OCR Computer Science AS Level

1.4.3 Boolean Algebra

Concise Notes

Specification:

1.4.3 a)

- Define problems using Boolean logic

1.4.3 b)

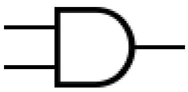
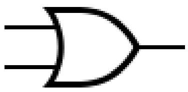
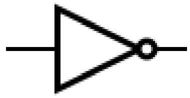
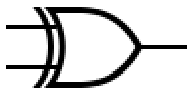
- Manipulate Boolean expressions
  - Karnaugh maps to simplify Boolean expressions

1.4.3 c)

- Use logic gate diagrams and truth tables

Logic Gate Diagrams and Truth Tables

- Problems can be defined using Boolean logic in Boolean equations
- A Boolean equation can equate to either True or False
- Four operations are used:

Operation	Conjunction	Disjunction	Negation	Exclusive Disjunction
Logic gate				
	AND	OR	NOT	XOR
Symbol	$\wedge$	$\vee$	$\neg$	$\veebar$

Truth tables

- A table showing every possible permutation of inputs to a logic gate and the corresponding output
- Inputs are usually labeled A, B, C etc
- 1 represents True, 0 represents False

Conjunction (AND)

- Applied to two literals (or inputs) to produce a single output
- Can be thought of as applying multiplication to its inputs
- Truth table shows $A \wedge B = Y$

AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Disjunction (OR)

- Operates on two literals and produces a single output
- Can be thought of as applying addition to its inputs
- As long as one input is True then the output is True
- Truth table shows $A \vee B = Y$

OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Negation (NOT)

- Only applied to one literal
- Reverses the truth value of the input
- Truth table shows $\neg A = Y$

NOT

A	Y
0	1
1	0

Exclusive Disjunction (XOR)

- Also known as exclusive OR
- Similar to disjunction but differs when both inputs are True
- Only outputs True when exactly one input is True
- Otherwise output is False
- Truth table shows $A \veebar B = Y$

XOR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Combining Boolean Operations

- Boolean equations are made by combining Boolean operators
- This is done in the same way that standard mathematical operators are combined
- Every boolean equation can be represented with a truth table

Manipulating Boolean Expressions

- Sometimes a long Boolean expression has the same truth table as another, shorter expression
- It tends to be desirable to use the shorter versions
- There are a variety of methods which can be used to simplify expressions

Karnaugh Maps

- Can be used to simplify Boolean expressions
- The tables are filled in corresponding to the expression’s truth table
- Can be used for a truth table with two, three or four variables
- It’s important that the values in the columns and rows are written using Gray code
- Columns and rows only ever differ by one bit, including wraparound
- To simplify a Boolean expression:
    - First write your truth table as a Karnaugh map
    - Then highlight all of the 1s in the map with a rectangle
    - The larger the rectangle you can highlight at once the better
    - Only groups of 1s with edges equal to a power of 2 (1, 2 or 4 in a row) can be highlighted, wraparound is included
    - Remove variables which change within these rectangles from the expression
    - Keep variables which do not change, but negate to become True if required