

Logic for Artificial Intelligence and Machine Learning (WIP)

Nathan McKeown-Luckly (Discord Username: skeletman)

Autumn 2023

Contents

1	Introduction	2
2	Propositional Logic	2
2.1	An Informal Discussion on Propositional Logic	2
2.1.1	What Propositional Logic can do	2
2.1.2	Logical Connectives	3
2.1.3	Truth Values	3
2.1.4	Logic vs Metalogic	7
2.2	The Language of Propositional Logic	8
	Index	9

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 Propositional Logic

2.1 An Informal Discussion on Propositional Logic

2.1.1 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 $p \wedge q$.

2.1.2 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

2.1.3 Truth Values

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	$p \wedge q$
0	0	0	0	0	0
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 now also use truth tables to display the truth values of more complicated sentences given fixed values of the primitive propositions in that expression.

Figure 2.3: The truth table for *Pierce's Law*

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 2.4: The truth table for *Axiom K*

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

Figure 2.5: The truth table for *Axiom S*

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

We also have a special proposition called False, written \perp . We can encode many common logical operations in terms of implication and falsehood.

Example (Logical not). In English, the saying "then pigs can fly" after a statement means that statement is ridiculous, so "If I am rich then pigs can fly" means "I am not rich". We could talk about the same thing in propositional logic by saying "I am rich implies false" to mean "I am not rich". If we were to call "I am rich" r , then we would write

$$r \implies \perp$$

to mean "I am not rich". We write as shorthand for this $\neg r$.

Example (Logical or). Now we have defined logical not, we can define more complicated things such as "p or q" as

$$\neg p \implies q$$

We write $p \vee q$ as shorthand for this.

Propositional logic, however, cannot quantify over objects.

Example. In propositional logic, we could not say "All zebras have stripes". To do this we would need to add *quantifiers* to our logic. Propositional logic with quantifiers is called *first order logic*, and is much more complicated.

In figure 2.6 we see the meaning of various logical shorthands

Figure 2.6: Various propositional logic shorthand

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

We can also make deductions via proofs, and we use the \vdash symbol to mean "the right hand side can be proven by the left hand side". We read the symbol \vdash as "proves" or "syntactically entails".

Example (Deductions). From p and $p \implies q$, we can deduce q via a rule called *modus ponens*. This constitutes a proof of q . We would write

$$p, p \implies q \vdash q$$

There is also a notion of truth of statements in propositional logic. We use the \models symbol to mean "whenever the left hand side is true, the right hand side is true" and read it as "semantically entails" or just "entails".

Example (Semantic Entailment). Whenever "If Bob is a zebra, then Bob has stripes" is true, we see that "If Bob has no stripes, then Bob is not a zebra" is also true. In symbols this would be written

$$p \implies q \models \neg q \implies \neg p$$

2.1.4 Logic vs Metalogic

2.2 The Language of Propositional Logic

In propositional logic, we assign some special symbols and read them as in figure 2.7. For the meantime, we do not assign a meaning to these symbols.

Figure 2.7: Special Symbols for Propositional Logic

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

Definition 2.1 (Propositional Language). Let S be a set of symbols that contains 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!

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

- If p is a primitive proposition, 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.

Note that we do not assign a meaning to sentences yet, these are simply strings of symbols at the current time.

Index

atomic sentences, [8](#)

Axiom K, [4](#)

Axiom S, [5](#)

first order logic, [6](#)

implication, [2](#)

logical and, [2](#)

logical connectives, [2](#)

modus ponens, [6](#)

Pierce's Law, [4](#)

primitive propositions, [2](#), [8](#)

propositional language, [8](#)

propositional logic, [2](#)

propositional sentences, [2](#)

quantifiers, [6](#)

sentences, [8](#)

truth table, [3](#)

truth value, [3](#)

valuation, [3](#)