

# CO140 - Logic

## Introduction

A logic system consists of 3 things:

1. Syntax - formal language used to express concepts
2. Semantics - meaning for the syntax
3. Proof theory - syntactic way of identifying valid statements of language

Considering the basic example in a program, we can then see the features;

```
if count > 0 and not found then
    decrement count;
    look for next entry;
end if
```

1. basic (**atomic**) statements (**propositions**) are either  $\top$  or  $\perp$  depending on circumstance;
  - i. `count > 0`
  - ii. `found`
2. **boolean operations**, such as **and**, **or**, **not**, etc. are used to build complex statements from **atomic propositions**
3. the final statement `count > 0 and not found` evaluates to either  $\top$  or  $\perp$

## Syntax

The formal language of logic consists of three ingredients;

1. Propositional atoms (propositional variables), evaluate to a truth value of either  $\top$  or  $\perp$ . These are represented with letters;  $p, p', p_0, p_1, p_2, p_n, q, r, s, \dots$
2. Boolean connectives;
  - **and** is written as  $p \wedge q$   $p$  and  $q$  both hold
  - **or** is written as  $p \vee q$   $p$  or  $q$  holds (or both)
  - **not** is written as  $\neg p$   $p$  does not hold
  - **if-then / implies** is written as  $p \rightarrow q$  if  $p$  holds, then so does  $q$
  - **if-and-only-if** is written as  $p \leftrightarrow q$   $p$  holds if and only if  $q$  holds
  - **truth**, and **falsity** are written as  $\top$ , and  $\perp$  respectively. logical constants
3. Punctuation. Similar to arithmetic, the lack of brackets can make an expression ambiguous. For example,  $p_0 \vee p_1 \wedge p_2$  can be read as either  $(p_0 \vee p_1) \wedge p_2$  or  $p_0 \vee (p_1 \wedge p_2)$ , which are different. The latter is the correct interpretation due to binding conventions.

We can order the boolean connectives by decreasing binding strength;

(strongest)  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$  (weakest)

While repeated disjunctions ( $\vee$ ), and conjunctions ( $\wedge$ ) are fine, as  $p \wedge q \wedge r$  is equivalent to  $p \wedge (q \wedge r)$ , and the same for  $\vee$ , due to associativity, the same isn't true for  $\rightarrow$ . Due to the ambiguity, brackets should always be used.

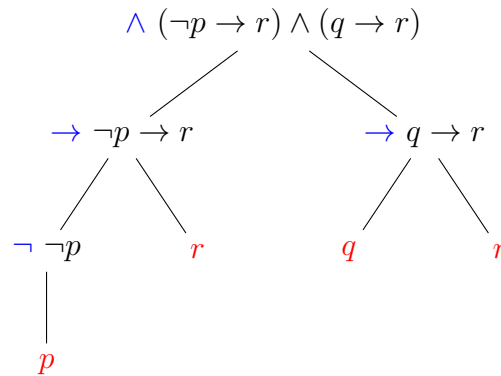
There are also exceptions to the rule, for example with  $p \rightarrow r \wedge q \rightarrow r$  - this should be  $p \rightarrow (r \wedge q) \rightarrow r$  according to our binding conventions, but brackets should be used to ensure the correct interpretation.

## Formulas

Something is a **well-formed formula** only if it is built from the following rules (the brackets are required);

1. a propositional atom ( $p, p', p_0, p_1, p_2, p_n, q, r, s, \dots$ ) is a propositional formula
2.  $\top$ , and  $\perp$  are both formulas
3. if  $A$  is a formula, then  $(\neg A)$  is also a formula
4. if  $A$ , and  $B$  are both formulas, then  $(A \wedge B), (A \vee B), (A \rightarrow B), (A \leftrightarrow B)$  are also formulas

We can also create a tree to parse a logical formula, for example;  $(\neg p \rightarrow r) \wedge (q \rightarrow r)$



Note that this tree shows the principal connective in blue, and the propositional atoms in red. Note that  $\wedge$  is the principal connective in the top layer, and it therefore has the general form  $A \wedge B$ , and so on going down.

## Definitions

- A formula is a **negated formula** when it is in the form  $\neg A$ , negated atoms are sometimes called **negated-atomic**.
- $A \wedge B$ , and  $A \vee B$  are **conjunctions**, and **disjunctions**.  $A$ , and  $B$ , are **conjuncts**, and **disjuncts**, respectively.
- $A \rightarrow B$  is an implication.  $A$  is the **antecedent**, and  $B$  is the **consequent**

## Semantics

The connectives covered above have a rough English translation. However a natural language has ambiguity, and as engineers, we need precise meanings for formulas. This is the truth table for every connective that will be used in this course (?):

$p$	$q$	$\top$	$\perp$	$p \wedge q$	$p \vee q$	$\neg p$	$p \rightarrow q$	$p \leftrightarrow q$	$p \uparrow q$
0	0	1	0	0	0	1	1	1	1
0	1	1	0	0	1	1	1	0	1
1	0	1	0	0	1	0	0	0	1
1	1	1	0	1	1	0	1	1	0

Note how we can also define new connectives (see how  $A \uparrow B$  was defined in the last column); this is a NAND connective - equivalent to  $\neg(A \wedge B)$ .

## Translation

### English to Logic

- **but** means **and**

"I will go out, but it is raining"  $(i \text{ will go out}) \wedge (it \text{ is raining})$

- **unless** generally means **or**

"I will go out unless it rains"  $(i \text{ will go out}) \vee (it \text{ will rain})$  (note the will)  
 $\neg(it \text{ will rain}) \rightarrow i \text{ will go out}$

There is also the strong form of **unless**, but in we generally use the weak form in computing

$(i \text{ will go out}) \leftrightarrow \neg(it \text{ will rain})$

- **or** generally refers to exclusive or (strong reading) in English, but it can also refer to inclusive or (weak reading). However, we always take the weak reading in computing.

### Modality

I don't know what this means, so I'm just ignoring it for now

### Logic to English

While the others are slightly more straightforward,  $\rightarrow$  is a pain to translate.

For example,  $(i \text{ am the pope}) \rightarrow (i \text{ am an atheist})$  evaluates to true, as falsity implies anything, however if we were to translate it into English, "If I am the Pope, then I am an atheist" is (most likely) untrue.

Another example is the following;  $p \wedge q \rightarrow r$ , and  $(p \rightarrow r) \vee (q \rightarrow r)$  are logically equivalent, but can be translated into different meanings. For example, let  $p$  be "event A happens", let  $q$  be "event B happens", and  $r$  be "event C happens". The former can be translated to "If both A and B happens, then C happens", whereas the latter becomes "If A happens, then C happens, or if B happens, then C also happens".

## Arguments

We use the double turnstile,  $\models$  (`\vdash` in L<sup>A</sup>T<sub>E</sub>X), to mean **therefore**. For example, the *Socrates syllogism* can be expressed as  $(\text{socrates is a man}), (\text{men are mortal}) \models (\text{socrates is mortal})$  in logic, and in English as;

- Socrates is a man
- Men are mortal
- Therefore, Socrates is mortal

The definition of a valid argument is as follows;

Given valid formulas  $A_1, A_2, \dots, A_n, B$ , and ' $A_1, \dots, A_n$  therefore  $B$ ', we can write it as  $A_1, \dots, A_n \models B$ , iff  $B$  is true in every situation where  $A_1, \dots, A_n$  are all true.

### Examples

- $A, A \rightarrow B \models B$  **modus ponens**
- $A \rightarrow B, \neg B \models \neg A$  **modus tollens**
- $A \rightarrow B, B \not\models A$   $A$  can be false, as falsity implies anything

## Definitions

- A propositional formula is logically **valid** if it's true in all situations ( $\models A$ ), if  $A$  is **valid**
- A propositional formula is **satisfiable** if it's true in at least one situation (hence **valid**  $\rightarrow$  **satisfiable**)
- Two propositional formulas are logically **equivalent** if they are true in the same situations.

argument	validity	satisfiability	equivalence
$A \models B$	$A \rightarrow B$ valid	$A \wedge \neg B$ unsatisfiable	$(A \rightarrow B) \equiv \top$
$\top \models A$	$A$ valid	$\neg A$ unsatisfiable	$A \equiv \top$
$A \not\models \perp$	$\neg A$ not valid	$A$ satisfiable	
$A \models B$ , and $B \models A$		$A \leftrightarrow \neg B$ unsatisfiable	$A \equiv B$

(copied directly from *Propositional Logic - Arguments and Validity.pdf*)

## Validity

The main ways used to check validity are as follows;

- Truth tables - check all possible situations, and check the results of each formula are  $\top$
- Direct argument
- Equivalences - using equivalences to reduce the initial formula to  $\top$
- Various proof systems - including Natural Deduction

In general, if we want to show that  $A$  is logically equivalent to  $B$ , we need to show  $A \leftrightarrow B$  is **valid**.

## Truth Tables

The use of truth tables to prove validity is fairly self-explanatory; as we're testing each situation, it's the easiest method (and it works for propositional logic since we have a finite number of configurations - doesn't work for first-order), however it's inelegant, and quite tedious depending on the number of propositional atoms.

For example, if we were to prove  $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$  is valid, we have to evaluate all of the subformulas.

$p$	$q$	$p \rightarrow q$	$\neg p$	$\neg p \vee q$	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$
0	0	1	1	1	1
0	1	1	1	1	1
1	0	0	0	0	1
1	1	1	0	1	1

(copied directly from *Propositional Logic - CheckValidity.pdf*)

As the propositional formula has 1 in all four possible configurations of  $p$ , and  $q$ , we can then say it is valid, and as such,  $p \rightarrow q$  is logically equivalent to  $\neg p \vee q$

## Direct Argument

We can show the validity of  $((p \rightarrow q) \rightarrow p) \rightarrow p$  (known as *Peirce's law*) with direct argument.

We can take an argument by cases, either  $p$  is  $\top$  or  $p$  is  $\perp$ .

- $p \leftrightarrow \top$  - we know this is true as  $A \rightarrow B$  is  $\top$  whenever  $B$  is  $\top$

- $p \leftrightarrow \perp$  - we have  $p \rightarrow q$  evaluating to  $\top$ , as  $A \rightarrow B$  is  $\top$  whenever  $A$  is  $\perp$ . As such, this formula is evaluated to  $(\top \rightarrow p) \rightarrow q$ . However, we know that  $p$  is  $\perp$ , hence we have  $\top \rightarrow \perp$ , which we know evaluates to  $\perp$  by the truth table for  $\rightarrow$ . As such, we have  $\perp \rightarrow p$ , hence it follows that it is valid, seeing as  $A \rightarrow B$  is  $\top$  whenever  $A$  is  $\perp$ .
- This is an argument by cases, known as **law of excluded middle** (you will use this often in Natural Deduction).

## Equivalences

Refer to *Logic cribsheet.pdf* for a full list of equivalences

1.  $A \wedge B \equiv B \wedge A$  commutativity of  $\wedge$
2.  $A \wedge A \equiv A$  idempotence of  $\wedge$
3.  $A \wedge \top \equiv A$
4.  $\perp \wedge A, \neg A \wedge A \equiv \perp$
5.  $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$  associativity of  $\wedge$
6.  $A \vee B \equiv B \vee A$  commutativity of  $\vee$
7.  $A \vee A \equiv A$  idempotence of  $\vee$
8.  $\perp \vee A \equiv A$
9.  $\top \vee A, \neg A \vee A \equiv \top$
10.  $(A \vee B) \vee C \equiv A \vee (B \vee C)$  associativity of  $\vee$
11.  $\neg \top \equiv \perp$
12.  $\neg \perp \equiv \top$
13.  $\neg \neg A \equiv A$
14.  $A \rightarrow A \equiv \top$
15.  $\top \rightarrow A \equiv A$
16.  $A \rightarrow \top \equiv \top$
17.  $\perp \rightarrow A \equiv \top$
18.  $A \rightarrow \perp \equiv \neg A$
19.  $A \rightarrow B \equiv \neg A \vee B$
20.  $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A) \equiv (A \wedge B) \vee (\neg A \wedge \neg B) \equiv \neg A \leftrightarrow \neg B$
21.  $\neg(A \leftrightarrow B) \equiv \neg A \leftrightarrow B \equiv \dots$  the rest can be derived from the above
22.  $\neg(A \wedge B) \equiv \neg A \vee \neg B$  de Morgan laws
23.  $\neg(A \vee B) \equiv \neg A \wedge \neg B$  de Morgan laws
24.  $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
25.  $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
26.  $A \vee (A \wedge B) \equiv A \vee (A \wedge B) \equiv A$

## Normal forms

- A formula is in **disjunctive** NF (**DNF** -  $\vee$ ) if it's a disjunction of conjunctions of literals
- A formula is in **conjunctive** NF (**CNF** -  $\wedge$ ) if it's a conjunction of disjunctions of literals (a conjunction of clauses)

## Rewriting

1. Get rid of  $\rightarrow$ , and  $\leftrightarrow$

Replace  $A \rightarrow B$  with  $\neg A \vee B$

Replace  $A \leftrightarrow B$  with  $(A \wedge B) \vee (\neg A \wedge \neg B)$

2. Use de Morgan laws to push negations down to the atoms
3. Delete double negations (replace  $\neg\neg A$  with  $A$ )
4. Rearrange with distributivity equivalences to the desired form
5. Use the equivalences which reduce two atoms to one ( $\perp \vee A \equiv A$  etc.) until no further progress can be made

## Example

Write  $p \wedge q \rightarrow \neg(p \leftrightarrow \neg r)$  in DNF

- $p \wedge q \rightarrow \neg(p \leftrightarrow \neg r)$
- $\neg(p \wedge q) \vee \neg(p \leftrightarrow \neg r)$  remove  $\rightarrow$
- $\neg(p \wedge q) \vee \neg((p \wedge \neg r) \vee (\neg p \wedge r))$  remove  $\leftrightarrow$
- $\neg p \vee \neg q \vee \neg(p \wedge \neg r) \wedge \neg(\neg p \wedge r)$  de Morgan
- $\neg p \vee \neg q \vee (\neg p \vee r) \wedge (p \vee \neg r)$  de Morgan