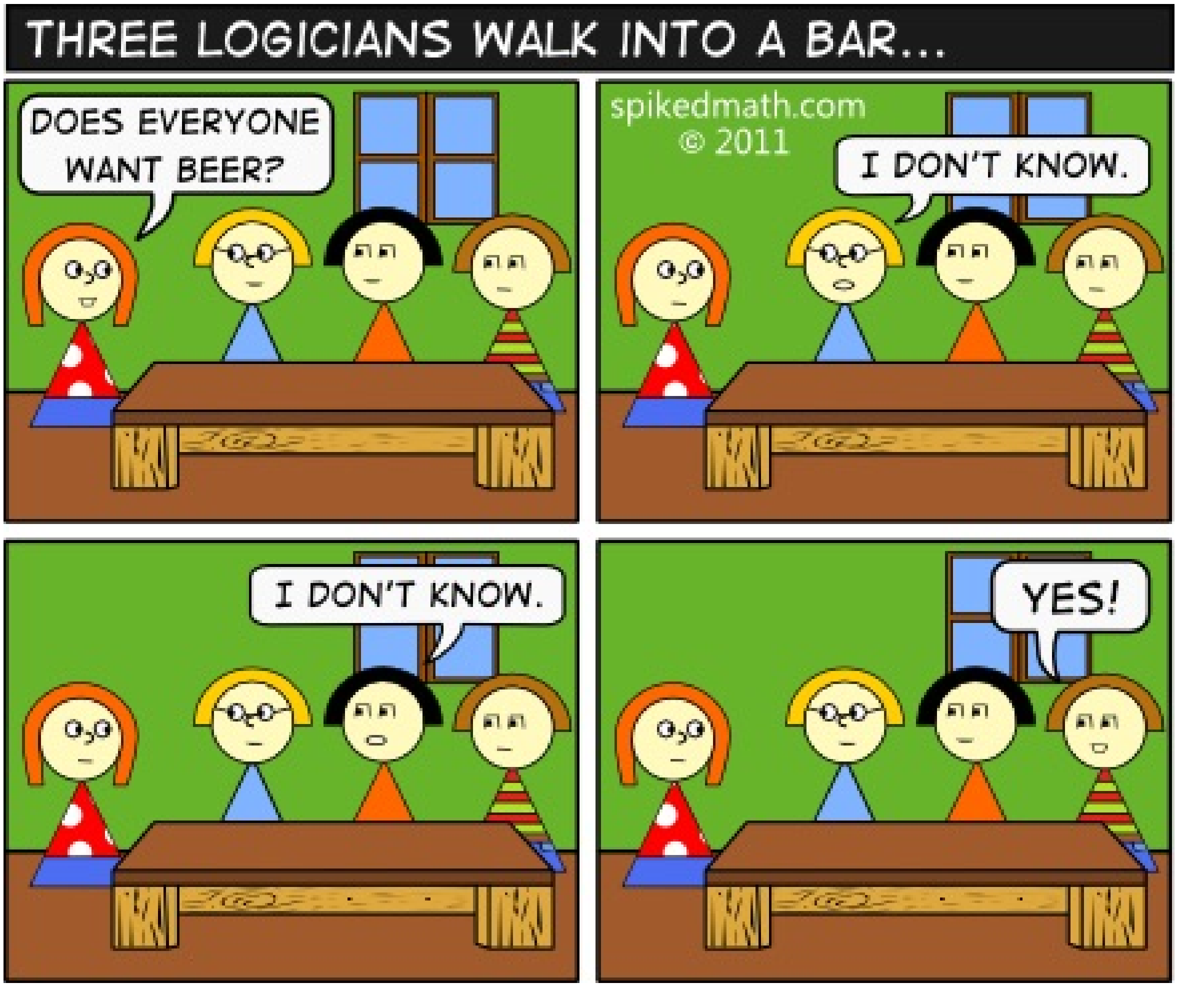
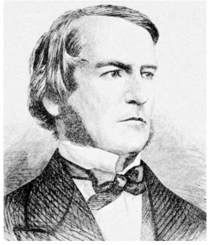
Chapter 1: Logic, truth tables and methods of proof.



Propositions

A proposition is a statement which has a truth value; it is either true (T) or false (F). Compare Boolean data types in programming. Here's a picture of Boole.

Boole: 

He lived in Cork. Look him up. In particular find out how he died.

Examples of Propositions are

the earth is flat
Sarah is a doctor
29 is a prime number.

The first is false and the third is true; the second depends on which 'Sarah' we are talking about, but assuming we agree on who she is we should be able to say if she is a doctor or not.

Sentences of the form

Ireland will win UEFA Euro 2020.
When is the next bus?

are not propositions. The first is a prediction whose truth value we cannot know now and the second is a question.

Since we will be doing operations on propositions and do not want to write the full statements each time we usually abbreviate propositions using capital letters. For example,

P: the earth is flat
Q: Sarah is a doctor
R: 29 is a prime number.

Logical operators

Logical operators allow us to construct new propositions called compound propositions from existing ones.

Negation: The simplest logical operator is negation. The negation of the general proposition P is the proposition **not** P. This is sometimes written as $\neg$ P. This proposition has truth value T when P is false and truth value F when P is true. For example,

not Q: Sarah is not a doctor

Conjunction: The conjunction logical operator produces a compound proposition P **and** Q from a general pair of propositions P and Q. This is sometimes written as P $\wedge$ Q. This proposition is true if both P and Q are true, otherwise it is false. For example,

P **and** R: the earth is flat and 29 is a prime number

Example: In the logicians joke the statement being considered was a conjunction:

(logician 1 wants beer) **and** (logician 2 wants beer) **and** (logician 3 wants beer)

Disjunction: The disjunction logical operator produces a compound proposition P **or** Q from a general pair of propositions P and Q. This is sometimes written as P $\vee$ Q. This proposition is only false if both P and Q are false, otherwise it is true. For example,

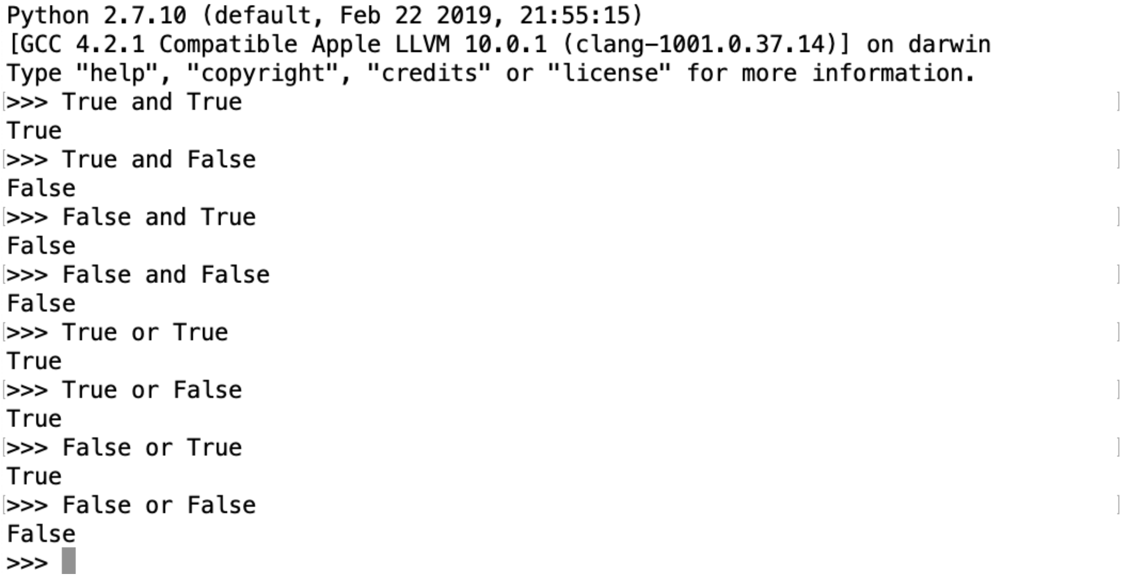
P **or** R: the earth is flat or 29 is a prime number

Note that this **or** is an ‘inclusive or’, meaning that P **or** R is true if P is true or R is true or both P and R are true. (Look up the **XOR** logical operator on wikipedia.)

Example: The following example may help to remember the values of **and** , **or** :
Suppose I want to get home by bus:
(a) I will get home if bus A **or** bus B arrives. Of the four possibilities only one means I do not get home, namely when neither bus arrives.

(b) The journey is in two parts. I will get home if bus C arrives **and** bus D arrives where bus C lets me off. Of the four possibilities only one means I get home, namely when both buses arrive.

Python agrees: Python understands 'True', 'False', 'not', 'and' and 'or'. Here's what it understands by 'and' and 'or'.



```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> True and True
True
>>> True and False
False
>>> False and True
False
>>> False and False
False
>>> True or True
True
>>> True or False
True
>>> False or True
True
>>> False or False
False
>>> 
```