Logical Implications

Gilles Cazelais

Consider the following insurance contract.

If your car is stolen, then you will receive \$10 000.

In which cases is the contract violated? It can only happen when your car is stolen and you do not get \$10 000. In all other cases, the contract is respected. We can summarize all cases as follows.

Your car is stolen.	You receive \$10 000	The contract is respected.
True	True	True
True	False	False
False	True	True
False	False	True

The third row would require a very generous insurance company but in no way would they violate the contract by giving you \$10 000 even if your car was not stolen!

Definition. Let p and q be two propositions. The *implication* $p \to q$ is the proposition defined by the following truth table.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

The implication $p \to q$ can be expressed in words as: "if p, then q".

To further motivate the definition of $p \to q$, let's consider the following proposition.

For all real numbers x, if $x > 2$, then $x^2 > 4$.

This is a true proposition about real numbers.

Consider the three values $x = 3, -3$, and 1.

x	$x > 2$	$x^2 > 4$
3	True	True
-3	False	True
1	False	False

The three values of x lead to different pairs of truth values for $x > 2$ and $x^2 > 4$ but in all three cases, the implication $(x > 2) \to (x^2 > 4)$ is true according to the definition of $p \to q$. It is important to observe that the only way the implication could be false is if we could find some real number x such that $x > 2$ but $x^2 \leq 4$.

⋆⋆⋆

The conditional `if-then` used in several programming languages should not be confused with logical implications. Suppose we encounter

`if` p `then` S

in some programming language, the statement S will be executed if p is true. If p is false, statement S will not be executed. In general, S is not a statement that has a truth value. It is some statement in the language that is executable.

Let's consider an example from the computer language Python.

```
>>> x = 1
>>> if x > 0:  x = x + 1
>>> print x
2
```

We see that statement `x = x + 1` was executed since `x > 0` was True.

```
>>> x = -2
>>> if x > 0:  x = x + 1
>>> print x
-2
```

This time, statement `x = x + 1` was not executed since `x > 0` was False.