Propositional Logic 2

Laws

When we were studying compound propositions we used truth tables. However, there is an alternative approach which is often useful. This involves manipulating the symbols according to a set of laws. Examples of such laws are the idempotence law, $p \vee p = p$, and the commutativity law, $p \wedge q = q \wedge p$.

Many of the names of the laws might seem strange and quite technical. It's not important to remember the precise names of these (you can always look them up) but it can help, when talking to others, to have a name for each law.

The set of laws and the method of applying them is known as **Boolean algebra**. There are many of them and they are all intertwined. The table below may seem daunting but by practicing with them they will become familiar, just like the rules of arithmetic.

$p \wedge p = p$	$p \vee p = p$	idempotence
$p \wedge q = q \wedge p$	$p \vee q = q \vee p$	commutativity
$(p \wedge q) \wedge r = p \wedge (q \wedge r)$	$(p \vee q) \vee r = p \vee (q \vee r)$	associativity
$p \wedge (p \vee q) = p$	$p \vee (p \wedge q) = p$	absorption
$p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$	distributivity
$p \wedge F = F$	$p \vee T = T$	annihilation
$p \wedge T = p$	$p \vee F = p$	identity
$p \wedge \neg p = F$	$p \vee \neg p = T$	cancellation
$\neg T = F$	$\neg F = T$	negation
$\neg (p \vee q) = \neg p \wedge \neg q$	$\neg (p \wedge q) = \neg p \vee \neg q$	de Morgan's laws
$p \wedge (\neg p \vee q) = p \wedge q$	$p \vee (\neg p \wedge q) = p \vee q$	negative absorption
$\neg \neg p = p$		involution

You might notice that each rule in the table above, apart from the last one, has two versions. There is a way to produce one from the other by simultaneously swapping $\wedge \leftrightarrow \vee$ and $T \leftrightarrow F$. This is a property peculiar to Boolean algebra, not seen in ordinary algebra done with numbers, called **duality**.

Alternative notation

The notation using $\vee$, $\wedge$, and $\neg$, which we have used so far, is useful when it is necessary to distinguish clearly which symbols represent logical operators. For example, if we want to say that a number x is a solution of one of two given quadratic equations, we might write $(x^2 + 3x + 1 = 0) \vee (2x^2 + 3x + 1 = 0)$.

There are situations, however, when we are not mixing arithmetic and logical operators, and then it may be convenient to use an alternative notation. In this notation $p \vee q$ is written $p + q$ and $p \wedge q$ is written as $p \cdot q$ or just pq. The negation $\neg p$ is written as $\overline{p}$. Then we denote F by 0 and T by 1.

The bar, meaning "not", can cover more than one symbol and acts like brackets. For example, $\overline{p+q}$ means $\neg (p \vee q)$. A particular case to watch out for is $\overline{p}\overline{q}$, which means $\neg p \wedge \neg q$, as opposed to $\overline{pq}$, which means $\neg (p \wedge q)$. Often this confusion can be avoided by including the 'dot': $\overline{p}\overline{q} = \overline{p} \cdot \overline{q}$. (Depending on what screen you read this on, you may not even be able to tell the difference between $\overline{p}\overline{q}$ and $\overline{pq}$. This is why the dot can be quite important!)

This notation is often used when Boolean expressions are used to represent the output from electronic logic circuits.

$pp = p$	$p + p = p$	idempotence
$pq = qp$	$p + q = q + p$	commutativity
$(pq)r = p(qr)$	$(p + q) + r = p + (q + r)$	associativity
$p(p + q) = p$	$p + pq = p$	absorption
$p(q + r) = pq + pr$	$p + qr = (p + q)(p + r)$	distributivity
$p0 = 0$	$p + 1 = 1$	annihilation
$p1 = p$	$p + 0 = p$	identity
$p\overline{p} = 0$	$p + \overline{p} = 1$	cancellation
$\overline{1} = 0$	$\overline{0} = 1$	negation
$\overline{pq} = \overline{p} + \overline{q}$	$\overline{p + q} = \overline{p} \cdot \overline{q}$	de Morgan's laws
$p(\overline{p} + q) = pq$	$p + \overline{p} \cdot q = p + q$	negative absorption
$\overline{\overline{p}} = p$		involution

The connectives in Boolean algebra have the same precedence as their corresponding notation in ordinary algebra. This can help when manipulating Boolean expressions because many of the laws above look like their ordinary algebra counterparts. Others look just as unusual in this notation as they do in the standard notation.

Implication

When we introduced connectives we briefly mentioned the connective $\Rightarrow$, which stands for IMPLIES. The most common way to read $p \Rightarrow q$ is "if p, then q". This is only roughly equivalent to the logical connective $\Rightarrow$ and can be misleading at times but it can be helpful to think of it in this way.

We'll try to understand its truth table by using an example. Suppose that a train has a safety system where pulling an alarm handle puts the brakes on. Two relevant propositions are

$$p = \text{"The alarm handle is pulled"},$$
$$n = \text{"The train stops"}.$$

The condition which says the system is working properly is

$$r = \text{"If the alarm handle is pulled, then the train stops"}.$$

A table which gives the truth values for r based on those of p and q is shown below.

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

The first row essentially says "If the alarm handle is *not* pulled, then the train *doesn't* stop" and has a value of `true`, which makes sense. The second row says that "If the alarm hand is *not* pulled, then the train stops". This also has value `true`, essentially saying that the train might stop even if the handle isn't pulled (in a railway station, for example).

The third row says "If the alarm handle is pulled, then the train *doesn't* stop". The value is false, essentially meaning that the safety system is faulty. The final row says "If the alarm handle is pulled, then the train stops", which is what we would most often interpret the original statement as if written in plain English.

The truth table for $p \Rightarrow q$, written in the standard notation, is given below.

p	q	$p \Rightarrow q$
F	F	T
F	T	T
T	F	F
T	T	T

The IMPLIES connective can be confusing. It can be helpful to use an equivalent formulation when working with Boolean algebra, which is that $p \Rightarrow q = \neg\, p \vee q$.

Some other English statements which are used to interpret the proposition $p \Rightarrow q$ are as follows.

- p only if q
- q if p,
- p is a **sufficient** condition for q,
- q is a **necessary** condition for p.

It can sometimes come in useful to know that $p \Rightarrow q = \neg\, q \Rightarrow \neg\, p$.

Video Visit the URL below to view a video:
`https://www.youtube.com/embed/_QbWobByNj8`

Video Visit the URL below to view a video:
`https://www.youtube.com/embed/Sd17TEZZdsE`

Video Visit the URL below to view a video:
`https://www.youtube.com/embed/dleXo5KuQE8`

Double Implication

One final connective which we will use is IFF, which is denoted by the symbol $\Leftrightarrow$. The letters IFF stand for "if and only if" and for propositions p and q, we define $p \Leftrightarrow q = (p \Rightarrow q) \wedge (q \Rightarrow p)$. We can also say in the case that $p \Leftrightarrow q$ that "p is a necessary and sufficient condition for q".

Both $\Rightarrow$ and $\Leftrightarrow$ have lower precedence than both $\vee$ and $\wedge$.

Building New Connectives

Sometimes we will find that we keep using an expression over and over again, so we can introduce notation in order to simplify our working. An example of this is the `XOR` connective (exclusive `OR`), which is built out of the connectives `NOT`, `AND`, and `OR`.

$$a \texttt{ XOR } b = (a \wedge \neg\, b) \vee (\neg\, a \wedge b)$$

It's useful to have the shorthand, so that we don't have to write the full right-hand side expression out each time. An example of a more complicated connective can be seen in the video.

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

Concept Checks

Test Yourself Visit the URL below to try a numbas exam:
`https://numbas.mathcentre.ac.uk/exam/16181/propositional-logic-2/embed`