AQA Computer Science A-Level

4.6.5 Boolean algebra

Advanced Notes

Specification:

4.6.5.1 Using Boolean algebra:

Be familiar with the use of Boolean identities and De Morgan's laws to manipulate and simplify Boolean expressions.

Boolean algebra

Just like algebra in Mathematics, Boolean algebra concerns representing values with letters and simplifying expressions. Boolean algebra uses the Boolean values TRUE and FALSE which can be represented as 1 and 0 respectively.

Notation

Expression	Meaning
$A,\ B,\ C,$ etc.	An unknown Boolean value being represented by a letter just like x or y in conventional algebra.
$\overline{A}$	NOT A. An overline represents the NOT operation being applied to what is below the line.
$A \bullet B$	A AND B, said "A dot B" where a dot represents the AND (multiplication) operation.
AB	An alternative notation for A AND B. Just like in Mathematics, the product of two algebraic values can be represented without any symbol.
$A + B$	A OR B, where an addition symbol represents the OR operation.

Order of precedence

Algebraic operations have an order of precedence, meaning that some operations must be applied before others. You may have met BODMAS in Mathematics, this is the same idea.

Operator	Precedence
Brackets	Highest
NOT	.
AND	.
OR	Lowest

For example, the expression B OR NOT C AND A would actually be carried out in the order B OR ((NOT C) AND A).

Boolean identities

There are a number of useful identities which can be used to simplify Boolean expressions.

$A \cdot 0 = 0$ — Anything AND 0 is always 0. This is because the AND operation represents multiplication.

$B \cdot 1 = B$ — Anything AND 1 is always the original value. This is because the AND operation represents multiplication.

$C \cdot C = C$ — Any Boolean value AND itself is equivalent to just the value, as the truth table below shows.

C	C • C
1	1 × 1 = 1
0	0 × 0 = 0

$D + 0 = D$ — Any Boolean value OR 0 is the equivalent of adding 0 to the value, which leaves the value unchanged.

$E + 1 = 1$ — Any Boolean value OR 1 is the equivalent of adding 1 to the value, which will always result in 1.

$F + F = F$ — Any Boolean value OR itself equals the value itself, as the truth table shows.

F	F + F
1	1 + 1 = 1
0	0 + 0 = 0

$\overline{\overline{G}} = G$ — Any Boolean value with two lines above has had the NOT operation performed on it twice, meaning the value has not been changed.

Note

In Boolean algebra,
1 + 1 = 1
i.e. TRUE + TRUE = TRUE

De Morgan's laws

Named after British logician Augustus De Morgan, these two laws of Boolean algebra come in incredibly useful when simplifying expressions.

De Morgan's laws can be remembered by recalling the phrase:

"break the bar and change the sign."

Where "the bar" refers to an overline representing the NOT operation and "the sign" refers to changing between + (OR) and • (AND).

For example, the Boolean expression $\overline{A + B}$ can have De Morgan's law applied to it as follows:

Break the bar:

$$\overline{A} + \overline{B}$$

Change the sign:

$$\overline{A} \bullet \overline{B}$$

$$\overline{A + B} = \overline{A} \bullet \overline{B}$$

De Morgan's law can also be applied in reverse, by changing the sign and building the bar.

For example, the Boolean expression $\overline{C} + \overline{D}$ can be simplified as follows:

Change the sign:

$$\overline{C} \bullet \overline{D}$$

Build the bar:

$$\overline{C \bullet D}$$

$$\overline{C} + \overline{D} = \overline{C \bullet D}$$

Distributive rules

Just like expanding brackets in Mathematics, you can use distributive rules in Boolean algebra as follows:

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

Examples

Example 1

Simplify the Boolean expression $A + \overline{B \cdot A}$

$A + \overline{B \cdot A}$

Use De Morgan's laws. Break the bar and change the sign.

$= A + \overline{B} + \overline{A}$

Use $A + \overline{A} = 1$

$= \overline{B} + 1$

Use $A + 1 = 1$

$= 1$

Example 2

Simplify the Boolean expression $C \cdot B + \overline{C} \cdot B$

$C \cdot B + \overline{C} \cdot B$

Take out B as a common factor

$B \cdot (C + \overline{C})$

Use $A + \overline{A} = 1$

$B \cdot (1)$

Use $A + 1 = A$

B