

Logic (WIP)

Nathan McKeown-Luckly (Discord Username: skeletman)

Autumn 2023

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 2 |
| 2 | An Informal Discussion on Propositional Logic | 2 |
| 2.1 | What Propositional Logic can do | 2 |
| 2.2 | Logical Connectives | 3 |
| 2.3 | Truth Values and Truth Tables | 3 |
| 2.4 | Tautologies | 5 |
| 2.5 | Semantic Entailment | 7 |
| 2.6 | Syntactic Entailment | 8 |
| 2.7 | Theories and Theorems | 9 |
| 2.8 | Metalogic | 9 |
| 3 | A Formal Formulation of Propositional Logic | 10 |
| 3.1 | The Language of Propositional Logic | 10 |
| 3.2 | Semantic Entailment | 11 |
| 3.3 | Syntactic Entailment | 12 |
| 4 | Metalogical Problems in Propositional Logic | 13 |
| 4.1 | The Completeness Problem | 13 |
| 4.2 | The Boolean Satisfiability Problem (SAT) | 14 |
| | Index | 15 |

1 Introduction

In this document, we will discuss logic as a foundation of formal reasoning. In doing so, we will discover what it means for a statement to be true (Semantic Entailment), and what it means for a statement to be proven (Syntactic Entailment) in various logical frameworks.

Logics are usually made up of a language of valid formulae and sentences, a set of rules for proving things, and a definition of structures in which the truth of a sentence is defined.

2 An Informal Discussion on Propositional Logic

2.1 What Propositional Logic can do

In *propositional logic*, we can talk about simple logical statements called *propositional sentences*. These consist of simple statements called *primitive propositions*, along with *logical connectives* which allow us to make more complicated expressions.

Example. In propositional logic, we can talk and reason about sentences like "If Bob is a zebra, then Bob has stripes". We call the if-then relationship *implication*, and say "Bob is a zebra implies Bob has stripes". Implication is an example of a logical connective, and "Bob is a zebra" and "Bob has stripes" are examples of primitive propositions.

Example. We could also have a sentence "My coat is red and my hat is blue". This is an example of *logical and*, another logical connective.

We abstract the actual statements by replacing them with symbols.

Example. We use \implies to mean "implies", and we might abstract away the primitive propositions like "Bob is a zebra" to a single symbol p and "Bob has stripes" to q . We then could write $p \implies q$ instead of "If Bob is a zebra then Bob has stripes".

Example. We use the \wedge symbol for logical and. If "My coat is red" is p and "My hat is blue" is q , then "My coat is red and my hat is blue" would be written $p \wedge q$.

2.2 Logical Connectives

Logical connectives are used to turn propositions into more complicated propositions. In figure 2.1 we see some important examples.

Figure 2.1: Logical connectives in propositional logic

| Connective | Meaning |
|------------|----------------|
| \implies | implies |
| \neg | not |
| \vee | or |
| \wedge | and |
| \iff | if and only if |

2.3 Truth Values and Truth Tables

We can also assign a true-or-false *truth value* to each sentence. This assignment can be treated as a function from the set of sentences to $\{0, 1\}$, where 0 represents false and 1 represents true. Each connective will have an associated rule for how truth values must behave. If these rules are satisfied, then the assignment is called a *valuation*. We can visualise these rules with a *truth table*.

Figure 2.2: Truth tables for logical connectives

| | | | | | | | |
|-----|-----|----------------|-----|----------|-----|-----|------------|
| p | q | $p \implies q$ | p | $\neg p$ | p | q | $p \vee q$ |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

| | | | | | |
|-----|-----|--------------|-----|-----|------------|
| p | q | $p \wedge q$ | p | q | $p \iff q$ |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

We interpret the truth tables in figure 2.2 as rules that an assignment must follow in order to be called a valuation. For instance, if v is a valuation, $v(p) = 1$, $v(q) = 1$, and we want to know $v(p \wedge q)$, then we read the corresponding line of the truth table for logical and to see that $v(p \wedge q) = 1$. Now that we

have defined rules of how truth valuations behave, we can use truth tables to display all possible valuations for a complex sentence, each row represents one valuation.

The valuation of a sentence depends only on the valuation of the primitive propositions contained in it, so we start by making a column for each sub-sentence, and a row for each possible truth value of the primitive propositions, filling out the respective cells with the truth values. Then, in each row, we use the rules in figure 2.2 to calculate the truth values of the complex sentences.

Figure 2.3: The truth table for $(p \wedge q) \implies r$

| p | q | r | $p \wedge q$ | $(p \wedge q) \implies r$ |
|-----|-----|-----|--------------|---------------------------|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Figure 2.4: The truth table for $\neg(p \implies \neg q)$

| p | q | $\neg q$ | $p \implies \neg q$ | $\neg(p \implies \neg q)$ |
|-----|-----|----------|---------------------|---------------------------|
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |

Notice that the truth table for $\neg(p \implies \neg q)$ is the same as for $p \wedge q$.

Figure 2.5: The truth table for $\neg p \implies q$

| p | q | $\neg p$ | $\neg p \implies q$ |
|-----|-----|----------|---------------------|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |

Notice that the truth table for $\neg p \implies q$ is the same as for $p \vee q$.

Sidenote. In figures 2.4 and 2.5, we see that both logical and and logical or are equivalent, up to having the same truth values, to expressions involving only implies and logical not. This is very useful when analysing the mathematical properties of the framework of propositional logic itself, as we can simplify our logic for the purpose of simplifying proofs about it. The study of the properties of logic is called *metalogic* (see section 2.8).

In practice, instead of simplifying to a logic involving only implies and logical not, we simplify even further to a logic which has implies and a primitive proposition "false", written \perp , for which a valuation must assign truth value 0.

In this logic, we encode logical not as follows

Figure 2.6: The truth table for $p \implies \perp$

| p | \perp | $p \implies \perp$ |
|-----|---------|--------------------|
| 0 | 0 | 1 |
| 1 | 0 | 0 |

Notice that the truth table for $p \implies \perp$ is the same as for $\neg p$.

2.4 Tautologies

Definition 2.1 (Tautology). A *tautology* is a sentence that always holds true. In propositional logic, this means that a sentence s is a tautology if for every valuation v , $v(s) = 1$.

In practice, a sentence is a tautology if it has a 1 in every row of its column in a truth table.

Figure 2.7: The truth table for *Pierce's Law*

Pierce's Law: $((p \implies q) \implies p) \implies p$

| p | q | $p \implies q$ | $(p \implies q) \implies p$ | $((p \implies q) \implies p) \implies p$ |
|-----|-----|----------------|-----------------------------|--|
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Figure 2.8: The truth table for *Axiom K*

Axiom K: $p \implies (q \implies p)$

| p | q | $q \implies p$ | $p \implies (q \implies p)$ |
|-----|-----|----------------|-----------------------------|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

Figure 2.9: The truth table for *Axiom S*

Axiom S: $[p \Rightarrow (q \Rightarrow r)] \Rightarrow [(p \Rightarrow q) \Rightarrow (p \Rightarrow r)]$

| p | q | r | $q \Rightarrow r$ | $p \Rightarrow (q \Rightarrow r)$ | $p \Rightarrow q$ | $p \Rightarrow r$ |
|-----|-----|-----|-------------------|-----------------------------------|-------------------|-------------------|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| $(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$ | $[p \Rightarrow (q \Rightarrow r)] \Rightarrow [(p \Rightarrow q) \Rightarrow (p \Rightarrow r)]$ |
|---|---|
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 0 | 1 |
| 1 | 1 |

Sidenote. The "K" in "axiom K" stands for "konstante" (German for constant), and the "S" in "axiom S" stands for "substitution". This is because in certain logical semantics, propositions are interpreted as types, and implication is interpreted as functions between types. In these semantics, axiom K corresponds to a constant function, and axiom S corresponds to a substitution of variables into a function to obtain a new function.

You can tell these are important axioms to logicians because they have names.

2.5 Semantic Entailment

Where tautologies are sentences which are always true, *semantic entailment* is when a sentence is necessarily true **given that** some set of sentences are true.

Definition 2.2 (Model). Let S be a set of sentences and v a valuation. v is a *model* of S if:

$$\forall p \in S, v(p) = 1$$

i.e. for all sentences p in S , its valuation $v(p)$ is 1.

Definition 2.3 (Semantic Entailment). We say that S *semantically entails* p , or just S *entails* p , if every model of S is a model of p . That is, every valuation where everything in S is true, also has p is true.

We say S are the *premises*, and p is a *semantic consequence*

For semantic entailment, we use the symbol \models . Given a set of sentences S , and a sentence p

$$S \models p$$

means " S semantically entails p ". In practice, $S \models p$ means that every row of a truth table of everything in S and p where everything in S has value 1, will have value 1 for p . If S is empty, this means p is a tautology, and we write $\models p$.

Examples. In these examples, I will color in green the rows of the table which all premises have value 1. Then only the green rows of the consequences are required to have value 1 for semantic entailment.

Figure 2.10: A truth table showing $p \iff q \models p \implies q$

| p | q | $p \iff q$ (premise) | $p \implies q$ (consequence) |
|-----|-----|----------------------|------------------------------|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Figure 2.11: A truth table showing $p, p \implies q \models q$

| p (premise) | q (consequence) | $p \implies q$ (premise) |
|---------------|-------------------|--------------------------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

2.6 Syntactic Entailment

Propositional logic also has a notion of proof, called *syntactic entailment*. Propositional logic comes equipped with a set of *axioms*, sentences which we assume to be always true for the purposes of proving things. Also, there are *deduction rules*, rules by which simple deductions can be made.

Example. For every sentence p, q and r , *axiom K* and *axiom S* are examples of logical axioms.

Axiom K: $p \Rightarrow (q \Rightarrow p)$

Axiom S: $[p \Rightarrow (q \Rightarrow r)] \Rightarrow [(p \Rightarrow q) \Rightarrow (p \Rightarrow r)]$

Example. An example of a deduction rule¹ is *modus ponens*, which states for every sentence p and q

From p and $p \Rightarrow q$ we can deduce q .

Definition 2.4 (Proof). In propositional logic, given a set of premises S and sentence p , a *proof* of p from S is a finite list of sentences (t_1, \dots, t_n) such that each t_i is

- An axiom
- A premise of S
- Deducible from previous sentences via a deduction rule.

And t_n is p . We call the t_i the lines of the proof, and usually display them vertically.

If there is a proof of p from S , we write $S \vdash p$ and say " S proves p " or " S syntactically entails p ".

Example. A proof of $p \Rightarrow r$ from premises $p \Rightarrow q$ and $q \Rightarrow r$

- | | |
|--|---------------------------|
| 1. $(q \Rightarrow r) \Rightarrow (p \Rightarrow (q \Rightarrow r))$ | (Axiom K) |
| 2. $q \Rightarrow r$ | (Premise) |
| 3. $p \Rightarrow (q \Rightarrow r)$ | (Modus Ponens on 1 and 2) |
| 4. $[p \Rightarrow (q \Rightarrow r)] \Rightarrow [(p \Rightarrow q) \Rightarrow (p \Rightarrow r)]$ | (Axiom S) |
| 5. $(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$ | (Modus Ponens on 3 and 4) |
| 6. $p \Rightarrow q$ | (Premise) |
| 7. $p \Rightarrow r$ | (Modus Ponens on 5 and 6) |

¹In some formulations of propositional logic the only example

2.7 Theories and Theorems

Definition 2.5 (Theory). A *theory* is a set of sentences.

We interpret theories as a set of premises, of which we study the logical consequences.

Definition 2.6 (Theorem). For a theory T , a *theorem* of T is a sentence p such that $T \vdash p$.

Definition 2.7 (Consistent Theory). A theory T is *consistent* if $T \not\vdash \perp$. Intuitively, a theory is consistent if we cannot prove any falsehoods.

Definition 2.8 (Satisfiable Theory). A theory T is *satisfiable* if it has a model, i.e. $T \not\vdash \perp$.

Definition 2.9 (Deductively Closed Theory). A theory T is *deductively closed* if for every sentence p , if $T \vdash p$ then $p \in T$.

Definition 2.10 (Complete Theory). A theory T is *complete* if it is consistent and for every sentence p , either $T \vdash p$ or $T \vdash \neg p$.

2.8 Metalogic

Metalogic refers to the study of the properties of a logical system.

" $p \implies q, q \implies r \vdash p \implies r$ " is an example of a *logical statement*. It is a statement made within the framework of our logic.

"For all theories T , and all sentences p , $T \vdash p$ if and only if $T \models p$ " is a *metalogical statement*. It is a statement about our logical framework.

3 A Formal Formulation of Propositional Logic

3.1 The Language of Propositional Logic

In propositional logic, we assign some special symbols and read them as in figure 3.1. For the meantime, we do not assign a meaning to these symbols.

Figure 3.1: Special symbols for propositional logic

| Symbol | Read as |
|------------|---------------|
| \implies | Implies |
| \perp | False |
| $($ | Left Bracket |
| $)$ | Right Bracket |

Definition 3.1 (Propositional Language). Let S be a set of symbols that contains no special symbols. S is known as a set of *primitive propositions* or *atomic sentences*. Usually we will use lowercase latin symbols to represent primitive propositions, such as p, q, r , but we leave the definition open as some applications may need more symbols than we have letters, maybe even infinitely many!

The *propositional language* $\mathcal{L}(S)$ consists of the set of *sentences* defined by the following construction

- If p is a primitive proposition (i.e. $p \in S$), then it is a sentence
- \perp is a sentence
- If α and β are sentences, then so is $(\alpha \implies \beta)$

Importantly, every valid sentence has a unique construction in this way.

Sidenote. Recall in figures 2.4, 2.5 and 2.6 we saw equivalent statements to $p \wedge q$, $p \vee q$ and $\neg p$. In our formulation of propositional logic, we take these as *definitions* (see fig. 3.2).

Figure 3.2: Logical shorthands

| Shorthand | Definition | Meaning |
|--------------|--|------------------------|
| $\neg p$ | $p \implies \perp$ | not p |
| $p \vee q$ | $\neg p \implies q$ | p or q |
| $p \wedge q$ | $\neg(p \implies \neg q)$ | p and q |
| $p \iff q$ | $(p \implies q) \wedge (q \implies p)$ | p if and only if q |

3.2 Semantic Entailment

Definition 3.2 (Valuation). A *valuation* is a function $v : \mathcal{L}(S) \rightarrow \{0, 1\}$ such that.

- $v(\perp) = 0$
- For all $p, q \in \mathcal{L}(S)$ we have

$$v(p \implies q) = 1 \text{ if and only if } [v(p) = 0 \text{ or } v(q) = 1]$$

i.e. v assigns each sentence a truth value, \perp is assigned false, and the truth values respect the law of implication.

Definition 3.3 (Theory). A *theory* is a set of sentences, i.e. $T \subseteq \mathcal{L}(S)$.

Definition 3.4 (Model). Let T be a theory. A valuation v is a *model* of T if

$$\text{for all } p \in T, v(p) = 1$$

i.e. everything in T is assigned true by v .

Definition 3.5 (Semantic Entailment). Let T be a theory and p a sentence. We say that T *semantically entails* p , or just S *entails* p , written $T \models p$, if every model of T is a model of p .

Definition 3.6 (Logical Equivalence). Let p and q be sentences. p and q are *logically equivalent*, written $p \equiv q$, if $p \models q$ and $q \models p$.

Let T and T' be theories. T and T' are logically equivalent if $T \models q$ for all $q \in T'$ and $T' \models p$ for all $p \in T$.

Warning. A theory T is **not necessarily** equivalent to some single sentence if T is infinite.

For example, if we have primitive propositions $S = \{p_i : i \in \mathbb{N}\}$, and our theory $T = S$ is all of the primitive propositions. Then if q is a sentence, it is finite, so may only mention finitely many of the p_i . Let M be the maximum i for which p_i appears in q .

Suppose $T \models q$. T has a model v , just assign all p_i to true. As $T \models q$, v is a model of q . Define a new valuation v' by

$$v'(p_i) = \begin{cases} 1 & \text{if } i \neq M+1 \\ 0 & \text{if } i = M+1 \end{cases}$$

v' must be a model of q still, as p_{M+1} does not appear in q . But v' is not a model of T . Hence $q \not\models T$.

3.3 Syntactic Entailment

Propositional logic also has a notion of proof, called *syntactic entailment*. Propositional logic comes equipped with a set of *axioms*, sentences which we assume to be always true for the purposes of proving things. Also, there are *deduction rules*, rules by which simple deductions can be made.

Definition 3.7 (Axioms of Propositional Logic). The *axioms of propositional logic* are

- **Axiom K:** For all sentences p and q

$$p \implies (q \implies p)$$

- **Axiom S:** For all sentences p , q and r

$$[p \implies (q \implies r)] \implies [(p \implies q) \implies (p \implies r)]$$

- **The Law of Excluded Middle (LEM):** For all sentences p

$$\neg\neg p \implies p$$

Definition 3.8 (Deduction Rule of Propositional Logic). The only *deduction rule* of this formulation of propositional logic is *modus ponens*, which states for every sentence p and q

From p and $p \implies q$ we can deduce q .

Definition 3.9 (Proof). In propositional logic, given a theory T and sentence p , a *proof* of p from T is a finite list of sentences (t_1, \dots, t_n) such that each t_i is either

- An axiom
- In T
- Deducible from previous sentences via modus ponens.

And t_n is p . We call the t_i the lines of the proof, and usually display them vertically.

If there is a proof of p from T , we write $S \vdash p$ and say " T proves p " or " T syntactically entails p ".

Sidenote. In practice when doing real proofs, we will add more deduction rules in order to simplify the proof. However, just modus ponens is strong enough to prove that all these other rules are sound, and is convenient to prove metalogical statements about thanks to its simplicity.

4 Metalogical Problems in Propositional Logic

4.1 The Completeness Problem

Definition 4.1 (Soundness Problem). Given that $T \vdash p$, can we show that $T \models p$? I.e. if we have a proof of p from T , can we then show that every model of T is also a model of p ?

Definition 4.2 (Adequacy Problem). Given that $T \models p$, can we show that $T \vdash p$? I.e. if we know that every model of T is a model of p , can we then determine there is a proof of p from T ?

Definition 4.3 (Completeness Problem). Do both the soundness and adequacy problems hold true? I.e. is it true that $T \vdash p$ if and only if $T \models p$?

The soundness problem is intuitively fairly obvious: if we could prove something that isn't true, then our deductive system would be flawed.

The adequacy problem, however, is much more difficult to reason about intuitively. How do we know that our proof system is strong enough to prove every consequence of a theory?

It turns out that the completeness problem does hold true (under standard mathematical assumptions). Some consequences of completeness are

Theorem 4.4 (Decidability Theorem). Let T be a finite theory, and p a sentence. There is an algorithm which determines in finite time whether $T \vdash p$ or not.

Proof. Completeness says $T \vdash p$ if and only if $T \models p$. An algorithm can check in finite time if $T \models p$ or not by constructing a truth table, then checking that every row where all of T holds, p holds. \square

Theorem 4.5 (Compactness Theorem). Let T be a (possibly infinite) theory. If every finite subset of T has a model, then T has a model. I.e.

$$\text{If every finite } S \subset T \text{ has } S \not\models \perp, \text{ then } T \not\models \perp$$

or equivalently

$$T \models \perp \text{ implies that there is finite } S \subseteq T \text{ that has } S \models \perp$$

Proof. By completeness, we can replace \models with \vdash to see that we need only show

$$T \vdash \perp \text{ implies that there is finite } S \subseteq T \text{ that has } S \vdash \perp$$

But if there is a proof P of \perp from T , this proof uses only finitely many premises from T , as proofs are finite. Let S be the set of premises of T appearing in P . Then S is finite and $S \vdash \perp$. \square

4.2 The Boolean Satisfiability Problem (SAT)

Index

- adequacy problem, [13](#)
- atomic sentences, [10](#)
- Axiom K, [6](#)
- axiom K, [8](#)
- Axiom S, [6](#)
- axiom S, [8](#)
- axioms, [8](#), [12](#)
- axioms of propositional logic, [12](#)

- compactness theorem, [13](#)
- complete, [9](#)
- completeness problem, [13](#)
- consistent, [9](#)

- decidability theorem, [13](#)
- deduction rule, [12](#)
- deduction rules, [8](#), [12](#)
- deductively closed, [9](#)

- entails, [7](#), [11](#)

- implication, [2](#)

- logical and, [2](#)
- logical connectives, [2](#)
- logical statement, [9](#)
- logically equivalent, [11](#)

- Metalogic, [9](#)

- metalogic, [5](#)
- metalogical statement, [9](#)
- model, [7](#), [11](#)
- modus ponens, [8](#), [12](#)

- Pierce's Law, [5](#)
- premises, [7](#)
- primitive propositions, [2](#), [10](#)
- proof, [8](#), [12](#)
- propositional language, [10](#)
- propositional logic, [2](#)
- propositional sentences, [2](#)

- satisfiable, [9](#)
- semantic consequence, [7](#)
- semantic entailment, [7](#)
- semantically entails, [7](#), [11](#)
- sentences, [10](#)
- Soundness Problem, [13](#)
- syntactic entailment, [8](#), [12](#)

- tautology, [5](#)
- theorem, [9](#)
- theory, [9](#), [11](#)
- truth table, [3](#)
- truth value, [3](#)

- valuation, [3](#), [11](#)