Truth tables

The truth value of a compound statement can be deduced from that of the original propositions using a truth table. This table has a row for each possible combination of truth values of the simple propositions and will have a column for the truths of each proposition constructed along the way to the final compound proposition. The tables for negation, conjunction and disjunction are

P	**not** P
T	F
F	T

P	Q	P **and** Q
T	T	T
T	F	F
F	T	F
F	F	F

P	Q	P **or** Q
T	T	T
T	F	T
F	T	T
F	F	F

Note: If the compound proposition has one simple proposition as in the case of negation there are two rows in the truth table. If the compound proposition involves two simple propositions as in the case of conjunction or disjunction there are four rows in the truth table. If the compound proposition involves three simple propositions there are eight rows in the truth table, etc.

Example: Compute the truth table for **not** (P **and** Q). Two simple propositions so 4 rows.

P	Q	P **and** Q	**not** (P **and** Q)
T	T	T	F
T	F	F	T
F	T	F	T
F	F	F	T

Example: Compute the truth table for (**not** P) **or** (**not** Q).

P	Q	**not** P	**not** Q	(**not** P) **or** (**not** Q)
T	T	F	F	F
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

Note: It seems as if the two previous examples are different descriptions of the same compound proposition.

Logical Equivalence: It can happen that two combinations of applications of the logical operators give compound statements which have the same truth values for every possible set of truth values of the original propositions. Such compound statements are said to be logically equivalent. This can be checked using a combined truth table. We write $R \equiv S$ to denote the fact that R and S are equivalent statements.

Note: It should not be too much of a surprise that logical equivalences arise. Suppose we just consider compound statements using two simple propositions P and Q. The truth table has four rows so the last column is one of 16 possibilities. (T or F in the first row, T or F in the second row, etc.) Therefore lots of compound statements in P and Q must be equivalent.

Example: Perhaps the simplest equivalence is **not** (**not** P) ≡ P.

Example: **not** (P **or** Q) is logically equivalent to (**not** P) **and** (**not** Q):

P	Q	P **or** Q	**not** (P **or** Q)	**not** P	**not** Q	(**not** P) **and** (**not** Q)
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

Here column 3 is the disjunction of columns 1 and 2, column 4 is the negation of column 3, column 5 is the negation of column 1, column 6 is the negation of column 2 and column 7 is the conjunction of columns 5 and 6. Since the fourth and seventh columns are the same the corresponding propositions are logically equivalent.

Example: In the logicians joke the negation of
(logician 1 wants beer) **and** (logician 2 wants beer) **and** (logician 3 wants beer)
is the statement
(logician 1 does not want beer) **or** (logician 2 does not want beer) **or** (logician 3 does not want beer)

Implication

Sometimes we wish to state that the truth of proposition P guarantees the truth of proposition Q. This is known as a conditional proposition and written $P \Rightarrow Q$. Some books may use $P \rightarrow Q$. As an example of implication, consider the propositions

P: It is raining

Q: I drive to work

Here $P \Rightarrow Q$ is the conditional statement 'If it is raining, I drive to work.' Note that it does not mean that I only drive to work when it is raining. I could drive to work for some other reason but I will definitely drive to work if it is raining. This conditional statement is false if P is true and Q is false, otherwise it is true. Thus it is equivalent to (**not** P) **or** Q. The table for $P \Rightarrow Q$ is

P	Q	$P \Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Exercise: Check for yourself using a truth table that $(P \Rightarrow Q) \equiv$ ((**not** P) **or** Q).

Example: The proposition ((**not** Q) $\Rightarrow$ (**not** P)) is equivalent to $(P \Rightarrow Q)$ and is called the contrapositive of $(P \Rightarrow Q)$.

P	Q	$(P \Rightarrow Q)$	**not** P	**not** Q	((**not** Q) $\Rightarrow$ (**not** P))
T	T	T	F	F	T
T	F	F	F	T	F
F	T	T	T	F	T
F	F	T	T	T	T

Predicates
Sometimes we want our statements to contain variables and their truth to depend on the value of the variable. Such statements are called predicates.

For example, the statements
  x is a computing student who likes mathematics,
  x is a mathematician who can program,
  x is an integer satisfying $x^2 > 3$
are predicates.

The first can be true or false depending on which computing student x is. Similarly, the second can be true or false depending on which mathematician x is. Finally, the third statement can be true or false depending on which integer x is. Specifically, the third statement is true for $x \leq -2$ or $x \geq 2$ but false for $x =, -1, 0, 1$.
(Remember that the integers are the whole numbers $\ldots -2, -1, 0, 1, 2, 3, \ldots$)

As such, predicates are not propositions since they do not have a truth value.

Predicates arise in programming in code such as

while $(x < 5)$ ...

Here the statement $x < 5$ is a predicate and its truth depends on the value of x which changes as the programme runs.

Quantifiers
These predicates can be turned into propositions using quantifiers such as ‘for all’ and ‘there exists’.

For example, the third predicate above
  x is an integer satisfying $x^2 > 3$
can be turned into the proposition
  For all integers x, $x^2 > 3$.

Now we can assign a truth value. Specifically, this new proposition is false since there are some integers, $x = 0$ for example, with $x^2 \leq 3$.