Chapter 1: The Foundations: Logic and Proofs

1.1 Propositional Logic

1.1 Propositional Logic

Introduction

- A **proposition** is a declarative sentence (a sentence that declares a fact) that is either true or false, but not both.
- Are the following sentences propositions?
  - Toronto is the capital of Canada. (Yes)
  - Read this carefully. (No)
  - 1+2=3 (Yes)
  - x+1=2 (No)
  - What time is it? (No)

1.1 Propositional Logic

- Propositional Logic – the area of logic that deals with propositions
- Propositional Variables – variables that represent propositions: p, q, r, s
  - E.g. Proposition p – "Today is Friday."
- Truth values – T, F

1.1 Propositional Logic

DEFINITION 1

Let p be a proposition. The negation of p, denoted by $\neg p$, is the statement
"It is not the case that p."

The proposition $\neg p$ is read "not p." The truth value of the negation of p, $\neg p$ is the opposite of the truth value of p.

- Examples
  - Find the negation of the proposition "Today is Friday." and express this in simple English.

    Solution: The negation is "It is not the case that *today is Friday*."
    In simple English, "Today is not Friday." or "It is not Friday today."
  - Find the negation of the proposition "Michael's PC runs Linux." and express this in simple English.

    Solution: The negation is "It is not the case that Michael's PC runs Linux."
    In simple English, "Michael's PC does not run Linux."

1.1 Propositional Logic

- Note: Always assume fixed times, fixed places, and particular people unless otherwise noted.
- Truth table:

The Truth Table for the Negation of a Proposition.	
p	$\neg p$
T	F
F	T

- Logical operators are used to form new propositions from two or more existing propositions. The logical operators are also called connectives.

1.1 Propositional Logic

DEFINITION 2

Let p and q be propositions. The *conjunction* of p and q, denoted by $p \wedge q$, is the proposition "p and q". The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

- Examples
  - Find the conjunction of the propositions p and q where p is the proposition "Today is Friday." and q is the proposition "It is raining today.", and the truth value of the conjunction.

Solution: The conjunction is the proposition "Today is Friday and it is raining today." The proposition is true on rainy Fridays.

1.1 Propositional Logic

DEFINITION 3

Let p and q be propositions. The *disjunction* of p and q, denoted by $p \vee q$, is the proposition "p or q". The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

- Note:

  inclusive or : The disjunction is true when at least one of the two propositions is true.

  - E.g. "Students who have taken calculus or computer science can take this class." – those who take one or both classes.

  exclusive or : The disjunction is true only when one of the proposition is true.

  - E.g. "Students who have taken calculus or computer science, but not both, can take this class." – only those who take one of them.

- Definition 3 uses *inclusive or*.

1.1 Propositional Logic

DEFINITION 4

Let p and q be propositions. The *exclusive or* of p and q, denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.

The Truth Table for the Conjunction of Two Propositions.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

The Truth Table for the Disjunction of Two Propositions.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

The Truth Table for the Exclusive *Or* (*XOR*) of Two Propositions.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

1.1 Propositional Logic

Conditional Statements

DEFINITION 5

Let p and q be propositions. The *conditional statement* $p \rightarrow q$, is the proposition "if p, then q." The conditional statement is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the *hypothesis* (or *antecedent* or *premise*) and q is called the *conclusion* (or *consequence*).

- A conditional statement is also called an implication.
- Example: "If I am elected, then I will lower taxes." $p \rightarrow q$

implication:

	p	q	$p \rightarrow q$
elected, lower taxes.	T	T	T
not elected, lower taxes.	F	T	T
not elected, not lower taxes.	F	F	T
elected, not lower taxes.	T	F	F

1.1 Propositional Logic

- Example:
  - Let p be the statement “Maria learns discrete mathematics.” and q the statement “Maria will find a good job.” Express the statement $p \longrightarrow q$ as a statement in English.

    Solution: Any of the following -

    “If Maria learns discrete mathematics, then she will find a good job.

    “Maria will find a good job when she learns discrete mathematics.”

    “For Maria to get a good job, it is sufficient for her to learn discrete mathematics.”

    “Maria will find a good job unless she does not learn discrete mathematics.”

1.1 Propositional Logic

- Other conditional statements:
  - Converse of $p \longrightarrow q$: $q \longrightarrow p$
  - Contrapositive of $p \longrightarrow q$: $\neg q \longrightarrow \neg p$
  - Inverse of $p \longrightarrow q$: $\neg p \longrightarrow \neg q$

1.1 Propositional Logic

- Find the contrapositive, the converse, and the inverse of the conditional statement:

  "The home team wins whenever it is raining."

  - Converse "If the home team wins, then it is raining.
  - Contrapositive If the home team does not win, then it is not raining.
  - Inverse If it is not raining, then the home team does not win.

1.1 Propositional Logic

DEFINITION 6

Let p and q be propositions. The *biconditional statement* $p \leftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called *bi-implications*.

- $p \leftrightarrow q$ has the same truth value as $(p \rightarrow q) \land (q \rightarrow p)$
- "*if and only if*" can be expressed by "*iff*"
- Example:
  - Let p be the statement "You can take the flight" and let q be the statement "You buy a ticket." Then $p \leftrightarrow q$ is the statement

    "You can take the flight if and only if you buy a ticket."

    Implication:

    If you buy a ticket you can take the flight.

    If you don't buy a ticket you cannot take the flight.

1.1 Propositional Logic

The Truth Table for the Biconditional $p \leftrightarrow q$.		
p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

1.1 Propositional Logic

Truth Tables of Compound Propositions

- We can use connectives to build up complicated compound propositions involving any number of propositional variables, then use truth tables to determine the truth value of these compound propositions.
- Example: Construct the truth table of the compound proposition

$$(p \vee \neg q) \longrightarrow (p \wedge q).$$

The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

1.1 Propositional Logic

Precedence of Logical Operators

- We can use parentheses to specify the order in which logical operators in a compound proposition are to be applied.
- To reduce the number of parentheses, the precedence order is defined for logical operators.

Precedence of Logical Operators.	
Operator	Precedence
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

E.g. $\neg p \wedge q = (\neg p) \wedge q$

$p \wedge q \vee r = (p \wedge q) \vee r$

$p \vee q \wedge r = p \vee (q \wedge r)$

1.1 Propositional Logic

Logic and Bit Operations

- Computers represent information using bits.
- A **bit** is a symbol with two possible values, 0 and 1.
- By convention, 1 represents T (true) and 0 represents F (false).
- A variable is called a Boolean variable if its value is either true or false.
- Bit operation – replace true by 1 and false by 0 in logical operations.

Table for the Bit Operators *OR*, *AND*, and *XOR*.

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

1.1 Propositional Logic

DEFINITION 7
A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.

- Example: Find the bitwise *OR*, bitwise *AND*, and bitwise *XOR* of the bit string 01 1011 0110 and 11 0001 1101.

Solution:

01 1011 0110

11 0001 1101

----

11 1011 1111 bitwise *OR*

01 0001 0100 bitwise *AND*

10 1010 1011 bitwise *XOR*