Fundamentals of Propositional Logic

1 The Rudiments of Propositional Logic

Propositional Logic:

- The most fundamental logic, lying at the heart of many other things
- Formalises day-to-day, common sense reasoning

Key to propositional logic are **propositions**:

- Declarative sentences can be either **true** or **false**

Propositions are represented by **propositional variables** (**Boolean variables, atoms**)

- Usually letters such as **x,Y,a** or subscripted letters such as $\mathbf{x_2}, \mathbf{Y_0}, \mathbf{a_1}$
- Which can take a truth value T (true) or F(false)

Syntax

New propositions called **formulae** or **Boolean formulae** or **propositional formulae** or **compound propositions** are formed from propositional variables and formulae by the use of logical operators

- $\wedge$ - conjunction(and)
- $\vee$ - disjunction(or)
- $\neg$ - negation(not)
- $\Rightarrow$ - implies (if left statement true, then right statement must be true, if LHS false, whole statement becomes true)
- $\Leftrightarrow$ - if and only if (iff)

2 Some formulae

2.1 Construction

The operators $\wedge, \vee, \Rightarrow$ and $\Leftrightarrow$ take two propositional formulae φ and ψ

The operator $\neg$ takes one propositional formula φ and yields a new one

- $\neg\varphi$

2.2 Use of parentheses

$(\varphi \wedge \psi) \vee \chi$ means first build $\varphi \wedge \psi$ and then build $(\varphi \wedge \psi) \vee \chi$
$\varphi \wedge (\psi \vee \chi)$ means first build $\psi \vee \chi$ and then build $\varphi \wedge (\psi \vee \chi)$

2.3 Some typical well formed formulae

$$\neg((\neg b \wedge a) \Rightarrow (c \vee \neg d))$$

$$((a \wedge \neg a) \vee ((b \vee c) \vee d)) \Leftrightarrow d$$

$$(((a \Rightarrow b) \Rightarrow c) \Rightarrow d)$$

3 Semantics of propositional logic

Semantics: all propositional variables take the value **T(True)** or **F(false)**

- The value of a formula under some **truth assignment** is ascertained by using the **truth tables** for the above logical connectives

The truth tables for our logical connectives are as follows:

p	q	$p \wedge q$	$p \vee q$	$\neg p$	$p \Rightarrow q$	$p \Leftrightarrow q$
T	T	T	T	F	T	T
T	F	F	T	F	F	F
F	T	F	T	T	T	F
F	F	F	F	T	T	T

In order to build the truth table of a formula we decompose the formula into sub formulae e.g.:

p	q	((p	∧	¬	q)	∨	p)	∧	¬	(p	∨	¬	q)
T	T	T	F	F	T	T	T	F	F	T	T	F	T
T	F	T	T	T	F	T	T	F	F	T	T	T	F
F	T	F	F	F	T	F	F	F	T	F	F	F	T
F	F	F	F	T	F	F	F	F	F	F	T	T	F

While this looks very complicated, just follow the logic through and it is simple to do.

4 Some basic notation

If we have a propositional formula $\varphi(x_1, x_2, ..., x_n)$ then we can call an assignment f of either **T** or **F** to each $x_1, x_2, ..., x_n$ i.e. a function

$$f : \{x_1, x_2, ..., x_n\} \rightarrow \{T, F\}$$

a **truth assignment (interpretation, valuation)** for φ

We say that φ **evaluates** to **T**(with respective to **F** under f. If the row of the truth table for φ corresponding to **f** evaluates to **T**(with respect to **F**.

for a set of inputs f, if a formula with these inputs comes to true, then f is a satisfying truth assignment of φ

φ is the formula under which the inputs f are put under, for example $p \wedge q$

If **f** evaluates φ to **T** then f **satisfies** φ or is a **satisfying truth assignment of** φ or a **model** of φ

If φ evaluates to **T** for every f then φ is a **tautology**
If φ evaluates to **F** for every f then φ is a **contradiction**

A **literal** is either a propositional variable, say **x**, or the negation of a propositional variable, say ¬x

5 Logical equivalence

Steps in a mathematical proof are often just the replacement of one statement by another (equivalent) statement which says the same thing e.g.
"If I don't explain this clearly then the students won't understand" is the same thing as
"Either I explain this clearly or the students won't understand"

To see this, denote the sub-statement "I don't explain this clearly" as **X** and
denote the sub statement "the students won't understand" as **Y**

The former statement is thus $X \Rightarrow Y$ and the latter:

X	Y	X	⇒	Y	¬	X	∨	Y
T	T	T	**T**	T	F	T	**T**	T
T	F	T	**F**	F	F	T	**F**	F
F	T	F	**T**	T	T	F	**T**	T
F	F	F	**T**	F	T	F	**T**	F

We say that the two propositional formulae are **(logically)** equivalent if they have identical truth tables
If φ and ψ are equivalent then we write $\varphi \equiv \psi$

6 A spot of practice

The **exclusive-OR** is written $X \oplus Y$ and if **true** iff exactly one of **X** and **Y** is **true**
Prove that $X \oplus Y$ *is logically equivalent to both* $(X \wedge \neg Y) \vee (\neg X \wedge Y)$ *and* $\neg(X \Leftrightarrow Y)$

X	Y	X	⊕	Y	(X	∧	¬	Y)	∨	(¬	X	∧	Y	¬	(X	⇔	Y)
T	T	T	**F**	T	T	F	F	T	**F**	F	T	F	T	**F**	T	T	T
T	F	T	**T**	F	T	T	T	F	**T**	F	T	F	F	**T**	T	F	F
F	T	F	**T**	T	F	F	F	T	**T**	T	F	T	T	**T**	F	F	T
F	F	F	**F**	F	F	F	T	F	**F**	T	F	F	F	**F**	F	T	F

7 De Morgan's Laws

These are two very useful logical equivalences known as De Morgan's Laws
De Morgan's Laws are:

- $\neg(X \wedge Y) \equiv \neg X \vee \neg Y$
- $\neg(X \vee Y) \equiv \neg X \wedge \neg Y$

These formulae are indeed equivalences:

X	Y	¬	(X	∧	Y)	¬	X	∨	¬	Y	¬	(X	∨	Y)	¬	X	∧	¬	y
T	T	**F**	T	T	T	F	T	**F**	F	T	**F**	T	T	T	F	T	**F**	F	T
T	F	**T**	T	F	F	F	T	**T**	T	F	**F**	T	T	F	F	T	**F**	T	F
F	T	**T**	F	F	T	T	F	**T**	F	T	**F**	F	T	T	T	F	**F**	F	T
F	F	**T**	F	F	F	T	F	**T**	T	F	**T**	F	F	F	T	F	**T**	T	F

- De Morgan's Laws can be applied not just to variables, but to formulae φ and ψ
- De Morgan's Laws are often used to simplify formulae with regard to negations

8 Applying De Morgan's Laws

In fact, not only can De Morgan's Laws be applied to formula, they can be applied to **sub-formulae** within a formula

Take the propositional formula:

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

and the sub formula

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

By De Morgan's Laws

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

So:

$$\neg(p \vee \neg(q \wedge \neg p)) \wedge \neg(p \Rightarrow q) \equiv \neg(p \vee (\neg q \vee p)) \wedge \neg(p \Rightarrow q)$$

Indeed, we can always replace any sub-formula of some propositional formula with an **equivalent formula** without affecting the truth(table) of the original.

9 A spot of practice

Consider $\neg(p \vee \neg(q \wedge \neg p)) \wedge \neg(p \Rightarrow q)$
Can we manipulate it so as to simplify it?

$\neg(p \vee \neg(q \wedge \neg p)) \wedge \neg(p \Rightarrow q)$	Apply De Morgan's laws
$\neg(p \vee (\neg q \vee \neg\neg p)) \wedge \neg(p \Rightarrow q)$	Remove double negation
$\neg(p \vee (\neg q \vee p)) \wedge \neg(p \Rightarrow q)$	apply De Morgan's laws
$(\neg p \wedge \neg(\neg q \wedge p)) \wedge \neg(p \Rightarrow q)$	apply De Morgan's laws
$(\neg p \wedge (\neg\neg q \wedge \neg p)) \wedge \neg(p \Rightarrow q)$	Remove double negation
$(\neg p \wedge (q \wedge \neg p)) \wedge \neg(p \Rightarrow q)$	$\Rightarrow$ using $\vee, \neg$
$(\neg p \wedge (q \wedge \neg p)) \wedge \neg(\neg p \vee q)$	apply De Morgan's Laws
$(\neg p \wedge (q \wedge \neg p)) \wedge (\neg\neg p \wedge \neg q)$	remove double negation
$(\neg p \wedge (q \wedge \neg p)) \wedge p(\wedge \neg q)$	Associativity of $\wedge$
$(\neg p \wedge q \wedge \neg p) \wedge (p \wedge \neg q$	Associativity of $\wedge$
$\neg p \wedge q \wedge \neg p \wedge p \wedge \neg q$	Commutativity of $\wedge$
$\neg p \wedge \neg p \wedge p \wedge q \wedge \neg q$	$X \wedge \neg X \equiv F$
$\neg p \wedge F \wedge q \wedge \neg q$	$F \wedge \varphi \equiv F$
F	

10 Generalised De Morgan's Laws

We can actually generalise De Morgan's laws so that negations can be "pushed inside" conjunction/disjunctions of more than two literals

To do this we apply De Morgan's laws to sub formulae of a formula

Consider $\neg(X \vee Y \vee Z)$
Rewrite this formula as $\neg(X \vee (Y \vee Z))$ and denote $Y \vee Z$ by φ
Applying De Morgan's laws to $\neg(X \vee \varphi)$

11 Some rules

$$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$$
$$(p \vee q) \vee r \equiv p \vee (q \vee r)$$
$$p \wedge \neg p \equiv F$$