

# Thesis - outline

Sasha Lifchez

February 2026

## 1 Introduction

### 1.1 Overview of central theme: intersectional analysis of algebra, topology, and logic

## 2 Basics of Logic

### 2.1 Propositional logic

#### 2.1.1 The Language of Propositional Logic

Propositional logic is the branch of logic examining propositions—that is, sentences which must be either true or false—and the ways they can be combined [1]. The language of propositional logic consists of atomic variables over propositions, denoted  $p_1, p_2, p_3, \dots$ , and a number logical connectives, which may be thought of as functions taking propositions as input and returning a single proposition as an output. The number of propositions taken by a connective as input is called the *arity* of the connective; a connective of arity  $n$  may also be called an  $n$ -ary connective.

Formulas of propositional logic are defined recursively as follows:

1. Every atomic propositional variable ( $p_1, p_2, p_3, \dots$ ) is a formula, called an *atomic formula*.
2. If  $c$  is a logical connective of arity  $n$ , and  $(P_1, P_2, \dots, P_n)$  a sequence of  $n$  formulas, then  $c(P_1, P_2, \dots, P_n)$ , the result of applying the connective  $c$  to the formulas  $(P_1, \dots, P_n)$ , is a formula.
3. Nothing else is a formula.

The standard logical connectives are the binary connectives  $\wedge, \vee, \supset$  and the unary connective  $\neg$ , which will be defined shortly (other logical systems may use different connectives; this is beyond the scope of this paper). The propositional constants  $\top$  (representing "true") and  $\perp$  (representing "false") are also often used, and may be considered as 0-ary logical connectives.

By convention, we write  $P \wedge Q$  in place of  $\wedge(P, Q)$ ;  $\vee$  and  $\supset$  are treated similarly.

By imposing a finite number of axioms and rules of inference, we obtain a proof system for propositional logic. By a *proof* we mean a finite sequence of formulas, each of which is either an axiom or follows from prior formulas in the sequence by a rule of inference; the final entry of the proof is called a *theorem*. If  $A$  is a theorem of propositional logic, we write  $\vdash_{PL} A$ ; if there is no ambiguity as to which logical system we are using, we simply write  $\vdash_A$ .

A formula  $A$  can be proven from a set  $\Gamma$  of premises, where  $\Gamma = \{B_1, B_2, B_3, \dots\}$  for some formulas  $B_1, B_2, B_3, \dots$  of propositional logic, if there exists a finite sequence of formulas, each of which is an axiom, is a member of  $\Gamma$ , or follows from prior formulas in the sequence by a rule of inference, whose last line is  $A$ . In this case we write  $\Gamma \vdash_{PL} A$ .

### 2.1.2 Semantics of Propositional Logic

A *valuation* is a function from the set of formulas of a language  $\mathcal{L}$  to the set  $\mathcal{V}$  of truth values; here,  $\mathcal{V} = \{0, 1\}$ , where 1 denotes *true* and 0 denotes *false* [8]. A valuation  $v$  is first defined for each atomic formula of  $\mathcal{L}$ , and is subsequently defined recursively as follows: for two formulas  $P$  and  $Q$ ,

1.  $v(\neg P) = 1$  if and only if  $v(P) = 0$
2.  $v(P \wedge Q) = 1$  if and only if  $v(P) = 1$  and  $v(Q) = 1$
3.  $v(P \vee Q) = 1$  if and only if  $v(P) = 1$  or  $v(Q) = 1$
4.  $v(P \supset Q) = 1$  if and only if  $v(P) = 0$  or  $v(Q) = 1$
5.  $v(\top) = 1$
6.  $v(\perp) = 0$

Given a formula  $A$  of  $\mathcal{L}$ , if  $v(A) = 1$ , we say that the interpretation  $I$  *models*  $A$ , and we write  $\models A$ . A formula is *satisfiable* if at least one valuation  $v$  models it; a set of formulas is satisfiable if at least one valuation models every formula in the set.

A formula of  $\mathcal{L}$  which is true under any valuation is called a *tautology*. As an example, if  $P$  is an atomic formula of  $\mathcal{L}$ , then the formula  $P \supset P$  is a tautology: by the above,  $v(P \supset P) = 1$  if and only if  $v(P) = 1$  or  $v(P) = 0$ . Since any valuation function maps any atomic formula  $P$  to either 0 or 1, we must have either  $v(P) = 1$  or  $v(P) = 0$ , and thus  $v(P \supset P) = 1$ .

## 2.2 First-Order logic

### 2.2.1 The Language of First-Order Logic

*First-order logic* is a branch of formal logic which involves all the properties of propositional logic, as well as quantifiers and predicates[7]. Like propositional logic, the language of first-order logic includes variables and logical connectives. In particular, a language  $\mathcal{L}$  of first-order logic consists of

1. The logical connectives  $\wedge, \vee, \supset$  and  $\neg$
2. The quantifier symbols  $\forall$  and  $\exists$
3. Constant symbols  $a_1, a_2, a_3, \dots$
4. Variables  $x, y, z, \dots$  ranging over the set of constant symbols
5. For each natural number  $n$ , function symbols  $f_0^n, f_1^n, f_2^n, \dots$
6. For each natural number  $n$ , predicate symbols  $P_0^n, P_1^n, P_2^n, \dots$

A function which takes  $n$  arguments is said to have *arity*  $n$ . A predicate which takes  $n$  arguments is called an  *$n$ -place predicate*. Some systems of first-order logic also include the identity symbol,  $=$ ; we will omit this for now.

The set of *terms* of a language  $\mathcal{L}$  of first-order logic is defined as follows:

1. The *constants*  $(a_1, a_2, a_3, \dots)$  of  $\mathcal{L}$  are terms.
2. The *variables*  $(x, y, z, \dots)$  of  $\mathcal{L}$  are terms.
3. If  $f$  is a function symbol of arity  $n$  and  $t_1, t_2, \dots, t_n$  are terms, then  $f(t_1, t_2, \dots, t_n)$  is a term.
4. Nothing else is a term.

The set of *formulas* of  $\mathcal{L}$  is defined recursively, much like it was for propositional logic:

1. If  $t_1, t_2, \dots, t_n$  are terms of  $\mathcal{L}$  and  $P$  an  $n$ -place predicate, then  $P(t_1, t_2, \dots, t_n)$  is a formula of  $\mathcal{L}$ , called an *atomic formula*.
2. If  $\varphi$  is a formula of  $\mathcal{L}$ , then so is  $\neg\varphi$ .
3. If  $\varphi$  and  $\psi$  are formulas of  $\mathcal{L}$ , then so are  $\varphi \wedge \psi$ ,  $\varphi \vee \psi$ , and  $\varphi \supset \psi$ .
4. If  $\varphi$  is a formula of  $\mathcal{L}$  and  $x$  a variable, then  $\forall x\varphi$  and  $\exists x\varphi$  are formulas of  $\mathcal{L}$ .
5. Nothing else is a formula.

### 2.2.2 Semantics of First-Order Logic

The semantics of first-order logic is somewhat more complicated than that of propositional logic. An *interpretation* of a first-order language  $\mathcal{L}$  is a structure  $M = \langle d, I \rangle$ , where  $d$  is a non-empty set (called the *domain*), and  $I$  is an *interpretation function*. This model is constructed as follows:

1. For every constant  $c$ ,  $I(c) \in d$
2. For every  $n$ -place predicate  $P$ ,  $I(P)$  is a set of ordered  $n$ -tuples of elements of  $d$ , i.e. a subset of  $d^n$

A *variable assignment*  $s$  on an interpretation  $M$  is a function from the set of variables to the set  $d$ . The *denotation* of a term  $t$ , written  $D_{M,s}(t)$ , is defined as  $I(t)$  if  $t$  is a constant, and  $s(t)$  if  $t$  is a variable.

The notion of *satisfaction* in a model  $M$  is similar to the idea of valuation in propositional logic. Given a formula  $\phi$ , an interpretation  $M$ , and a variable assignment  $s$ , we write  $M, s \models \phi$  to represent  $M$  *satisfies*  $\phi$  *under*  $s$ , and define satisfaction recursively:

1. If  $P$  is an  $n$ -place predicate and  $t_1, t_2, \dots, t_n$  are terms, then  $M, s \models S(t_1, t_2, \dots, t_n)$  if and only if  $\langle D_{M,s}(t_1), D_{M,s}(t_2), \dots, D_{M,s}(t_n) \rangle \in I(P)$ .

For any two formulas  $\theta, \psi$ , we have:

2.  $M, s \models \neg\theta$  if and only if  $M, s \not\models \theta$  (i.e.  $M$  does *not* satisfy  $\theta$  under  $s$ )
3.  $M, s \models \theta \wedge \psi$  if and only if  $M, s \models \theta$  and  $M, s \models \psi$
4.  $M, s \models \theta \vee \psi$  if and only if  $M, s \models \theta$  or  $M, s \models \psi$
5.  $M, s \models \theta \supset \psi$  if and only if  $M, s \not\models \theta$  or  $M, s \models \psi$
6.  $M, s \models \forall x\theta$  if and only if  $M, s' \models \theta$  for *every* assignment  $s'$  which agrees with  $s$  except possibly at  $x$
7.  $M, s \models \exists x\theta$  if and only if  $M, s' \models \theta$  for *some* assignment  $s'$  which agrees with  $s$  except possibly at  $x$

A formula with no free variables is called a *sentence*. If  $\theta$  is a sentence, it can be shown that  $M, s' \models \theta$  for a particular variable assignment  $s'$  if and only if  $M, s \models \theta$  for *all* variable assignment  $s$ . Thus, if  $\theta$  is a sentence, we can write  $M \models \theta$  with no ambiguity. We say  $\theta$  is *satisfiable* if there exists some interpretation  $M$  such that  $M \models \theta$ . A set  $\Gamma$  of sentences is satisfiable if there exists some interpretation  $M$  such that  $M \models \theta$  for *every*  $\theta \in \Gamma$ ; in this case, we say  $M$  is a *model* of  $\Gamma$

For a language  $\mathcal{L}$ , we write  $\Gamma \models \theta$  to mean that for every interpretation  $M$  of  $\mathcal{L}$ , if  $M \models \psi$  for every  $\psi \in \Gamma$ , we must have  $M \models \theta$ . If  $\Gamma \models \theta$ , we say that  $\theta$  is a *model-theoretic consequence* of  $\Gamma$ .

As with propositional logic, we can obtain a deductive system for first-order logic by imposing various axioms and rules of inference. The following section will discuss the relationship between deductive systems and models.

## 3 Properties of Logical Systems

### 3.1 Soundness and Completeness

Given a first-order language  $\mathcal{L}$ , we assume a deductive system, denoted  $D$ .

First-order logic and propositional logic both hold the properties of *soundness* and *completeness*.

**Theorem** (Soundness). *For any sentence  $\theta$  and any set of sentences  $\Gamma$ , if  $\Gamma \vdash \theta$ , then  $\Gamma \models \theta$ .*

That is, if a sentence  $\theta$  can be proven from the premises  $\Gamma$  using a deductive system  $D$ , it must also be a semantic consequence of those premises.

**Theorem** (Completeness). *For any sentence  $\theta$  and any set of sentences  $\Gamma$ , if  $\Gamma \models \theta$ , then  $\Gamma \vdash \theta$ .*

That is, if a sentence  $\theta$  is a model-theoretic consequence of the sentences in  $\Gamma$ , it must also be possible to prove  $\theta$  from the sentences  $\Gamma$  via our deductive system  $D$ .

The proofs of soundness and completeness for first-order logic are omitted here.

Examples of logic where soundness and completeness fail: second-order

## 3.2 Compactness Theorem

Another important property a logical system may have is *compactness*:

**Theorem** (Compactness). *A set of sentences  $\Gamma$  has a model if and only if every finite subset  $\Gamma_0 \subset \Gamma$  has a model.*

The forward direction of this theorem is straightforward to prove: If  $\Gamma$  has a model, then every sentence  $\theta$  in  $\Gamma$  is satisfiable. If  $\Gamma_0$  is a finite subset of  $\Gamma$ , then every sentence  $\theta$  in  $\Gamma_0$  is also a member of  $\Gamma$ , and thus satisfiable; by definition, this means  $\Gamma_0$  has a model.

In order to prove the backwards direction, we will first prove a different statement, then show that the desired result follows [6]:

**Theorem.** *For any set of sentences  $\Gamma$  and any sentence  $\theta$ , if  $\Gamma \models \theta$ , then there exists some finite subset  $\Gamma_0 \subset \Gamma$  such that  $\Gamma_0 \models \theta$ .*

*Proof.* This proof follows straightforwardly from soundness and completeness. Suppose  $\Gamma \models \theta$ . By completeness,  $\Gamma \vdash_D \theta$ . Proofs in our deductive system are finite by definition, and each line of this proof is either an axiom, a member of  $\Gamma$ , or a formula following from earlier lines in the proof by a rule of inference. So it must be possible to derive  $\theta$  from a finite subset of  $\Gamma$ , denoted  $\Gamma_0$ . Thus  $\Gamma_0 \vdash \theta$ , and by soundness,  $\Gamma_0 \models \theta$ .  $\square$

Now, we sketch a proof of the backwards direction of the compactness theorem: [6]

*Proof.* Let  $\Gamma$  be a (possibly infinite) set of sentences, and suppose  $\Gamma$  does *not* have a model.

Then, for any interpretation  $M$ , there exists some sentence  $\theta$  in  $\Gamma$  such that  $M \not\models \theta$ . Let  $\psi$  be any sentence; recall that  $M \models \psi$  if, for every  $M$  such that  $M \models \Gamma$ , we have  $M \models \psi$ . Therefore, it must be true that every  $M$  which models  $\Gamma$  also models  $\psi$  (since no such  $M$  exists). Thus, for every sentence  $\psi$ ,  $\Gamma \models \psi$ .

By the previous theorem, for every sentence  $\psi$ , there exists some finite subset  $\Gamma_0 \subset \Gamma$  such that  $\Gamma_0 \models \psi$ .

Let  $\psi$  be a sentence in  $\Gamma$ . Then  $\neg\psi$  is a sentence of first-order logic, so there exists some  $\Gamma_0 \subset \Gamma$  such that  $\Gamma_0 \models \psi$ .

Suppose  $\Gamma_0$  were satisfiable, and let  $M$  be a model such that  $M \models \Gamma_0$ . By the above paragraph, since  $\Gamma_0 \models \neg\psi$ ,  $M \models \neg\psi$ , and thus  $M \not\models \psi$  by definition of satisfaction for the negation of a sentence. However, since  $M$  models  $\Gamma_0$ , it should model every sentence in  $\Gamma_0$ , and so  $M \models \psi$ . This is impossible! So, no such  $M$  can exist; thus,  $\Gamma_0$  is a finite subset of  $\Gamma$  which is unsatisfiable.  $\square$

This proof is informal; a rigorous proof would rely on the notion of *consistency* and an alternate notion of completeness, both of which are omitted here for simplicity. Nonetheless, it is clear that compactness follows straightforwardly from soundness and completeness.

### 3.3 Comparison to Compact Spaces

In the field of topology, *compactness* is also defined as a property of topological spaces. To define compactness, we first define the *finite intersection property*:

**Definition 1** (The Finite Intersection Property). *For a (possibly infinite) collection of sets, the finite intersection property is the property that every finite subcollection of these sets has a nonempty intersection.*

Now, we can define compactness.

**Definition 2** (Compactness (Finite Intersection definition)). *A topological space  $X$  is called compact if, for every collection of closed subsets of  $X$  having the finite intersection property has a nonempty intersection. [6]*

The following is an alternative definition:

**Definition 3** (Compactness (Open Cover definition)). *A topological space  $X$  is called compact if every open cover of  $X$  has a finite subcover.*

Here, an *open cover* of  $X$  is a (possibly infinite) collection of open sets whose union contains  $X$ ; a *finite subcover* is a finite subcollection of these sets whose union contains  $X$ . It can be shown that the two definitions are equivalent.

The similarities between compactness of a topological space and compactness of a logic are immediate: if an infinite collection of items has some property, a finite subcollection must also have that property. In fact, we will show that the compactness theorem for propositional logic is equivalent to the claim that the valuation space of the logic is compact. [6].

Possible addition: k-compactness

## 4 Proving the Equivalence of Logical and Topological Compactness

In order to prove the equivalence of two definitions of compactness, we must somehow impose a topology on a propositional logic. [6]

A *topology* on a space  $X$  is a collection  $\tau$  of subsets of  $X$ , called *open sets*, satisfying the following properties [3]:

1.  $\emptyset$  and  $X$  belong to  $\tau$
2. Any union of sets in  $\tau$  belongs to  $\tau$ , including infinite and arbitrary unions.
3. Any finite intersection of sets in  $\tau$  belongs to  $\tau$

Let  $PL_k$  denote a propositional logic with  $k$  sentence letters, equipped with truth-functional connectives. We restate the compactness of  $PL_k$  as follows:

**Definition 4.** *We say a propositional logic  $PL_k$  is compact to mean that a set of sentences  $\Sigma$  of  $PL_k$  is satisfiable if and only if every finite subset of  $\Sigma$  is satisfiable; or, equivalently,  $\Sigma$  is unsatisfiable if and only if some finite subset of  $\Sigma$  is unsatisfiable.*

Let  $V_k$  denote the set of all valuations of  $PL_k$ . For a sentence  $\phi$  of  $PL_k$ , we define the set

$$U(\phi) = \{v \in V_k : v(\phi) = 1\}$$

In other words,  $U(\phi)$  denotes the set of valuations which map the sentence  $\phi$  to the value 1. Let  $\mathcal{B}$  denote the collection of all sets  $U(\phi)$  for all sentences  $\phi$  in  $PL_k$ .

For any two sentences  $\phi$  and  $\psi$  and any valuation  $v$ ,  $v(\phi \wedge \psi) = 1$  if and only if  $v(\phi) = 1$  and  $v(\psi) = 1$ . Thus,

$$\begin{aligned} U(\phi) \cap U(\psi) &= \{v \in V_k : v(\phi) = 1\} \cap \{v \in V_k : v(\psi) = 1\} \\ &= \{v \in V_k : v(\phi) = 1 \text{ and } v(\psi) = 1\} \\ &= \{v \in V_k : v(\phi \wedge \psi) = 1\} \end{aligned}$$

Thus, the sets  $\mathcal{B}$  form a *basis* for a topology  $\tau$  on the valuation space  $V_k$ : that is, the collection  $\tau$  of open sets of  $V_k$  consists of all possible unions of sets in the collection  $\mathcal{B}$ . By definition, all unions and finite intersections of sets in  $\tau$  are contained in  $\tau$ ; thus, to show  $\tau$  is a valid topology, we need only show that  $\emptyset \in \tau$  and  $V_k \in \tau$ :

Let  $\phi$  be any sentence in  $PL_k$ ; then  $U(\phi \wedge \neg\phi) = \emptyset$  since, for any valuation  $v$ , we cannot have both  $v(\phi) = 1$  and  $v(\neg\phi) = 1$ , and thus we can't have  $v(\phi \wedge \neg\phi) = 1$  for any valuation.

Similarly, for any sentence  $\phi$  and any valuation  $v$ , we must have  $v(\neg(\phi \wedge \neg\phi)) = 1$ , since  $v(\phi \wedge \neg\phi) = 0$ . Thus,  $U(\neg(\phi \wedge \neg\phi)) = V_k$ .

Thus,  $\emptyset \in \tau$  and  $V_k \in \tau$ , and so  $\tau$  is a topology on  $V_k$ . Note that in this construction, we assume the set of logical connectives of  $PL_k$  is truth-functionally complete; that is, this set can express every possible truth table of  $PL_k$ .

Now, let  $\Sigma$  be any set of sentences of  $PL_k$ . Then,

$$\begin{aligned} \Sigma \text{ is unsatisfiable} &\iff \text{there exists no valuation assigning 1 to every } s \in \Sigma \\ &\iff \text{for every valuation } v, v(s) = 0 \text{ for some } s \in \Sigma \\ &\iff \text{for every valuation } v, v(\neg s) = 1 \text{ for some } s \in \Sigma \\ &\iff \text{for every valuation } v, v \in U(\neg s) \text{ for some } s \in \Sigma \\ &\iff \{U(\neg s) : s \in \Sigma\} \text{ is an open cover for } V_k \end{aligned}$$

**Theorem.**  *$PL_k$  is compact if and only if  $V_k$  is compact.*

*Proof.* For clarity, we prove each direction of the statement individually.

$\Leftarrow$ : Suppose  $V_k$  is compact. Then every open cover of  $V_k$  has a finite subcover. Let  $\Sigma$  be a collection of sentences of  $PL_k$ . If  $\Sigma$  is unsatisfiable, then by the above,  $\{U(\neg s) : s \in \Sigma\}$  is an open cover of  $V_k$  and thus has a finite subcover, some finite subcollection of sets  $U(\neg s)$  where  $s \in \Sigma$ . We can define this subcover as  $\{U(\neg s) : s \in \Sigma_0\}$ , where  $\Sigma_0$  is some finite subset of  $\Sigma$ .  $\Sigma_0$  is still by definition an open cover of  $V_k$ , so  $\Sigma_0$  must be unsatisfiable. Thus, a set of sentences  $\Sigma$  of  $PL_k$  is unsatisfiable if and only if some finite subset  $\Sigma_0 \subset \Sigma$  is unsatisfiable, and thus  $PL_k$  is compact.

$\Rightarrow$ : Suppose  $PL_k$  is compact. Let  $\mathcal{C}$  be an open cover of  $V_k$ . Since  $\mathcal{B}$  is a basis, every set  $A \in \mathcal{C}$  can be written as a union of sets of the form  $U(\phi)$ , where  $\phi$  is a sentence in  $PL_k$ . Thus, by rewriting  $A$  as the union of basis sets, we can consider  $\mathcal{C}$  to be a collection of sets of the form  $U(\phi)$ . Let  $s = \neg\phi$ , and let  $\Sigma$  denote the set of sentences  $s$  contributing to the open cover; that is,  $\mathcal{C} = \{U(\neg s) : s \in \Sigma\}$ . Then  $\{U(\neg s) : s \in \Sigma\}$  is an open cover of  $V_k$ , and thus  $\Sigma$  is unsatisfiable. By compactness of  $PL_k$ , there exists some finite subset  $\Sigma_0 \subset \Sigma$  which is unsatisfiable. Thus,  $\mathcal{C}_0 = \{U(\neg s) : s \in \Sigma_0\}$  is an open cover of  $V_k$ ;  $\mathcal{C}_0 \subset \mathcal{C}$ , and  $\mathcal{C}_0$  is finite by the finiteness of  $\Sigma_0$ . Thus, the open cover  $\mathcal{C}$  must have a finite subcover. By definition, this means  $V_k$  is compact.  $\square$

Finally, we prove that  $V_k$  is compact.

**Theorem.**  *$V_k$  is compact*

*Proof.*  $V_k = \{f : K \rightarrow \{0, 1\}\} \cong \{0, 1\}^k$ , where  $K = \{1, 2, \dots, k\}$  denotes the  $k$  sentence letters of  $PL_k$ , and  $\{0, 1\}$  is given the discrete topology (that is, every subset of  $\{0, 1\}$  is considered an open set). Clearly,  $\{0, 1\}$  is compact (as any open cover must already be finite, and thus is its own finite subcover), so by Tychonoff's theorem, which states that any product of compact spaces is itself compact,  $\{0, 1\}^k$  is compact, and thus  $V_k$  is compact.  $\square$

The symbol  $\cong$  denotes that two sets are homeomorphic, meaning that they share important topological properties like compactness.

## 5 Algebraic Logic

### 5.1 Boolean algebras

Propositional logic can also be represented by Boolean Algebras, which are defined as follows: [2]

**Definition 5** (Boolean Algebra). *A Boolean Algebra is a set  $A$  together with binary operations  $+$  and  $\cdot$ , the unary operation  $-$ , and the elements  $0, 1 \in A$  such that the following properties hold:*

- *Commutativity and associativity hold for  $+$  and  $\cdot$ .*
- *Distributivity holds for  $+$  over  $\cdot$  and  $\cdot$  over  $+$ .*
- $x + (x \cdot y) = x$
- $x \cdot (x + y) = x$
- $x + (-x) = 1$
- $x \cdot (-x) = 0$

For any Boolean Algebra, we can define the partial order  $\leq$ , where  $x \leq y$  if and only if  $x + y = y$ .

#### 5.1.1 Boolean Algebra as an Algebra of Sets

We may view a Boolean Algebra as a collection  $A$  of subsets of a larger space  $X$ , with the operations  $+$  corresponding to  $\cup$  (union),  $\cdot$  corresponding to  $\cap$  (intersection), and  $-$  corresponding set complement; here,  $1$  corresponds to the entire space  $X$ , and  $0$  corresponds to the empty set  $\emptyset$ . Thus, the last four properties can be understood as:

- $x \cup (x \cap y) = x$
- $x \cap (x \cup y) = x$
- $x \cup (-x) = X$
- $x \cap (-x) = \emptyset$

Here, the partial order  $\leq$  may be understood as the subset relation  $\subset$ :  $x \leq y$  if and only if  $x \cup y = y$ .

#### 5.1.2 Boolean Algebra and Propositional Logic

We can view the two-element Boolean Algebra as a representation of propositional logic [5]. Here, the two elements are  $1$  and  $0$ —or  $\top$  and  $\perp$ , the propositional constants representing truth and falsity, respectively.  $+$  corresponds to  $\vee$ ,  $\cdot$  to  $\wedge$ , and  $-$  to  $\neg$ , and thus the four final properties are:

$x$	$y$	$x \wedge y$	$x \vee y$	$x \vee (x \wedge y) = x$	$x \wedge (x \vee y) = x$	$x \vee (-x)$	$x \wedge (-x)$
T	T	T	T	T	T	T	F
T	F	F	T	T	T	T	F
F	T	F	T	F	F	T	F
F	F	F	F	F	F	T	F

- $x \vee (x \wedge y) = x$
- $x \wedge (x \vee y) = x$
- $x \vee (-x) = \top$
- $x \wedge (-x) = \perp$

The following truth table represents these properties, for any sentences  $x$  and  $y$  in a propositional language

## 5.2 Algebras as a Model

We can formally describe an algebra as a model of a propositional logic by viewing it in terms of first-order logic [2]. Given a propositional language  $\mathcal{L}$ , we take  $\mathcal{L}$  instead as the first-order language whose terms are exactly the formulas of the propositional language. A logical connective of arity  $n$  is considered a function symbol of arity  $n$ ; 0-ary logical connectives like  $\perp$  are treated as constants. The sentence letters of the propositional language become the variables, and the propositional formulas are considered as terms.

Now, we may introduce algebras as a model of the first-order language. An *algebra*  $\mathcal{A}$  of type  $\mathcal{L}$ , also called an  $\mathcal{L}$ -algebra, is a set  $A$ , called the *carrier* or *universe* of  $\mathcal{A}$ , together with a function  $*^A$  on  $A$ , with the arity of  $*$ , for every connective  $*$  in  $\mathcal{L}$ . If  $*$  is 0-ary,  $*^A$  is simply an element of  $A$ ; if  $A$  is a singleton set,  $\mathcal{A}$  is trivial.

A *valuation* on an  $\mathcal{L}$ -algebra (not to be confused with the valuation function on the formulas of  $\mathcal{L}$  described above) is a map  $v$  from the set of variables of  $\mathcal{L}$  into  $A$ ; the valuation of a formula of  $\mathcal{L}$  is described as follows:

- $v(p) = p$  if  $p$  is a variable
- $v(*) = *^A$  if  $t$  is a 0-ary connective
- $v(*(\phi_1, \phi_2, \dots, \phi_n)) = *^A(v(\phi_1), v(\phi_2), \dots, v(\phi_n))$  if  $*$  is an  $n$ -ary connective,  $n > 0$ , and  $\phi_1, \phi_2, \dots, \phi_n$  are formulas.

We let  $\mathbf{FM}_{\mathcal{L}}$  denote the *algebra of formulas* of  $\mathcal{L}$ , which has the set of all  $\mathcal{L}$ -formulas as a carrier. For every  $n$ -ary connective  $*$  (where  $n > 0$ ), the function  $*\mathbf{FM}_{\mathcal{L}}$  maps the tuple of formulas  $(\phi_1, \phi_2, \dots, \phi_n)$  to the formula  $*(\phi_1, \phi_2, \dots, \phi_n)$ , and, if  $*$  is a 0-ary connective,  $*\mathbf{FM}_{\mathcal{L}} = *$ .

It is clear that this algebra is a model of the first-order language  $\mathcal{L}$ .

### 5.3 Algebraic Semantics

Representing first-order logic in algebraic terms.

### 5.4 The Lindenbaum-Tarski Algebra

In order to more thoroughly show the relationship between Boolean Algebras and a propositional language  $\mathcal{L}$ , we introduce an equivalence relation on the set of formulas of  $\mathcal{L}$  [5]. We say two formulas  $\alpha$  and  $\beta$  are *logically equivalent* if  $\alpha \models \beta$  and  $\beta \models \alpha$ .

Note that  $\alpha \models \beta \iff$  for every valuation function  $v$ ,  $v(\alpha) = 1$  implies  $v(\beta) = 1 \iff v(\alpha \supset \beta) = 1$ , which follows immediately from the definition of  $\models$ . Thus, the logical equivalence of  $\alpha$  and  $\beta$  is exactly equivalent to the statement  $\models (\alpha \supset \beta) \wedge (\beta \supset \alpha)$ , which we abbreviate as  $\models \alpha \Leftrightarrow \beta$ . This is exactly the statement  $\alpha$  if and only if  $\beta$ , which is the usual colloquial definition of logical equivalence.

The relation of logical equivalence, denoted  $\equiv$ , is an equivalence relation. It is symmetric by definition, and easily seen to be reflexive ( $\alpha \models \alpha$ , thus  $\alpha \equiv \alpha$ ). If  $\alpha \equiv \beta$  and  $\beta \equiv \gamma$ , then note that for any valuation  $v$ ,  $v(\alpha) = 1 \iff v(\beta) = 1 \iff v(\gamma) = 1$ , and thus  $\alpha \equiv \gamma$ . Thus, we can decompose the set of all formulas into its equivalence classes under  $\equiv$ , and denote the equivalence class of  $\alpha$  as  $[\alpha]$ .

In this manner, we can define the *Lindenbaum-Tarski Algebra* associated with a propositional logic:

**Definition 6.** *The Lindenbaum-Tarski algebra for a set  $X$  of propositional formulas is the Boolean algebra formed from the set  $X'$  of equivalence classes of formulas, written  $\langle X', +, \cdot, -, 1, 0 \rangle$ .*

We define  $+$ ,  $\cdot$ ,  $-$ ,  $1$ , and  $0$  as follows:

- $[\alpha] + [\beta] = [\alpha \vee \beta]$
- $[\alpha] \cdot [\beta] = [\alpha \wedge \beta]$
- $-[\alpha] = [-\alpha]$
- $0 = [\perp]$
- $1 = [\top]$

**Theorem.** *The Lindenbaum-Tarski algebra is, in fact, a Boolean Algebra*

*Proof.* Commutativity, associativity, and distributivity all apply to  $\wedge$  and  $\vee$  as we expect, so they must hold for the Lindenbaum-Tarski algebra. Thus, it remains to prove the other properties of Boolean Algebras.

Note first that for any formulae  $\phi$  and  $\psi$ ,  $\psi \equiv \phi$  if and only if, for any valuation function  $v$ ,  $v(\phi) = 1 \Leftrightarrow v(\psi) = 1$ ; in other words,  $\phi \equiv \psi$  if and only if  $v(\phi) = v(\psi)$  for every valuation function  $v$ .

- $[\alpha] + ([\alpha] \cdot [\beta]) = [\alpha]$  becomes  $[\alpha \vee (\alpha \wedge \beta)]$ . If  $v(\alpha) = 1$ , then  $v(\alpha \vee (\alpha \wedge \beta)) = 1$ . If  $v(\alpha \vee (\alpha \wedge \beta)) = 1$ , then  $v(\alpha) = 1$ . So  $\alpha \models \alpha \vee (\alpha \wedge \beta)$  and  $\alpha \vee (\alpha \wedge \beta) \models \alpha$ , so  $\alpha \equiv (\alpha \vee (\alpha \wedge \beta))$ , meaning they belong to the same equivalence class. Thus,  $[\alpha \vee (\alpha \wedge \beta)]$ .
- $([\alpha] \cdot ([\alpha] + [\beta])) = [\alpha]$  becomes  $[\alpha \wedge (\alpha \vee \beta)] = [\alpha]$ . If  $v(\alpha) = 1$ , then  $v(\alpha \wedge (\alpha \vee \beta)) = 1$ , and thus  $v(\alpha \vee \beta) = 1$ . If  $v(\alpha \wedge (\alpha \vee \beta)) = 1$ , then  $v(\alpha) = 1$ . Thus,  $(\alpha \wedge (\alpha \vee \beta)) \equiv \alpha$ , so  $[\alpha \wedge (\alpha \vee \beta)] = [\alpha]$
- $[\alpha] + [-\alpha] = 1$  becomes  $[\alpha \vee \neg\alpha] = [\top]$ .  $v(\alpha) = 1$  or  $0$ , in which case  $v(\neg\alpha) = 1$ , so  $v(\alpha \vee \neg\alpha) = 1 = v(\top)$ , so  $[\alpha \vee \neg\alpha] = [\top]$
- $[\alpha] \cdot [-\alpha] = 0$  becomes  $[\alpha \wedge \neg\alpha] = [\perp]$ . We cannot have both  $v(\alpha) = 1$  and  $v(\neg\alpha) = 0$ , so  $v(\alpha \wedge \neg\alpha) = 0 = v(\perp)$ . Thus,  $[\alpha \wedge \neg\alpha] = [\perp]$

□

## 5.5 Cylindric Algebras

Just as we represented propositional logic via Boolean Algebras, we can also represent first-order logic via *Cylindric Algebras*, defined as follows [4]:

**Definition 7.** A *cylindric algebra of dimension  $n$*  is a structure

$$\mathcal{A} = \langle A, +, \cdot, -, 0, 1, c_i, d_{ij} \rangle$$

, where  $\langle A, +, \cdot, -, 0, 1 \rangle$  is a Boolean Algebra,  $c_i$  is a unary operation on  $A$  for each  $i < n$ ,  $d_{ij} \in A$  for all  $i, j < n$ , and the following properties hold for all  $i, j, k < n$  and all  $x, y \in A$ :

1.  $c_i(0) = 0$
2.  $x \leq c_i(x)$
3.  $c_i(x \cdot c_k(y)) = c_i(x) \cdot c_i(y)$
4.  $c_i(c_j(x)) = c_j(c_i(x))$
5.  $d_{ii} = 1$
6. If  $i \neq j$  and  $i \neq k$ , then  $d_{jk} = c_i(d_{ji} \cdot d_{ik})$
7. If  $i \neq j$ , then  $c_i(d_{ij} \cdot x) \cdot c_i(d_{ij} \cdot \neg x) = 0$ .

For a first-order language  $\mathcal{L}$

## References

- [1] Curtis Franks. “Propositional Logic”. In: *Stanford Encyclopedia of Philosophy* Winter 2024 Edition (2023). URL: <https://plato.stanford.edu/archives/win2024/entries/logic-propositional/>.
- [2] Ramon Jansana. “Algebraic Propositional Logic”. In: *The Stanford Encyclopedia of Philosophy* (2022).
- [3] Bert Mendelson. *Introduction to Topology*. Dover Publications, Inc, 1962.
- [4] J. Donald Monk. *Mathematical Logic*. Springer-Verlag, 1976.
- [5] J. Donald Monk. “The Mathematics of Boolean Algebra”. In: *The Stanford Encyclopedia of Philosophy* (2022).
- [6] A. C. Paseau and Robert Leek. “The Compactness Theorem”. In: *Internet Encyclopedia of Philosophy* ().
- [7] Stewart Shapiro and Teresa Kouri Kissel. “<https://plato.stanford.edu/archives/fall2025/entries/logic-classical/>”. In: *Stanford Encyclopedia of Philosophy* (2025).
- [8] R. M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.