The results from the test will be back to us next week.

Introduction to Logic

Logic

Logic is the study of rational argument.

Given a set of facts, and an argument claiming that a conclusion follows from these facts, how do we decide whether or not the argument is justified?

To do this we need a precise language in which to talk.

First studied by the ancient Greeks, e.g. Aristotle.

Boole published "An Investigation of the Laws of Thought" in 1854.

Why is logic useful?

- can design circuits with it
- design programs etc.
- prove that a security system stops unauthorised access
- verify that circuit designs do the job they are supposed to do
- check that databases contain consistent info
- build systems that understand natural language
- build artifically intelligent systems

Note: he's going extremely quickly, hard to take things down

Logic and humans

We are not good at logic:

- we find it difficult to ignore our emotions
    - are our emotions never relevant, though?
- we jump to conclusions based on what we want to happen
- we find it difficult to follow formal arguments

What will we be looking at?

Propositional logic

We consider the logic of simple statements:

- how to write them
- how to build complex statements
- how to determine whether they are true or false
- how to construct proofs that they are true

Predicate logic

In CS1113 we will move on to more complicated statements which involve *variables* and collections of objects, and we will do similar things with them.

Simple statements

An *atomic statement* or *proposition* is a phrase or sentence that declares a fact and is either true or false.

E.g.:

- Dublin is the capital of Ireland
- Homer Simpson lives in Cork
- 7 + 5 = 1

The following are not propositions:

- Are we nearly there yet?
- Look before you leap.
- x = y + 2

Note that the variables x and y do not have defined values, so the statement cannot be a proposition.

Truth Values

You can picture a function from the set of all propositions to the set {True, False}. These are often denoted {T, F}.

Compound Statements

We can build more complicated statements by combining simple statements:

1. Dublin is the capital of Ireland *or* Enda Kenny played soccer for Bohemians. *This is true since one of the substatements is true.*

2. etc

A compound statement is a combination of simple statements combined with connectives.

A formal language

Lowercase letters to represent propositions.

T, F to represent the truth values true and false

Connectives:

- ¬ — not
- ∧ — and
- ∨ — or
- → — implies
- ↔ — if and only if (equivalency)

Negation (not)

Switches the truth value of the proposition it is applied to.

I.e. if it is applied to p:

- if p is False, ¬p is True
- if p is True, ¬p is False

Example

If p represents the proposition "Dublin is the capital of Ireland", then ¬p (which has the value F), represents "Dublin is not the capital of Ireland".

Conjunction (and)

The conjunction of two propositions is true if both of the smaller ones are true, otherwise it is false.