Logic and Proofs

Madiba Hudson-Quansah

Contents

Chapter 1 **Propositional Logic** Page 2

1.1 Proposition 2

1.2 Logical Operators / Connectives 3

Conjunction — 3 • Disjunction — 3

1.2.2.1 Exclusive or . . . 4

Conditional Statement / Implication — 4

1.2.3.1 Contrapositive . . . 6

1.2.3.2 Inverse . . . 6

1.2.3.3 Converse . . . 6

Biconditionals / Bi-implications — 7

1.3 Compound Propositions 9

Precedence of Logical Operators — 9 • Logic and Bit Operations — 9

Chapter 2 **Applications of Propositional Logic** Page 11

Translating English Sentences — 11 • System Specifications — 12 • Boolean Searches — 13 • Logic Puzzles — 14 • Logic Circuits — 15

2.1 Exercises 15

Chapter 3 **Logical Equivalences** Page 17

3.1 Equivalence Laws 18

Logical Equivalences Involving Conditional Statements — 18 • Logical Equivalences Involving Biconditional Statements — 18 • Identity Laws — 18 • Domination Laws — 18 • Double negation Law — 18 • Idempotent Laws — 18 • Commutative Laws — 19 • Associative Laws — 19 • Distributive Laws — 19 • De Morgan's Laws — 19 • Absorption Laws — 19 • Negation Laws — 19

3.2 Exercises 19

Chapter 4 **Predicates and Quantifiers** Page 22

4.1 Propositional Function 22

Preconditions and Postconditions — 24

4.2 Quantifiers 25

Universal Quantifier — 26 • Existential Quantifier — 27 • Uniqueness Quantifier — 27 • Quantifiers with Restricted Domains — 27 • De Morgan's Laws for Quantifiers — 28 • Negating Quantifications — 28

4.3 Nested Quantifiers 29

Negating Nested Quantifiers — 29

Chapter 1

Propositional Logic

Definition 1.0.1

- Proof - A correct mathematical argument.
- Theorem - A proven mathematical statement.

1.1 Proposition

Definition 1.1.1: A Proposition

A declarative sentence that is either true or false, but not both. e.g.

- $1 + 1 = 2$ True
- $2 + 2 = 3$ False

Propositional variables / Statement variables are used to represent propositions, by convention one of these variables $p, q, r, s \ldots$ The truth value of a position can be denoted by T if it is a **true proposition** and F if it is a **false proposition**.

Therefore, Let p be a proposition. The *negation of* p, denoted by $\neg p$ / $\overline{p}$, is the statement

"It is not the case that p"

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

Example 1.1.1

Question 1

Find the negation of the proposition
"Michael's PC runs Linux"
and express this in simple English.

Solution: "Michael's PC does not run Linux"

Example 1.1.2

Question 2

Find the negation of the proposition
"Vandana's smartphone has at least 32GB of memory"
and express this in simple English.

Solution: "Vandana's has less than 32GB of memory"

Definition 1.1.2: Truth Table

Displays the relationships between the truth values of propositions.

p	$\neg p$
T	F
F	T

Table 1.1: The truth table for the negation of a proposition

The negation of a proposition can also be considered the result of the operation of the *negation operator* on the proposition.

1.2 Logical Operators / Connectives

1.2.1 Conjunction

Definition 1.2.1: Conjunction

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

Table 1.2: The truth table of $p \wedge q$

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

1.2.2 Disjunction

Definition 1.2.2: Disjunction

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

The use of the **connective** *or* in a disjunction corresponds to one of the two ways the word *or* is used in English. **Inclusive or** and **Exclusive or**, e.g.

"Students who have taken calculus or computer science can take this class"

"Students who have taken calculus or computer science, but not both can take this class"

Table 1.3: The truth table of $p \vee q$

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

Respectively. Therefore taking the disjunction $p \vee q$ an **Exclusive or** disjunction will F when $q = T$ and $p = T$ or $q = F$ and $p = F$, and T only when $q = T$ and $p = F$ or $q = F$ and $p = T$

1.2.2.1 Exclusive or

Definition 1.2.3: Exclusive Or

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

Table 1.4: The truth table of $p \oplus q$

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

1.2.3 Conditional Statement / Implication

Definition 1.2.4: Conditional Statement / Implication

Let p and q be propositions. The *conditional statement* $p \longrightarrow q$ is the proposition "if p, then q". $p \longrightarrow q$ is F when p is T and q is F, and T otherwise. In this connective, p is called the *hypothesis / antecedent / premise* and q is called the *conclusion / consequence*.

Table 1.5: The truth table of $p \longrightarrow q$

p	q	$p \longrightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

$p \longrightarrow q$ is called a conditional statement because, it asserts that q is T on the condition that p holds. $p \longrightarrow q$ is T when p is F no matter the value of q.

Conditional statements can be expressed in various ways, some are listed below.

"if p, then q"

"if p, q"

"p is sufficient for q"

"q if p"

"q when p"

”a necessary condition for p is q”

”q unless $\neg p$”

”p implies q”

”p only if q”

”a sufficient condition for q is p”

”q whenever p”

”q is necessary for p”

”q follows from p”

For the more confusing statements ”p only if q” and ”q unless $\neg p$”, the explanation follows.

”p only if q” corresponds to ”if p, then q”, because ”p only if q” says that p cannot be T when q is not T, i.e. the statement is F when is p is T but q is F. If p is F q maybe either F or T because the statement says nothing about the value of q.

”q unless $\neg p$” expresses the same conditional statement as ”if p, then q”, because ”q unless $\neg p$” means that if ”if $\neg p$” is F then q must be T, That is the statement ”q unless $\neg p$” is F when p is T but q is F, but T otherwise.

Or

$$\begin{aligned} q \text{ unless } \neg p \\ q \text{ if } \neg\neg p \\ q \text{ if } p \\ \therefore \text{ if } p \text{ then } q \end{aligned}$$

Example 1.2.1

Question 3

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

Solution:

”Maria will find a good job, if 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”

We can form new conditional statements from a given conditional statement, lets say $p \to q$. These are

- **Converse -** $q \to p$
- **Contrapositive -** $\neg q \to \neg p$
- **Inverse -** $\neg p \to \neg q$

Definition 1.2.5: Equivalence

When two compound propositions always have the same truth value.

1.2.3.1 Contrapositive

Definition 1.2.6: Contrapositive

The *contrapositive* of the conditional statement $p \rightarrow q$ is the conditional statement $\neg q \rightarrow \neg p$

In this statement, the hypothesis ,p, and conclusion , q, are reversed and negated. This results in an identical truth table, as the contrapositive is only F when $\neg p$ is F and $\neg q$ is T.

Note:-

A **Conditional Statement** ($p \rightarrow q$) is equivalent to it's **Contrapositive** ($\neg q \rightarrow \neg p$)

Table 1.6: The truth table of $\neg q \rightarrow \neg p$

p	q	$\neg p$	$\neg q$	$\neg q \rightarrow \neg p$
T	T	F	F	T
T	F	F	T	F
F	T	T	F	T
F	F	T	T	T

1.2.3.2 Inverse

Definition 1.2.7: Inverse

The inverse of the conditional statement $p \rightarrow q$ is the conditional statement $\neg p \rightarrow \neg q$

In this statement, the hypothesis ,p, and conclusion ,q, are negated. This results in a truth table differing from the original conditional statement but equivalent to the statement's **converse**.

Note:-

A conditional statement's ($p \rightarrow q$) **Inverse** ($\neg p \rightarrow \neg q$) is equivalent to its **Converse** ($q \rightarrow p$)

Table 1.7: The truth table of $\neg p \rightarrow \neg q$

p	q	$\neg p$	$\neg q$	$\neg p \rightarrow \neg q$
T	T	F	F	T
T	F	F	T	T
F	T	T	F	F
F	F	T	T	T

1.2.3.3 Converse

Definition 1.2.8: Converse

The converse of the conditional statement $p \rightarrow q$ is the conditional statement $q \rightarrow p$

In this statement, the hypothesis , p and conclusion, , q, are reversed. This results in a truth table equivalent to the conditional statement's inverse.

Note:-

A conditional statement's ($p \rightarrow q$) **Converse** ($q \rightarrow p$) is equivalent to its **Inverse** ($\neg p \rightarrow \neg q$)

Table 1.8: The truth table of $q \to p$

p	q	$q \to p$
T	T	T
T	F	T
F	T	F
F	F	T

Example 1.2.2

Question 4

What are the contrapositive, converse, and inverse of the conditional statement
"The home team wins whenever it is raining?"

Solution:

$$\begin{aligned} p &\to q \\ q &\text{ whenever } p \\ q &= \text{Home team wins} \\ p &= \text{It is raining} \\ &= \text{If it is raining, the home team wins} \end{aligned}$$

- Contrapositive - If the home team loses, then it's not raining.
- Inverse - If its not raining then the home team loses.
- Converse - If the home team wins, then it is raining.

Example 1.2.3

Question 5

Find the converse, inverse and contrapositive of "Raining is a sufficient condition for my not going to town"

Solution:

If it is raining, then I will not go to town
$p \to q$
p = It is raining
q = I will not go to town
Converse
$q \to p$
If I am not going to town, then it is raining
Contrapositive
$\neg q \to \neg p$
If I am going to town, then it is not raining
Inverse
$\neg p \to \neg q$
If it is not raining, then I am going to town

1.2.4 Biconditionals / Bi-implications

Another way to combine proposition that expresses they have the same truth value.

Definition 1.2.9: Biconditionals / Bi-implications

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 T when p and q have the same truth values, and is F otherwise.

$p \leftrightarrow q$ breaks down to $(p \rightarrow q) \wedge (q \rightarrow p)$, and can be expressed as below

"p is necessary and sufficient for q"

"if p then q, and conversely"

"p iff q"

Note:-

"iff" - If and only If

Table 1.9: The truth table of $p \leftrightarrow q$

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Example 1.2.4

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"

1.3 Compound Propositions

Question 6

Construct the truth table of the compound proposition

$$(p \lor \neg q) \rightarrow (p \land q)$$

Solution:

p	q	$\neg q$	$p \lor \neg q$	$p \land q$	$(p \lor \neg q) \rightarrow (p \land q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Question 7

Construct

$$p \lor q \rightarrow \neg r$$

Solution:

p	q	r	$\neg r$	$p \lor q$	$p \lor q \rightarrow \neg r$
T	T	T	F	T	F
T	T	F	T	T	T
T	F	F	T	T	T
T	F	T	F	T	F
F	F	F	T	F	T
F	T	T	F	T	F
F	F	T	F	F	T
F	T	F	T	T	T

1.3.1 Precedence of Logical Operators

Table 1.10: Precedence Table

Operator	Precedence
$\neg$	1
$\land$	2
$\lor$	3
$\rightarrow$	4
$\leftrightarrow$	5

Precedence shown from 1 to 5, with 1 having the highest precedence and 5 having the lowest precedence. Operators with higher precedence are evaluated before operators with lower precedence. Precedence can be overridden by using parentheses.

1.3.2 Logic and Bit Operations

A bit can be used to represent a truth value due to its *binary* nature. A variable representing a truth value can be called a *boolean variable*. Computer bit operations correspond to logical operations, with the operations *OR*, *AND*, and *XOR* corresponding to the connectives, $\lor$, $\land$, and $\oplus$ respectively.

Table 1.11: Truth value to bit table

Truth value	Bit
T	1
F	0

Table 1.12: Bit operations table

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

Definition 1.3.1: Bit String

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

Example 1.3.1

101010011 is a bit string with a length of 9.

Extending bit operations to bit strings we can define *bitwise AND*, *bitwise OR*, and *bitwise XOR* of two strings of the same length. The new bit string created can be called the *AND*, *OR*, and *XOR* of the two strings respectively.

Example 1.3.2

Question 8

Find the bitwise AND, OR, and XOR of the bit strings 01 1011 0110 and 11 0001 1101.

Solution:

AND

$$\begin{array}{c} 0110110110 \\ \underline{1100011101} \\ 0100010100 \end{array}$$

OR

$$\begin{array}{c} 0110110110 \\ \underline{1100011101} \\ 1110111111 \end{array}$$

XOR

$$\begin{array}{c} 0110110110 \\ \underline{1100011101} \\ 1010101011 \end{array}$$

Chapter 2

Applications of Propositional Logic

Statements in natural language are often imprecise and ambiguous, to make these statements more precise they can be represented with propositional logic.

2.0.1 Translating English Sentences

Example 2.0.1

Question 9

How can this English sentence be translated into a logical expression?
"You can access the Internet from campus only if you are a computer science major or you are not a freshman".

Solution:

Let p be "You can access the internet from campus"
Let q be "You are a computer science major"
Let r be "You are a freshman"
Therefore the sentence can be translated as

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

Example 2.0.2

Question 10

How can this English sentence be translated into a logical expression?
"You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old."

Solution:

Let p be "You can ride the roller coaster"
Let q be "You are under 4 feet tall"
Let r "You are at least 16 years old"
Therefore the sentence can be translated as

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

2.0.2 System Specifications

In system specifications, we can use propositional logic to express requirements given in natural language in precise and unambiguous specifications that can serve as the basis for system development.

Example 2.0.3

Question 11

Express the specification ”The automated reply cannot be sent when the file system is full” using logical connectives

Solution:

Let p be ”The automated reply can be sent”
Let q be ”The file system is full”
Therefore the specification can be expressed as

$$q \rightarrow \neg p$$

System specifications should be consistent, and therefore should not contain contradictory requirements.

Question 12

Determine whether these system specifications are consistent:
“The diagnostic message is stored in the buffer or it is retransmitted.”
“The diagnostic message is not stored in the buffer.”
“If the diagnostic message is stored in the buffer, then it is retransmitted.”

Solution:

Let p be ”The diagnostic message is stored in the buffer”

Let q be ”The diagnostic message is retransmitted”

Therefore the specifications can be expressed as:

$$\begin{aligned} p \lor q \\ \neg p \\ p \to q \end{aligned}$$

To have all these specifications be consistent, it must be possible for them to all be T at the same time. In this case making $p = T$ and $q = T$ will result in $\neg p$ being F, which would cause the only the first and second specifications to pass. But making $p = F$ and $q = T$ would result in all the specifications being T, as proven by the truth table below.

p	q	$p \lor q$	$\neg p$	$p \to q$	$(p \lor q) \land \neg p \land (p \to q)$
T	T	T	F	T	F
T	F	T	F	F	F
F	T	T	T	T	T
F	F	F	T	T	F

Question 13

Do the system specifications in the previous example remain consistent if the specification “The diagnostic message is not retransmitted” is added?

Solution: With the addition of the specification represented logically by $\neg q$ it would make it impossible for all the specifications to evaluate to T, therefore the specifications would be inconsistent.

2.0.3 Boolean Searches

Boolean searches are used to search for information in large collections, such as web page indexes and databases. These searches usually use the connectives *AND*, *OR*, and *NOT / AND NOT*.

2.0.4 Logic Puzzles

Puzzles that can be solved using logical reasoning.

Question 14

On an island with two types of people, knights and knaves, knights always tell the truth and knaves always lie. You encounter two people A and B. What are A and B if A says "B is a knight" and B says "The two of us are opposite types"?

Solution:

Let r be "A is a knight"

Let q be "B is a knight"

Let $\neg r$ be "A is a knave"

Let $\neg q$ be "B is a knave"

First we consider the case where $r = T$. In this case A is a knight making everything he says true meaning B is a knight too. But in B being a knight everything he says is also true but that cannot be as in this scenario both A and B are knights, making this scenario invalid.

Next we consider the case where $\neg r = T$. In this case A is a knave meaning he is lying about B being a knight. Following this we can conclude that B is also a knave and everything they also say is a lie, which is valid.

Next we consider the case where $q = T$. In this case B is a knight making everything he says true and thus A is a knave. But as A is a knave everything they say is false which calls B's knighthood into question, rendering this scenario invalid.

Finally we consider the case where $\neg q = T$. This is case B is a knave making everything they say false. This allows us to conclude that A and B are not of opposite types, and since A is believed to be a knave further supports B being a knave, making this scenario valid.

Concluding we see in the scenario's we've gone through there are only two valid paths and both point to both A and B being knaves.

Question 15

A father tells his two children, a boy and a girl, to play in their backyard without getting dirty. However, while playing, both children get mud on their foreheads. When the children stop playing, the father says "At least one of you has a muddy forehead," and then asks the children to answer "Yes" or "No" to the question: "Do you know whether you have a muddy forehead?" The father asks this question twice. What will the children answer each time this question is asked, assuming that a child can see whether his or her sibling has a muddy forehead, but cannot see his or her own forehead? Assume that both children are honest and that the children answer each question simultaneously.

Solution:

Let q be "boy has muddy forehead"

Let r be "girl has muddy forehead"

The first time the question is asked both children answer "No" because as they can see each other's foreheads they can tell if the other has a muddy forehead but cannot tell if they do, the boy and girl know $r = T$ but not the vale of q and $q = T$ but not the value of r.

The second time the question is asked taking account of the answer of either sibling for the first question and the condition that at least one of them has a muddy forehead the children can infer that since they can see mud on their sibling's forehead but they answered no but for statement the father made which can be put as $q \vee r$ to be true at least one of the propositions have to be true, resoling in them both answering "Yes".

2.0.5 Logic Circuits

A logic circuit receives input signals $p_1, p_2 \ldots p_n$, each a bit, either 1 or 0, and produces output signals $s_1, s_2 \ldots s_n$, each also a bit.

Complicated digital circuits can be constructed from three basic circuits, called **gates**. These are:

- Inverter / NOT gate - Takes an input bit p and produces $\neg p$ as output.
- OR gate - Takes two input signals p and q and produces the output signal $p \vee q$.
- AND gate - Takes two input signals p and q and produces the output signal $p \wedge q$.

Given a circuit built from these gates we can determine the output by tracing through the circuit.

2.1 Exercises

Translate the given statements into propositional logic

Question 16

You cannot edit a protected Wikipedia entry unless you are an administrator. Express your answer in terms of e: "You can edit a protected Wikipedia entry" and a: "You are an administrator."

Solution:

$e \longrightarrow a$

Question 17

You can see the movie only if you are over 18 years old or you have the permission of a parent. Express your answer in terms of m: "You can see the movie," e: "You are over 18 years old," and p: "You have the permission of a parent."

Solution: $m \longrightarrow (e \vee p)$

Question 18

You can graduate only if you have completed the requirements of your major and you do not owe money to the university and you do not have an overdue library book. Express your answer in terms of g: "You can graduate," m: "You owe money to the university," r: "You have completed the requirements of your major," and b: "You have an overdue library book."

Solution: $g \longrightarrow (r \wedge \neg m \wedge \neg b)$

Question 19

To use the wireless network in the airport you must pay the daily fee unless you are a subscriber to the service. Express your answer in terms of w: "You can use the wire- less network in the airport," d: "You pay the daily fee," and s: "You are a subscriber to the service."

Solution: $w \longrightarrow (d \vee s)$

Question 20

You can upgrade your operating system only if you have a 32-bit processor running at 1 GHz or faster, at least 1 GB RAM, and 16 GB free hard disk space, or a 64- bit processor running at 2 GHz or faster, at least 2 GB RAM, and at least 32 GB free hard disk space. Express you answer in terms of u: “You can upgrade your operating system,” b32: “You have a 32-bit processor,” b64: “You have a 64-bit processor,” g1: “Your processor runs at 1 GHz or faster,” g2: “Your processor runs at 2 GHz or faster,” r1: “Your processor has at least 1 GB RAM,” r2: “Your processor has at least 2 GB RAM,” h16: “You have at least 16 GB free hard disk space,” and h32: “You have at least 32 GB free hard disk space.”

Solution: $u \longrightarrow (b_{32} \land g_1 \land r_1 \land h_{16}) \lor (b_{64} \land g_2 \land r_2 \land h_{32})$

Question 21

Constuct a truch table for the compound proposition below

$$(p \longrightarrow q) \lor (\neg p \longrightarrow r)$$

Solution:

p	q	r	$\neg r$	$p \longrightarrow q$	$\neg p \longrightarrow r$	$(p \longrightarrow q) \lor (\neg p \longrightarrow r)$

Chapter 3

Logical Equivalences

Definition 3.0.1: Tautology

A proposition that is always T. e.g. $p \vee \neg p$

Definition 3.0.2: Contradiction

A proposition that is always F e.g. $p \wedge \neg p$

Definition 3.0.3: Contingency

A proposition that is neither a tautology or a contradiction. e.g.. p

Table 3.1: Demonstration

Two compound propositions p and q are logically equivalent if $p \leftrightarrow q$, where q and p are compound propositions.

3.1 Equivalence Laws

3.1.1 Logical Equivalences Involving Conditional Statements

$$\begin{aligned}
p \to q &\equiv \neg p \lor q \\
p \to q &\equiv \neg q \to \neg p \\
p \lor q &\equiv \neg p \to q \\
p \land q &\equiv \neg \left(p \to \neg q\right) \\
\neg \left(p \to q\right) &\equiv p \land \neg q \\
\left(p \to q\right) \land \left(p \to r\right) &\equiv p \to \left(q \land r\right) \\
\left(p \to r\right) \land \left(q \to r\right) &\equiv \left(p \lor q\right) \to r \\
\left(p \to q\right) \lor \left(p \to r\right) &\equiv p \to \left(q \lor r\right) \\
\left(p \to r\right) \lor \left(q \to r\right) &\equiv \left(p \land q\right) \to r
\end{aligned}$$

3.1.2 Logical Equivalences Involving Biconditional Statements

$$\begin{aligned}
p \leftrightarrow q &\equiv \left(p \to q\right) \land \left(q \to p\right) \\
p \leftrightarrow q &\equiv \neg p \leftrightarrow \neg q \\
p \leftrightarrow q &\equiv \left(p \land q\right) \lor \left(\neg p \land \neg q\right) \\
\neg \left(p \leftrightarrow q\right) &\equiv p \leftrightarrow \neg q
\end{aligned}$$

3.1.3 Identity Laws

$$\begin{aligned}
p \land T &\equiv p \\
p \lor F &\equiv p
\end{aligned}$$

3.1.4 Domination Laws

$$\begin{aligned}
p \lor T &\equiv T \\
p \land F &\equiv F
\end{aligned}$$

3.1.5 Double negation Law

$$\neg \left(\neg p\right) \equiv p$$

3.1.6 Idempotent Laws

$$\begin{aligned}
p \lor p &\equiv p \\
p \land p &\equiv p
\end{aligned}$$

3.1.7 Commutative Laws

$$
\begin{aligned}
p \lor q &\equiv q \lor p \\
p \land q &\equiv q \land p
\end{aligned}
$$

3.1.8 Associative Laws

$$
\begin{aligned}
(p \lor q) \lor r &\equiv p \lor (q \lor r) \\
(p \land q) \land r &\equiv p \land (q \land r)
\end{aligned}
$$

3.1.9 Distributive Laws

$$
\begin{aligned}
p \lor (q \land r) &\equiv (p \lor q) \land (p \lor r) \\
p \land (q \lor r) &\equiv (p \land q) \lor (p \land r)
\end{aligned}
$$

3.1.10 De Morgan's Laws

$$
\begin{aligned}
\neg (p \land q) &\equiv \neg p \lor \neg q \\
\neg (p \lor q) &\equiv \neg p \land \neg q
\end{aligned}
$$

3.1.11 Absorption Laws

$$
\begin{aligned}
p \lor (p \land q) &\equiv p \\
p \land (p \lor q) &\equiv p
\end{aligned}
$$

3.1.12 Negation Laws

$$
\begin{aligned}
p \lor \neg p &\equiv T \\
p \land \neg p &\equiv F
\end{aligned}
$$

3.2 Exercises

Example 3.2.1

Question 22

Show that $\neg (p \lor (\neg p \land q))$ is logically equivalent to $\neg p \land \neg q$

Solution:

$$\begin{aligned}
\neg\left(p \vee \left(\neg p \wedge q\right)\right) &= \neg p \wedge \neg\left(\neg p \wedge q\right) && \text{By the Second De Morgan's Law}\\
&= \neg p \wedge \left(\neg\left(\neg p\right) \vee \neg q\right) && \text{By First De Morgan's Law}\\
&= \neg p \wedge \left(p \vee \neg q\right) && \text{By Double Negation Law}\\
&= \left(\neg p \wedge p\right) \vee \left(\neg p \wedge \neg q\right) && \text{By Second Distributive Law}\\
&= F \vee \left(\neg p \wedge \neg q\right) && \text{By Second Negation Law}\\
&= \left(\neg p \wedge \neg q\right) \vee F && \text{By First Commutative Law}\\
&= \neg p \wedge \neg q && \text{By Second Identity Law}
\end{aligned}$$

Question 23

Show that

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

is a tautology

Solution:

$$\begin{aligned}
(p \wedge q) \longrightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) && \text{Conditional Disjunction Equivalence}\\
&\equiv (\neg p \vee \neg q) \vee (p \vee q) && \text{By the First De Morgan's law}\\
&\equiv \left(\neg p \vee p\right) \vee \left(\neg q \vee q\right) && \text{By the First Associative law}\\
&\equiv \left(p \vee \neg p\right) \vee \left(q \vee \neg q\right) && \text{By the First Commutative law}\\
&\equiv \left(p \vee \neg p\right) \vee T && \text{By the First Negation Law}\\
&\equiv T \vee T && \text{By the First Negation Law}\\
&\equiv T && \text{By the First Domination Law}
\end{aligned}$$

Question 24

Use logical equivalences to prove that each of these conitional statements is a tautology

$$\left(p \wedge q\right) \longrightarrow p \tag{3.1}$$

$$\neg p \longrightarrow \left(p \longrightarrow q\right) \tag{3.2}$$

$$[\neg p \vee \left(p \vee q\right)] \longrightarrow q \tag{3.3}$$

$$[p \wedge \left(p \longrightarrow q\right)] \longrightarrow q \tag{3.4}$$

Solution:

1.

2.

$$\begin{aligned}
\neg p \to \left(p \to q\right) &\equiv \neg\left(\neg p\right) \vee \left(p \to q\right) && \text{Conditional Disjunction Equivalence} \\
&\equiv p \vee \left(p \to q\right) && \text{By the Double Negation law} \\
&\equiv p \vee (\neg p \vee q) && \text{Conditional Disjunction Equivalence} \\
&\equiv q \vee (\neg p \vee p) && \text{By the First Associative Law} \\
&\equiv q \vee \left(p \vee \neg p\right) && \text{By the First Commutative Law} \\
&\equiv q \vee T && \text{By the First Negation Law} \\
&\equiv T && \text{By the First Domination Law}
\end{aligned}$$

3.

4.

$$\begin{aligned}
[p \wedge \left(p \to q\right)] \to q &\equiv [p \wedge \left(\neg p \vee q\right)] \to q && \text{Conditional Disjunction Equivalence} \\
&\equiv \neg[p \wedge \left(\neg p \vee q\right)] \vee q && \text{Conditional Disjunction Equivalence} \\
&\equiv [\neg p \vee \neg\left(\neg p \vee q\right)] \vee q && \text{By the First De Morgan's Law} \\
&\equiv [\neg p \vee \left(\neg\left(\neg p\right) \wedge \neg q\right)] \vee q && \text{By the Second De Morgan's Law} \\
&\equiv [\neg p \vee \left(p \wedge \neg q\right)] \vee q && \text{By the Double Negation Law} \\
&\equiv [\left(\neg p \vee p\right) \wedge \left(\neg p \vee \neg q\right)] \vee q && \text{By the First Distributive Law} \\
&\equiv \left(p \vee \neg p\right) \wedge \left(\neg p \vee \neg q\right) \vee q && \text{By the First Commutative Law} \\
&\equiv T \wedge \left(\neg p \vee \neg q\right) \vee q && \text{By the First Negation Law} \\
&\equiv T \wedge (q \vee \neg q) \vee p
\end{aligned}$$

By the First Associative Law and the First Commutative Law

$$\begin{aligned}
&\equiv T \wedge T \vee p && \text{By the First Negation Law} \\
&\equiv T \vee p && \text{By the First Identity Law} \\
&\equiv T && \text{By the First Domination Law}
\end{aligned}$$

Chapter 4

Predicates and Quantifiers

Predicate logic uses the following features:

- Variables
- Predicates
- Quantifiers

4.1 Propositional Function

Definition 4.1.1: Propositional Function

Generalization of propositions / A proposition containing variables

A propositional function is made up of three components:

- Variables
- Subject
- Predicate

Definition 4.1.2: Subject

The main variable in the function

Definition 4.1.3: Predicate

Describes the property or properties of a subject

Example 4.1.1

Question 25

$$x \text{ is friends with } y$$

Solution: In this example the predicate is "is friends with", and the variables are x and y where x is the subject. This function can be denoted by:

$$Q(x, y)$$

Where Q is the predicate "is friends with".

Propositional function become propositions when their variables are each replaced by a value from the domain (U) or bound by a quantifier.

The statement $P(x)$ is said to be the value of the propositional function P at x

Example 4.1.2

Question 26

Let $x + y = z$ be denoted by $R(x, y, z)$ and U (for all three variables) be integers. Find the truth value of

$$R(2, -1, 5)$$

Solution:

$$\begin{aligned} R(2, -1, 5) \\ x = 2 \\ y = -1 \\ z = 5 \\ 2 + -1 \neq 5 \\ \therefore R(2, -1, 5) = F \end{aligned}$$

Example 4.1.3

Question 27

If $P(x)$ denotes $x \geq 0$, find this truth value

$$P(3) \longrightarrow \neg P(-1)$$

Solution:

$$\begin{aligned} P(3) &= 3 > 0 \\ P(3) &= T \\ \\ P(-1) &= -1 > 0 \\ P(-1) &= F \\ \\ \therefore T &\rightarrow \neg F \\ &= T \rightarrow T \\ &= T \end{aligned}$$

Example 4.1.4

Question 28

Let $A\left(c, n\right)$ denote the statement "Computer c is connected to network n," where c is a variable representing a computer and n is a variable representing a network. Suppose that the computer MATH1 is connected to network CAMPUS2, but not to network CAMPUS1. What are the values of $A\left(\text{MATH1}, \text{CAMPUS1}\right)$ and $A\left(\text{MATH1}, \text{CAMPUS2}\right)$?

Solution:

$$\begin{aligned} A\left(\text{MATH1}, \text{CAMPUS1}\right) &= \text{Computer MATH1 is connected to network CAMPUS1} \\ A\left(\text{MATH1}, \text{CAMPUS1}\right) &= F \\ \\ A\left(\text{MATH1}, \text{CAMPUS2}\right) &= \text{Computer MATH1 is connected to network CAMPUS2} \\ A\left(\text{MATH1}, \text{CAMPUS2}\right) &= T \end{aligned}$$

4.1.1 Preconditions and Postconditions

Predicates are also used to establish the correctness of computer programs, i.e. Verifying that programs always produce the desired output when given a valid input.

Definition 4.1.4: Precondition

Statements that describe valid input for a computer program

Definition 4.1.5: Postcondition

Conditions the output of a computer program should satisfy when run.

Example 4.1.5

Question 29

Consider this program designed to swap the values of two variables x and y.

```
temp := x;
x := y;
y := temp;
```

Find predicates that we can use as the precondition and the postcondition to verify the correctness of this program. Then explain how to use them to verify that for all valid input the program does what is intended.

Solution:

Let $P\left(x, y\right)$ be the statement $x = a$ and $y = b$, where a and b are the values of x and y before the program is run.

Let $Q\left(x, y\right)$ be the statement $x = b$ and $y = a$

To verify this program we need to show that $P\left(x, y\right) \longrightarrow Q\left(x, y\right)$ is a tautology.

We can prove this by following the program's execution. First we assume $P\left(x, y\right) = T$. In the first step of the program temp := x assigns the value of x to temp, so at this point the state of the program is $x := a,$ temp $:= a, y := b$. In the second step of the program $x := y$ assigns the value of y to x, so the state is now $x := b, y := b,$ temp $:= a$. And the final step of the program $y :=$ temp assigns the value of temp to y, so the final state is $x := b, y := a,$ temp $:= a$, making $Q\left(x, y\right) = T$

4.2 Quantifiers

Definition 4.2.1: Quantifiers

Used to express the meaning of English words such as ”all”, ”some”, ”none”, ”there exists”, ”for one and only one”, etc.

Definition 4.2.2: Domain of Discourse / Universe of Discourse / Domain

The range of all values a variable can take. Denoted as U. Without a specified domain we cannot determine the truth value of a Propositional Function.

4.2.1 Universal Quantifier

Definition 4.2.3: Universal Quantifier

”For all”: $\forall$
Meaning for every variable in the domain (U), the propositional function is true.

Example 4.2.1

$$\forall x P(x)$$

Asserts $P(x)$ is true for every x in U
So if $P(x) : x$ has stripes, and x is a zebra then $\forall x P(x)$ means ”Every zebra has stripes” or ”All zebras have stripes”

The Universal Quantifier is equivalent to the conjunction of the propositional function for all values in its specified domain. i.e. Where $P(x)$ is the statement $x > 0$ and U is all positive real numbers

$$\forall x P(x) \equiv P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge \cdots \wedge P(x_n)$$

Example 4.2.2

1. If $P(x)$ denotes $x > 0$ and U is the integers then, $\forall x P(x)$ is false.
2. If $P(x)$ denotes $x > 0$ and U is the positive integers, then $\forall x P(x)$ is true

Example 4.2.3

Question 30

What is the truth value if $\forall x P(x)$, where $P(x)$ is the statement $x^2 < 10$ and the domain consists of the positive integers not exceeding 4.

Solution:

$$\forall x P(x) \equiv P(1) \wedge P(2) \wedge P(3) \wedge P(4)$$

$$P(4) = (4)^2 < 10$$
$$= 16 < 10$$
$$\therefore P(4) = F$$
$$\therefore \forall x P(x) = F$$

4.2.2 Existential Quantifier

Definition 4.2.4: Existential Quantifier

"There exists": $\exists$
Meaning for some variables in U, the propositional function is true.

Example 4.2.4

$$\exists x P(x)$$

Asserts $P(x)$ is true for some x in U

The Existential Quantifier is equivalent to the disjunction of the propositional function for all values in its specified domain. i.e. Where $P(x)$ is the statement $x > 5$ and U is all positive real numbers

$$\exists x P(x) \equiv P(x_1) \vee P(x_2) \vee P(x_3) \vee \ldots P(x_n)$$

Example 4.2.5

1. If $P(x)$ denotes $x > 0$ and U is the integers, then $\exists x P(x)$ is true. It is also true if U is the positive integers
2. If $P(x)$ denotes $x < 0$ is the positive integers, then $\exists x P(x)$ is false

4.2.3 Uniqueness Quantifier

Definition 4.2.5: Uniqueness Quantifier

"There exists one and only one": $\exists!$

Example 4.2.6

$$\exists! P(x)$$

Asserts $P(x)$ is only true for one x in U

4.2.4 Quantifiers with Restricted Domains

The notation for a restricted domain is in the form:

QUANTIFIER VARIABLE "CONDITION A VARIABLE MUST SATISFY"

Example 4.2.7

1. $\forall x < 0 \left(x^2 > 0\right)$
2. $\forall y \neq 0 \left(y^3 \neq 0\right)$
3. $\exists z > 0 \left(z^2 = 2\right)$

Means:

1. The square of all negative integers is positive
2. The cube of every nonzero number is nonzero.

4.2.5 De Morgan's Laws for Quantifiers

4.2.6 Negating Quantifications

Consider $\forall x F(x)$, where $F(x)$ is " x has taken a course in Java " and U is students in your class.

The negation of this follows

$$\neg \forall x F(x) \equiv \exists x \neg F(x) \qquad \text{By De Morgan's Law}$$

Example 4.2.8

Question 31

Let $R(x)$ be x is a rabbit
Let $H(x)$ be x hops
Let U consists of all animals

1. $\forall x (R(x) \longrightarrow H(x))$
2. $\forall (R(x) \land H(x))$
3. $\exists x (R(x) \longrightarrow H(x))$
4. $\exists x (R(x) \land H(x))$

Solution:

1. For all animals, if they are a rabbit then they hop.
2. All animals are rabbits and they hop.
3. There exists some animals, if they are a rabbit then they hop.
4. There exists some animals that are rabbits and hop.

4.3 Nested Quantifiers

Definition 4.3.1: Nested Quantifier

A situation where one quantifier is within the scope of another.

$$\forall x \exists y \left(x + y = 0\right)$$

$\left(x + y = 0\right)$ can be made the propositional function $P\left(x, y\right)$ giving us

$$\forall x \exists y P\left(x, y\right)$$

We can then make $\exists y P\left(x, y\right)$ another propositional function $Q\left(x, y\right)$ giving us

$$\forall x Q\left(x, y\right)$$

This statement can be written as For all x there exists some y where $x + y = 0$

4.3.1 Negating Nested Quantifiers

$$\neg \forall x \exists y P\left(x, y\right) \equiv \exists x \forall y \neg P\left(x, y\right) \qquad \text{By De Morgan's Law}$$