Logic (WIP)

For Artificial Intelligence and Machine Learning

Nathan McKeown-Luckly (Discord Username: skeletman)

Autumn 2023

Contents

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
$\Longrightarrow$	implies
$\neg$	not
$\vee$	or
$\wedge$	and
$\Longleftrightarrow$	if and only if

2.1.3 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	$p \Longrightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

p	$\neg p$
0	1
1	0

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$p \Longleftrightarrow q$
0	0	1
0	1	0
1	0	0
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 subsentence, 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 2.6, 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 2.1.6).
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 $\bot$, 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 $p \implies \bot$

p	$\bot$	$p \implies \bot$
0	0	1
1	0	0

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

2.1.4 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	$q \Rightarrow r$	$p \Rightarrow (q \Rightarrow r)$	$p \Rightarrow q$	$p \Rightarrow r$	$(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$	$[p \Rightarrow (q \Rightarrow r)] \Rightarrow [(p \Rightarrow q) \Rightarrow (p \Rightarrow r)]$
0	0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1
0	1	1	1	1	1	1	1	1
1	0	0	1	1	0	0	1	1
1	0	1	1	1	0	1	1	1
1	1	0	0	0	1	0	0	1
1	1	1	1	1	1	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.

2.1.5 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 $\vDash$. Given a set of sentences S, and a sentence p

$$S \vDash p$$

means "S semantically entails p". In practice, $S \vDash p$ means that every row of a truth table ofeverything in S and p where everything in S has value 1, will have value 1 for 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 $p \iff q \vDash p \implies q$

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 $p,\ p \implies q \vDash q$

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

2.1.6 Metalogic

Figure 2.12: Various propositional logic shorthand

Shorthand	Meaning	Definition
$\neg p$	not p	$p \implies \bot$
$p \vee q$	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

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

2.2 The Language of Propositional Logic

In propositional logic, we assign some special symbols and read them as in figure 2.13. For the meantime, we do not assign a meaning to these symbols.

Figure 2.13: Special Symbols for Propositional Logic

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

Definition 2.4 (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
- $\bot$ 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