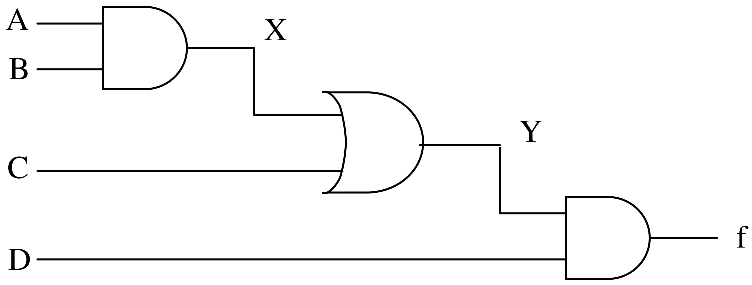
3. Boolean Minimisation Using Boolean Algebra

- Boolean algebra provides a concise way to express the operation of a logic circuit formed by the combination of logic gates so that the output can be determined for various combinations of the input values.

- Before we consider how to use Boolean algebra to minimise a logic circuit, let us first consider how to analyse a circuit.

3.1 Boolean Analysis of Logic Circuits

- Consider the following circuit with inputs A, B, C and D and output f. Intermediate variables X and Y have been added for convenience.



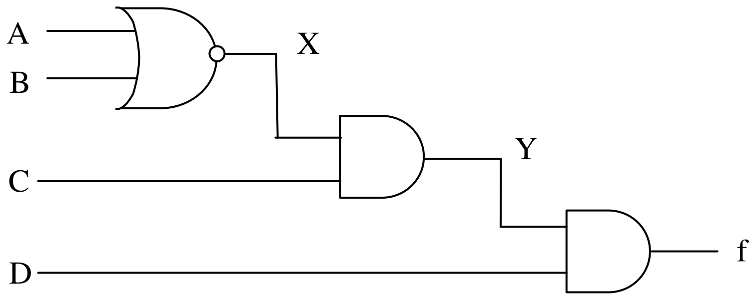
- We can obtain a Boolean expression for this circuit as follows. Writing out an expression for each of the gate outputs f, Y and X, we get:

$f = DY \qquad Y = C + X \qquad X = AB$

- Combining these three expressions, we obtain the following Boolean expression relating the output f to the inputs A, B, C and D:

$$\mathbf{f = D\,(C + AB)}$$

- ***Ex. 3.1 Obtain a Boolean expression for the following circuit:***



- Once we have obtained a Boolean expression for a logic circuit, it is possible to construct a truth table from this expression.

- To do this, we can take every combination of the inputs and determine the corresponding output using the Boolean expression. For example, consider the expression:

$$f = D\,(C + AB)$$

- Taking the combination of ABCD = 0000, we can determine f as follows:

$AB = 0.0 = 0$ (*i.e. 0 AND 0 = 0*)

$C + AB = 0 + 0 = 0$ (*i.e. 0 OR 0 = 0*)

$f = D\,(C + AB) = 0.0 = \mathbf{0}$

- We can repeat this for all 16 combinations of ABCD. Note that we can also determine this output directly from the logical circuit.

- Alternatively, and a quicker method, is to use the Boolean expression to simply find the values of the variables that make the expression equal to 1.

- Thus, for $f = D\,(C + AB) = 1$ we can deduce that:

$D = 1$ and $C + AB = 1$

(both inputs in an AND gate = 1 if the output = 1)

- Now for $C + AB = 1$:

$C = 1$ and / or $AB = 1$

(refer to the truth table of the OR gate)

- And for $AB = 1$:

$A = 1$ and $B = 1$

- So, overall, we can state that:

$f = D\,(C + AB) = 1$ when:

D = 1 and C = 1 regardless of A and B

and

D = 1 and A = 1 and B = 1 regardless of C

All other combinations result in $f = 0$.

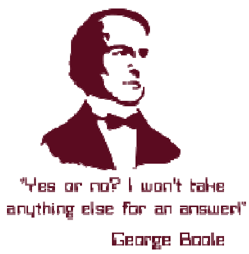
- Hence, we can complete the truth table as follows:

A	B	C	D	f = D (C + AB)
0	0	0	0	**0**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	**1**
0	1	0	0	**0**
0	1	0	1	**0**
0	1	1	0	**0**
0	1	1	1	**1**
1	0	0	0	**0**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	**1**
1	1	0	0	**0**
1	1	0	1	**1**
1	1	1	0	**0**
1	1	1	1	**1**

- *Exercise – determine the truth table for the circuit in Ex 3.1.*

- We are now going to look at the laws, rules and theorems of Boolean algebra that can be used to simplify general Boolean expressions.

3.2 Boolean Algebra – Rules and Theorems



- The following Boolean rules and theorems are useful:

$$\left.\begin{array}{l} A + 0 = A \\ A + 1 = 1 \\ A + \bar{A} = 1 \\ A + A = A \end{array}\right\} \text{properties of the OR gate}$$

$$\left.\begin{array}{l} A.0 = 0 \\ A.1 = A \\ A.\bar{A} = 0 \\ A.A = A \end{array}\right\} \text{properties of the AND gate}$$

$$\left.\bar{\bar{A}} = A \right\} \text{property of the NOT gate}$$

$$A + AB = A$$
$$A\,(A + B) = A$$
$$A + \bar{A}B = A + B$$

- The commutative, associative and distributed laws can also be applied as follows:

A + B = B + A
A.B = B.A

Commutative **–** order of variables does not matter when using the AND and OR operations

A (BC) = (AB) C
A + (B + C) = (A + B) + C

Associative **–** the result of applying an operation over 3 variables is not affected by the order taken

A (B + C) = AB + AC
(A + B)(A + C) = A + BC

Distributive **–** certain operations can be combined in different ways to give the same result, as shown

- Finally, and most importantly, we have **De Morgan's theorems**, as follows:

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

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

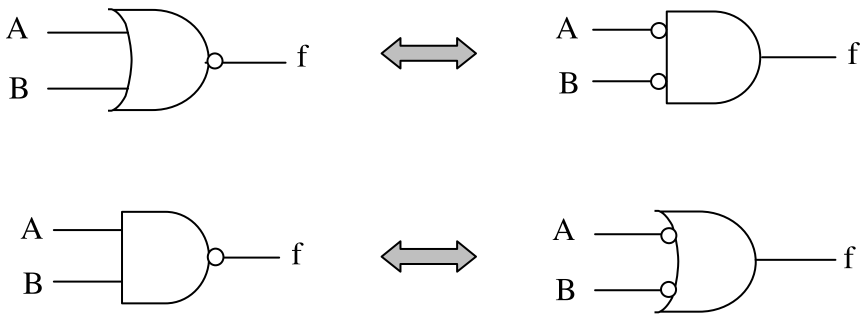

Augustus De Morgan

- The first theorem states that complement of a sum of variables is equal to the product of the complement of the variables.

- The second theorem states that complement of a product of variables is equal to the sum of the complement of the variables.

- These theorems extend to any number of variables.

- Visually, we can illustrate these theorems as follows:



- We will examine these theorems in more detail when we look at NAND and NOR implementation.

- All the aforementioned rules and theorems can be used to minimise Boolean expressions.

Aside – some proofs …

- Show that **A + AB = A**

- Show that **A(A + B) = A**

- Show that $\mathbf{A + \overline{A}B = A + B}$

- Show that **(A + B)(A + C) = A + BC**

3.3 Boolean Algebra – Minimisation

- In this section we will use the various rules, laws and theorems of Boolean algebra to simplify and minimise general Boolean expressions.

- It is important to minimse Boolean functions as this often leads to a reduction in the number of gates and/or inputs required.

- For example, consider the following function:

$$f = AB + A\,(B + C) + B\,(B + C)$$

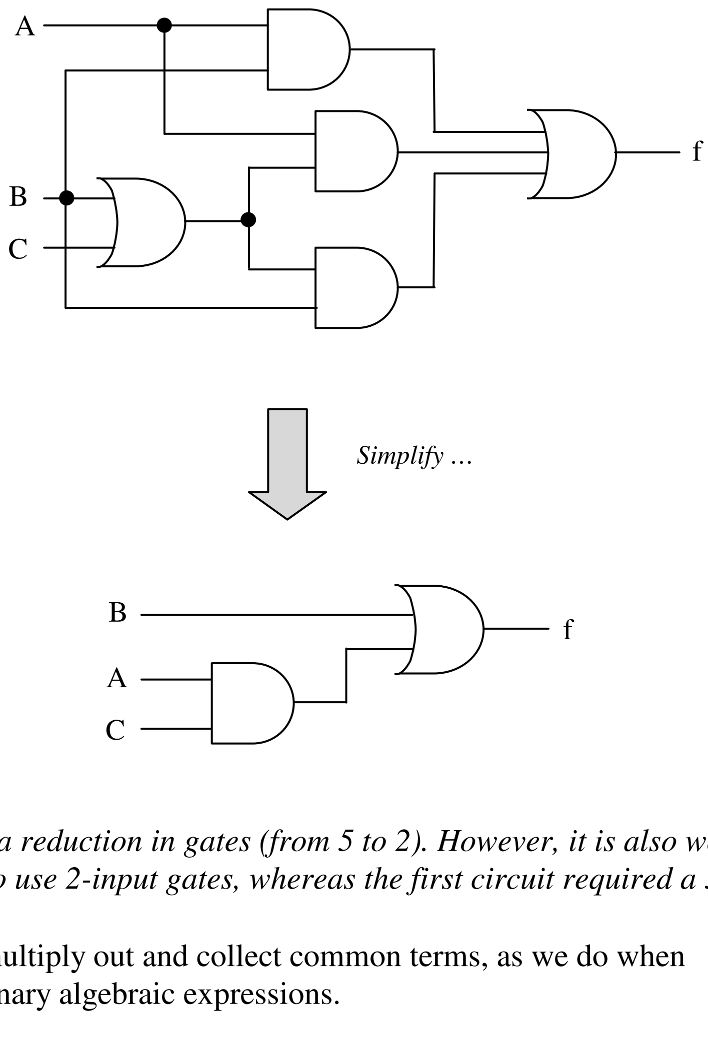
- As it stands, this requires 5 gates – *can you work out what gates these are?*

- We can minimise this function as follows:

$AB + A(B + C) + B(B + C)$	
$= AB + AB + AC + BB + BC$	(*distribution*)
$= AB + AC + B + BC$	($X + X = X$; $X.X = X$)
$= B(A + 1 + C) + AC$	(*note* $X = X.1$)
$= B + AC$	($X + 1 = 1$)

- Now, this simplified expression, $f = B + AC$, only requires 2 gates.

- Visually, we can illustrate the obvious reduction in gates be looking at the circuit before and after simplification:



- *Clearly there is a reduction in gates (from 5 to 2). However, it is also worth noting that we now only need to use 2-input gates, whereas the first circuit required a 3-input OR gate.*

- In general, we multiply out and collect common terms, as we do when simplifying ordinary algebraic expressions.

- In some cases, we need to add in additional terms (without changing the function) in order to make the minimisation process easier.

- ***Ex. 3.2 Minimise the following logic function:***

$$f_{(A,B,C)} = \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC$$

Can we reduce this further?

We know that X + YZ = (X + Y) (X + Z). Hence:

- Here is an ***alternative solution*** to the same problem:

- *Note:* In this instance, adding an extra term resulted in a much easier solution.

- ***Ex. 3.3 Minimise the following logic function:***

$$f_{(X,Y,Z)} = XY + \overline{X}Z + YZ$$

- ***Ex. 3.4 Write out the Boolean function represented by the following truth table and hence obtain a minimal Boolean expression:***

A	B	C	F
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

- Two key limitations of the Boolean algebraic approach are:

  (i) procedures are difficult to apply systematically and
  (ii) it is difficult to determine the optimum solution.

- The task is greatly simplified by visualising the Boolean functions using **Karnaugh Maps.**