Introduction to Logic 2

Last time:

- What is Logic and why study it?
- Object-language/Meta-language.
- Propositions, Beliefs and Contradictions.
- Formalization

This time:

- The Propositional Calculus.
- The Language of Propositional Logic.
- What does it all mean?
- Arguments

Introduction to the Propositional Calculus

• The *Propositional Calculus* (PC) is a simple language for expressing certain kinds of propositions.

• Essentially, it allows us to write down Boolean combinations of simple declarative sentences.

For example:

- Logic is fun
- Logic is *not* fun
- Bill teaches Logic *and* Logic is fun
- Logic is fun *or* Bill is happy
- *if* Logic is fun, *then* Bill is happy

Connectives

- Statements are combined with words such as *not*, *and*, *or* and *if ..., then* to build more complex statements
  - The words will be called the **connectives**
  - The connectives are **truth-functional**

  The statement

  "It is raining and it is snowing"

  is **true** if
  - the statement "it is raining" is **true**; and
  - the statement "it is snowing" is **true**

  otherwise it is **false**.

- The truth-value of the whole can be calculated once the truth-values of the parts are known.

The Language of PC

- Rather than using the English connectives, we will use the following symbols:

Symbol	English Equivalent
$\neg$	not
$\wedge$	and
$\vee$	or
$\rightarrow$	implies (if ... then ...)
$\leftrightarrow$	... if and only if ...

- We will also require some further symbols:
  - left and right parentheses: '(' and ')'
  - a stock of *propositional variables* : $p, q, r, s, \ldots$

  These symbols (connectives, propositional variables, parentheses) form the *alphabet* of our language.

The *well-formed formulas* (wffs) of the language are strings (i.e. sequences) of symbols from the alphabet.

Definition: *(The language of the PC)*

1. *Any propositional variable is a wff;*
2. *If A and B are wffs, then so are:*
   - $(\neg A)$
   - $(A \wedge B)$
   - $(A \vee B)$
   - $(A \to B)$
   - $(A \leftrightarrow B)$
3. *Nothing is a wff except in virtue of 1 and 2 above.*

NB: I may also refer to a wff as a *sentence* or a *statement*.

Which of these are wffs?

p

$(p \wedge q)$

$(p \wedge (\neg q))$

$((p \rightarrow q) \vee (\neg r))$

$(p \rightarrow \wedge q)$

$p \vee q$

$(A \wedge B)$

A Note about Brackets

In practice, we adopt conventions that permit parentheses to be dropped where no confusion can arise from doing so

Operator Precedence: $\neg$ takes precedence over $\wedge$ and $\vee$, so:

$$(((\neg p) \wedge q) \to r)$$

becomes

$$((\neg p \wedge q) \to r)$$

and $\wedge$ and $\vee$ in turn take precedence over $\to$ and $\leftrightarrow$ so:

$$((\neg p \wedge q) \to r)$$

becomes

$$(\neg p \wedge q \to r)$$

Outermost Parentheses: these can always be dropped, so:

$$(\neg p \wedge q \to r)$$

becomes

$$\neg p \wedge q \to r$$

What does it all mean?

- We have provided a definition of the wffs of the PC.
  - This tells us what *form* they take (their *syntax*)
  - It does not tell us about their meaning (their *semantics*)
- **Questions:**
  - How do we assign meaning to the sentences of the PC?
  - What do we mean by ‘meaning’ anyway?

Meaning and Truth

- Sentences express propositions, and these may be either **true** or **false**.
  - we say that sentences denote *truth-values*
- We have already noted that the connectives are *truth-functional*
  - they allow us to calculate the meaning (i.e. truth-value) of complex sentences as a function of the meaning (i.e. truth-values) of their parts.

Arguments

- Part of our motivation for introducing Propositional Logic is to formalize and study 'patterns of reasoning' or *arguments*.
- We have seen some examples of arguments, both 'good' and 'bad':

  e.g.

  If Logic is fun, then Bill is happy

  Logic is fun

  Therefore, Bill is happy

  If Logic is fun, then Bill is happy

  Programming is fun

  Therefore, Bill is happy

- We can now ‘formalize’ these arguments (and many others) in the sense of providing an abstract representation of their *structure*.

- Consider the first argument given on the last overhead. Let

  p stand for ‘Logic is fun’

  q stand for ’Bill is happy’

  then, we can write:

  $$\begin{array}{l} (p \to q) \\ p \\ \hline q \end{array}$$

- Or perhaps:

$$((p \to q) \wedge p) \to q$$

Summary

- The Propositional Calculus (PC) is a simple language for expressing propositions
- Sentences of the PC are built up from propositional variables, connectives and parentheses.
- Sentences of the PC are either **true** or **false**
- Connectives are truth-functional
- The PC allows us to formalize the structure of arguments
- The formal representations are *abstract*.