Propositional Logic 1

Definition of a proposition

A **proposition** is a sentence which has a **truth value**. That is, it must either be `true` or `false`, often shortened to T or F. Examples of propositions are:

- "Sheffield is a city in England";
- "The sum of every pair of odd numbers is even";
- "All people have green hair";
- "$x = x + 1$";
- "This statement is the second statement in this list";
- "$3 = 3$".

Some sentences require a bit of thought in order to realise that they are propositions. For example, the statement "It rained last week" is a proposition. It definitely did *or* did not rain next week.

A proposition whose truth value depends on a particular circumstance is called **hypothetical**, while a proposition which does not depend on any particular circumstances is called **categorical**.

Propositions which are always true are called **tautologies**, such as the last statement in the list above. Conversely, propositions which are always false are called **contradictions**, such as the fourth statement in the list above.

Examples of sentences which are not propositions are:

- "Come here!";
- "What day is it?";
- "This sentence is false".

The first of these is a command; the second is a question; the third, though it might look like a proposition, is neither true nor false, since either would give rise to a contradiction.

It can get messy if we keep writing the full statement of each proposition. Often we will abbreviate propositions using letters, such as p, q, or r. For example, we might say that p stands for the proposition "18 is an even number". Then we can say that p has truth value T, or even $p = T$.

Video Visit the URL below to view a video:
`https://www.youtube.com/embed/K_rexhiMux0`

Connectives

Some propositions are really combinations of other, simpler, propositions. For example, the statement “18 is divisible by 2 and 3” is a proposition which is `true`. Its truth value is dependent on the truth value of the two statements “18 is divisible by 2” and “18 is divisible by 3”. We will look at the different types of **connectives** that are used in logic to build complex propositions out of simpler ones. The simplest of these are called NOT, AND, and OR.

NOT

Suppose that

$$\begin{aligned} p &= \text{“It is cold”}, \\ n &= \text{“It isn't cold”}. \end{aligned}$$

The truth value of n can be worked out from the truth value of p.

p	n
`false`	`true`
`true`	`false`

We write the **negation** of p as $\neg p$ and read it as “not p”. This is true when p is false and false when p is true.

The **truth table** is as follows. This table shows how the truth value of p affects the truth value of $\neg p$.

p	$\neg p$
F	T
T	F

There are two rows because there is one statement, p, and two truth values, T or F, giving $2^1 = 2$ possibilities.

AND

Suppose that we have the propositions

$$\begin{aligned} p &= \text{“It is cold”}, \\ q &= \text{“It is raining”}, \\ r &= \text{“It is cold and raining”}. \end{aligned}$$

We can use the table below to see how the truth value of r depends on the truth value of p and q. The proposition r is true only when *both* p and q are true.

p	q	r
`false`	`false`	`false`
`false`	`true`	`false`
`true`	`false`	`false`
`true`	`true`	`true`

We'll use the notation $p \wedge q$ to mean p AND q. This is called the **conjunction** of p and q and is true only when both p and q are true.

Drawing out the truth table for AND gives the following.

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

Notice that there are four rows here, not just two. This is because for each proposition, p and q, there are two possible truth values, T or F. So there are $2^2 = 4$ rows in the truth table.

Similarly, if we had a statement which used three propositions, p, q, and r, then there would be $2^3 = 8$ rows.

OR

We now introduce another proposition

$$\begin{aligned} p &= \text{“It is cold”}, \\ q &= \text{”It is raining”}, \\ s &= \text{“It is either cold or raining or both”}. \end{aligned}$$

Again the truth value of s can always be worked out from those of p and q.

p	q	s
`false`	`false`	`false`
`false`	`true`	`true`
`true`	`false`	`true`
`true`	`true`	`true`

We'll use the notation $p \vee q$ to mean p OR q. This is called the **disjunction** of p and q.

The truth value of $p \vee q$ can again be worked out using a truth table. (Notice that we have four rows again.)

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

The proposition $p \vee q$ is true when just p is true, when just q is true, or when both p and q are true. This is called the **inclusive** OR.

In ordinary English, we sometimes use the word OR in an **exclusive** sense, meaning one or the other but *not* both. For example, “Would you like tea or coffee?” - it's generally understood that you are being asked to make a choice of one or the other. The **exclusive** OR expression, p XOR q, is similar to the inclusive OR, $p \vee q$, except that it is false if p and q are both true.

Combining connectives

Suppose now that

$$p = \text{“It is cold”},$$
$$q = \text{“It is raining”},$$
$$u = \text{“It is cold but it's not raining”}.$$

We could define a new connective between p and q to give u, as above. However, there is no need, since u can be expressed by a combination of connectives we have already.

$$u = p \text{ AND } (\text{NOT } q)$$
$$u = p \wedge (\neg q)$$

The truth table can be worked out from those for $\neg$ and $\wedge$.

p	q	$\neg q$	$p \wedge \neg q$
F	F	T	F
F	T	F	F
T	F	T	T
T	T	F	F

The expression $p \wedge \neg q$ is an example of a statement formula or **compound proposition**. Its truth value is determined once we know the truth values of the **atomic propositions** p and q. Atomic propositions are those which can't be broken down into simpler propositions. E.g., “5 is a prime number” is an atomic proposition.

Using the logical connectives we can form more complicated expressions, just as we can build up ordinary algebraic expressions using addition, subtraction, multiplication, and division, etc.

Since each atomic proposition can take only two values, it is often practicable to make a complete table of possible values for a compound statement. We have seen some examples of such truth tables already. Each extra atomic proposition in a compound proposition will double the number of rows in the truth table.

In addition to the elementary connectives AND, OR and NOT, there are a number of other ones which are useful. We shall display them in the composite table below.

p	q	p XOR q	$p \Rightarrow q$	$p \Leftrightarrow q$	p NAND q	p NOR q
F	F	F	T	T	T	T
F	T	T	T	F	T	F
T	F	T	F	F	T	F
T	T	F	T	T	F	F

Propositional logic is important in understanding electronic logic circuits, which are used, for example, in designing computer processors. The connectives NAND and NOR are important in logic circuits because they can be implemented with fewer components than the others. They stand for ‘NOT AND’ and ‘NOT OR’, respectively. We shall discuss $\Rightarrow$ (IMPLIES) and $\Leftrightarrow$ (IFF) later.

Video Visit the URL below to view a video:
`https://www.youtube.com/embed/oHhFwJUHHH4`

Below is a template for a three-column truth table.

p	q	r
F	F	F
F	F	T
F	T	F
F	T	T
T	F	F
T	F	T
T	T	F
T	T	T

Precedence rules

We have sometimes used parentheses (brackets) in compound propositions so that there can be no doubt as to which operators take precedence over others. However, long expressions soon become so cluttered with parentheses that it is very difficult to understand them. Precedence rules are therefore established, just as in arithmetic multiplication and division take precedence over addition and subtraction, allowing us to ignore many of the parentheses. These state that, unless parentheses indicate otherwise, one should always perform the operators in the following order.

NOT	$\neg$	highest precedence
AND	$\wedge$	
OR	$\vee$	lower precedence
IMPLIES, etc.	$\Rightarrow$	lowest precedence

Despite this order of precedence, it's usually worth having some parentheses in an expression so that the reader doesn't have to figure it all out for themselves.

Concept Checks

Test Yourself Visit the URL below to try a numbas exam:
`https://numbas.mathcentre.ac.uk/exam/16180/propositional-logic-1/embed/`