},\ \overline{\mathtt{h}},\ \overline{\mathtt{s}}\}, \{\overline{\mathtt{b}},\ \overline{\mathtt{h}},\ \mathtt{s}\}, \{\overline{\mathtt{b}},\ \mathtt{h},\ \mathtt{s}\}, \{\mathtt{b},\ \mathtt{h},\ \mathtt{s}\}\}.$$

which are indeed all models of $\mathtt{b} \to \mathtt{s}$, and so we answer the question in the affirmative.

Consequence operators are functions over a given language that map sets of formulae to their consequences. They were introduced by Tarski [6, p. 84] as the symbol $\mathcal{C}n$ representing a *general* idea that sets can be closed under a specified notion consequence. In future, we use $\mathcal{C}n$ to refer specifically to closure under logical (Tarskian) consequence; and so, application of the $\mathcal{C}n$ operator to a set Γ would yield $\mathcal{C}n(\Gamma) := \{\varphi \mid \Gamma \vDash \varphi\}$. Where we wish to reference closure under a different notion of consequence, suppose that of a Hilbert system, we will use a subscript to denote the system $\mathcal{C}n_{\mathcal{H}}$. To refer to the general notion of closure under *some* notion of consequence, we write $\mathcal{C}n_X$ [7, p. 4].

A *theory* (also called a *deductive system*, but we avoid this terminology) is a set of formulae Γ that equals its closure $\mathcal{C}n_X(\Gamma)$. The study of such operators is useful in its ability to reveal properties about the respective notion of consequence. For instance, it was observed by Tarski [6, p. 84] that $\mathcal{C}n$ satisfies the following properties:

$$\text{(inclusion)} \quad \Gamma \subseteq \mathcal{C}n(\Gamma) \tag{1.12}$$

$$\text{(idempotency)} \quad \mathcal{C}n(\Gamma) = \mathcal{C}n(\mathcal{C}n(\Gamma)) \tag{1.13}$$

$$\text{(monotonicity)} \quad \Gamma \subseteq \Gamma' \Rightarrow \mathcal{C}n(\Gamma) \subseteq \mathcal{C}n(\Gamma') \tag{1.14}$$

Inclusion is a relatively easy property to justify: all that is required is that the consequences of some information includes at least that starting information. *Idempotency* requires that the supposed set of all consequences is in fact the set of *all* consequences.

1.2.4 Deductive Systems

Logical consequence, described in Sub-section 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* that affirm the truth of one formula from the truth of others [3, p. 49].

Definition 8. 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 then *provable*, and so we write $\vdash \phi$.

A *theory* (or, *deductively closed theory*) in a deduction system is a set of formulae closed under application of axioms and inference rules of the system; and so a set of formulae Γ is a theory if it is equal to its deductive closure $\mathcal{C}n_{\mathcal{H}}(\Gamma) := \{\varphi \mid \Gamma \vdash \varphi\}$. As before, elements of a theory are called theorems.

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,

$$\text{(Axiom 1)} \quad \vdash (\phi \to (\psi \to \phi)), \tag{1.15}$$

$$\text{(Axiom 2)} \quad \vdash (\phi \to (\psi \to \gamma)) \to ((\phi \to \psi) \to (\phi \to \gamma)), \tag{1.16}$$

$$\text{(Axiom 3)} \quad \vdash (\neg\phi \to \neg\psi) \to (\phi \to \psi), \tag{1.17}$$

and a single rule of inference:

$$\text{(Modus Ponens)} \quad \frac{\vdash \phi, \qquad \vdash \phi \to \psi}{\vdash \psi}. \tag{1.18}$$

The axiom schemata themselves are not axioms, but rather patterns containing meta-variables that, when uniformly substituted for formulae, result in an instantiated axiom. The turnstile symbol ($\vdash$) is the syntactic counterpart to the double-turnstile ($\vDash$) 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. As an illustration, we provide the following Hilbert-style proof for the inference made in Example 2.

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)

□

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 facilitates the construction of proofs that are conditioned on a hypothesis, without requiring that the hypothesis be an axiom.

$$\text{(Deduction Theorem)} \quad \frac{\Delta \cup \phi \vdash \psi}{\Delta \vdash \phi \to \psi} \tag{1.19}$$

As an illustration, the proof of Theorem 1 can be restated using the *transitivity* derived rule:

$$\text{(Transitivity)} \quad \frac{\Delta \vdash \varphi \to \psi, \qquad \Delta \vdash \psi \to \gamma}{\Delta \vdash \varphi \to \gamma} \tag{1.20}$$

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.

The fact that it was able to be shown that "Billy Pilgrim survived the bombing of Dresden" through both semantic and syntactic notions of consequence is not a coincidence. This correspondence between logical consequence and Hilbert-style deduction systems holds in general for propositional logic (and, in fact for many other systems with relatively limited expressive power). We capture this correspondence more formally through the notions of *soundness* and *completeness*.

Definition 9. Given a set of formulae Γ and one γ in $\mathcal{L}$, $\Gamma \vdash \gamma$ if and only if $\Gamma \vDash \gamma$. Where $\vDash$ and $\vdash$ are used in the context with which they have been introduced.

The proofs for Definition 9 are well-known, and can be found in [3, p. 64].

1.2.5 Consequence Relations

In the preceding sections the discussions on consequence—be they either semantic or syntactic—have been concrete realisations of the more general idea of *consequence relations*. When discussing consequence relations we will abuse notation and denote the relation with '$\vdash$' (relying on the surrounding context to distinguish between consequence relations and syntactic derivations). Any particular consequence relation will be denoted by a subscript referencing the system.

Definition 10. A *consequence relation* $\vdash \subseteq \mathcal{L} \times \mathcal{L}$ is a binary relation over a formal language that satisfies

$$\text{(reflexivity)} \quad \varphi \in \Gamma \text{ implies } \Gamma \vdash \varphi \tag{1.21}$$

$$\text{(monotonicity)} \quad \Gamma \vdash \varphi \text{ and } \Gamma \subseteq \Psi \text{ imply } \Psi \vdash \varphi \tag{1.22}$$

$$\text{(cut)} \quad \Gamma \vdash \varphi \text{ and } \Psi, \varphi \vdash \psi \text{ imply } \Gamma, \Psi \vdash \psi \tag{1.23}$$

We can view a consequence relation as a set of ordered pairs $\{(\Gamma_0, \varphi_0), \dots, (\Gamma_n, \varphi_n), \dots\}$; in a pair (Γ, φ) Γ represents a premise, and φ a conclusion [7, p. 16]. Then, the presence of said pair describes that φ is a consequence of Γ. The absence of such a pair is understood to mean that φ is not a consequence of Γ.

This is a much more significant abstraction on the matter of consequence, requiring no concrete proof system or semantics.

As we will see in Chapter 3, consequence relations provide a useful abstraction which allow a logic to be described in terms of certain properties, or *postulates*, corresponding to a certain desired behaviour. The properties are able to provide an intuition for the logic without the construction of a semantics, or deduction-system.

Chapter 2

Formal Concept Analysis

Formal Concept Analysis (henceforth initialised as FCA) provides a mathematical framework for reasoning about concepts and their corresponding conceptual structures. It is an example of applied lattice theory, and benefits from rich, rigorous mathematical theory, as well as an easily graspable intuition.

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 11. 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 3. 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.

Algebraic Structures	closure	associativity	identity	divisibility	commutativity
magma	×				
semigroup	×	×			
monoid	×	×	×		
group	×	×	×	×	
abelian group	×	×	×	×	×
loop	×		×	×	
quasiagroup	×			×	
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 12. 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 13. 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

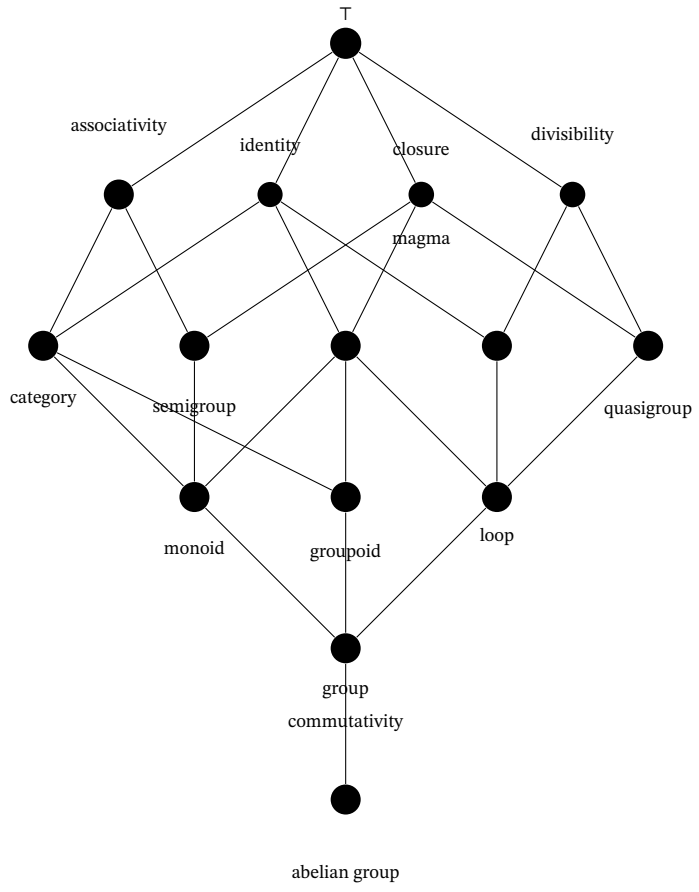


Figure 2.2: The concept lattice associated with the formal context in Figure 2.1

2.2 Contextual Attribute Logic

The logic that underlies FCA [8, 9]

Chapter 3

Non-Monotonic Reasoning

Hello there my dog A wfaew

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. Tarski, "On the concept of logical consequence," in *Logic, Semantics, Metamathematics*, J. H. Woodger, Ed., Originally published in 1936, Oxford University Press, 1956, pp. 409–420.

[6] A. Tarski, "Fundamental concepts of the methodology of the deductive sciences," in *Logic, Semantics, Metamathematics*, J. H. Woodger, Ed., Originally published in 1936, Oxford University Press, 1956, pp. 60–109.

[7] A. Citkin and A. Muravitsky, *Consequence relations: An introduction to the Lindenbaum-Tarski method*. Oxford University Press, 2022.

[8] B. Ganter, "The language of Contextual Attribute Logics – Introduction and survey," *International Journal of Approximate Reasoning*, vol. 186, p. 109 487, 2025.

[9] B. Ganter and R. Wille, "Contextual Attribute Logic.," in *International Conference on Conceptual Structures (ICCS)*, 1999, pp. 377–388.

Index

Atom, 5
Axiom, 8
Axiom scheme, 8

binary relation, 2
Boolean operator, 5
    and, 5
    associativity, 4
    commutativity, 4
    idempotency, 4
    material implication, 5
    negation, 5
    or, 5

completeness
    propositional logic, 9
consequence relations, 9

Deductive system, 8
    Hilbert system, 8
derivation operators
    attribute extent, 12
    object intent, 12

extension, 11

formal concept, 13
formal concept analysis, 11
formal context, 11

Galois connections, 13
Genzten systems, 9

Hasse diagrams, 3

intension, 11

join, 3

Logical consequence
    propositional logic, 6

lower bound, 3

map
    order-embedding, 3
    order-isomorphism, 3
    order-preserving, 3
maximal, 3
maximum, 3
meet, 3
metalanguage, 7
minimal, 3
minimum, 3
model, 6

object language, 7
order-reversing map, 3

partial-order, 2
    strict partial-order, 3
Propositional atom, 5

Rule of inference, 8
    modus ponens, 8
    deduction, 9

satisfaction
    propositional logic, 6
semantics
    propositional logic, 6
syntax
    propositional logic, 5

upper bound, 3

valuations, 6