MEI

Mathematics in Education and Industry

MEI Structured Mathematics

Module Summary Sheets

Decision Mathematics 2
(Version B: references to new book)

Topic 1: Linear Programming

Topic 2: Networks

Topic 3: Decision analysis

Topic 4: Logic

MEI, Oak House, 9 Epsom Centre, White Horse Business Park, Trowbridge, Wiltshire. BA14 0XG.
Company No. 3265490 England and Wales Registered with the Charity Commission, number 1058911
Tel: 01225 776776. Fax: 01225 775755.

Summary D2 Topic 1: Linear Programming – 1

References: Chapter 1 Pages 1-4

Terminology:

Basic Variables

In forming the Simplex Tableau there are a number of variables (x, y, s_1 etc.). If the numbers in the column for a variable are all 0 bar a single 1, then the variable takes the RHS value from the row containing the 1. The variable is called a basic variable. All other variables are non-basic and take the value 0.

Pivots

The pivot column is the column containing the (non-basic) variable which we are going to adjust. The pivot row is the row in which we will put a 1 in the pivot column, and which we will then use to adjust the other rows to have zeros in the pivot column. The pivot element is the value which is in the pivot column and row.

E.g. Maximise $4x + 5y$ subject to

$3x + 2y \le 18$ $\quad 2x + 4y \le 24$ $\quad 2y \le 11$

1. Add slack variables to all the constraints, and write the objective function in the form $P - 4x - 5y = 0$, placing the coefficients of the objective function and all the constraints in a tableau.

P	x	y	s_1	s_2	s_3	RHS
1	-4	-5	0	0	0	0
0	3	2	1	0	0	18
0	2	4	0	1	0	24
0	0	2	0	0	1	11

2. Choose a variable to increase - its coefficient in the objective row must be <u>negative</u> - say x.
3. Using "RHS ÷ pivot column" for each constraint row gives 18/3=6, 24/2=12 and 11/0 undefined. The first is the smallest positive so this becomes the pivot row with the pivot element at the intersection.
4. Divide through the pivot row by 3.
5. Add 4 × pivot row to first row, subtract 2 × pivot row from 3rd row, creating a new tableau.

1	0	$-\frac{7}{3}$	$\frac{4}{3}$	0	0	24
0	1	$\frac{2}{3}$	$\frac{1}{3}$	0	0	6
0	0	$\frac{8}{3}$	$-\frac{2}{3}$	1	0	12
0	0	2	0	0	1	11

Note: this represents solution $x = 6$, $y = 0$, with profit of 24 and slack of 12 and 11 on 2nd and 3rd constraints respectively.

6. Repeat with y column as pivot column. 3rd row will be pivot row.

1	0	0	0.75	0.875	0	34.5
0	1	0	0.5	-0.25	0	3
0	0	1	-0.25	0.375	0	4.5
0	0	0	0.5	-0.75	1	2

This represents solution of $x = 3$, $y = 4.5$, with profit of 34.5 and slack of 2 on 3rd constraint.

Since there are now no negative values in the row corresponding to the objective function, we cannot choose any variables to increase and have therefore found the optimal solution.

References: Chapter 1 Pages 5-11

Exercise 1A Q. 2, 5

The Simplex Algorithm

1. Represent the problem in a tableau.
2. Use the objective row to find the pivot column.
3. Use the ratio test to find the pivot element.
4. Divide through the pivot row by the pivot element.
5. Add/subtract multiples of the transformed pivot row to/from the other rows to create zeros in the pivot column.
6. Repeat until there are no negatives in objective row.
7. Set basic variables (with columns containing a 1 and 0s) to corresponding RHS, and non–basic variables to zero.

References: Chapter 1 Pages 16-17

Minimising **e.g. Minimise $2x - y$**

Either Maximise $-2x + y$

Or stick to $2x - y$ but look for positive values in the objective row and reduce them to zero.

There are many occasions, particularly when minimising, that the origin is not a feasible solution. In such cases you would usually use either the two-stage or big M approach (see next sheet).

References: Chapter 1 Page 31

Equalities

E.g. $2x + y = 8$ This can be written as two inequalities: $2x + y \le 8$ and $2x + y \ge 8$ and then treated as normal.

Summary D2 Topic 1: **Linear Programming – 2**

References: Chapter 1 Pages 17-22

Example 1.2 Page 28

Exercise 1B Q. 2

Two Stage Algorithm

1. Introduce a surplus variable and an artificial variable to each $\geq$ constraint.
2. Define a supplementary objective function – the sum of the artificial variables.
3. Express the supplementary objective function in terms of the other variables by eliminating the artificial variables.
4. Use simplex to minimise the supplementary objective, keeping track of the original objective.
5. If the minimum value of the supplementary objective is greater than 0,conclude that there is no feasible solution to the original problem.
6. If the minimum value of the supplementary objective is 0, then we have a feasible solution to the original problem. Discard the supplementary objective and the artificial variables and use simplex to proceed to the optimal solution.

E.g. Maximise $4x+5y$ subject to
$3x + 2y \leq 18$, $2x + 4y \leq 24$, $y \geq 2$

1. We add slack variables to the first two constraints. The third constraint requires us to subtract a surplus variable, and then to add an artificial variable (so that $x = y = 0$ can be satisfied without any variable taking a negative value). The original objective function is written $P - 4x - 5y = 0$ but this becomes the second row and we form a new objective function. (See 2.)

$$3x + 2y + s_1 = 18$$
$$2x + 4y + s_2 = 24$$
$$y - s_3 + a = 2$$

2. Our new objective is to minimise the sum of the artificial variables (in this case, just a single one, a) so we want to minimise $A = a$.

3. This must be expressed in terms of the other variables giving Minimise $A = -y + s_3 + 2$.

4. Form tableau and perform one or more iterations.

A	P	x	y	s_1	s_2	s_3	a	RHS
1	0	0	1	0	0	–1	0	2
0	1	–4	–5	0	0	0	0	0
0	0	3	2	1	0	0	0	18
0	0	2	4	0	1	0	0	24
0	0	0	1	0	0	–1	1	2

Since we are minimising we choose y as our pivot value as it has a positive coefficient. The pivot row is the minimum of 18/2, 24/4 and 2/1 so the 4th row becomes our pivot row with the pivot value of 1.

A	P	x	y	s_1	s_2	s_3	a	RHS
1	0	0	0	0	0	0	–1	0
0	1	–4	0	0	0	–5	5	10
0	0	3	0	1	0	2	–2	14
0	0	2	0	0	1	4	–4	16
0	0	0	1	0	0	–1	1	2

5. We have a minimum value of zero.

6. Our artificial variable a is zero, so this can be eliminated along with the associated objective, leaving a standard simplex tableau to maximise P to be solved.

P	x	y	s_1	s_2	s_3	RHS
1	–4	0	0	0	–5	10
0	3	0	1	0	2	14
0	2	0	0	1	4	16
0	0	1	0	0	–1	2

References: Chapter 1 Pages 24-26

Example 1.3 Page 29

Exercise 1C Q. 2

Exercise 1D Q. 6, 18

The Big M Method

This requires the same formation of equations using slack, surplus and artificial variables, but instead of forming a new objective function, a large quantity (M) of the sum of the artificial variables is subtracted from the objective function.

E.g. Maximise $4x + 5y$ subject to
$3x + 2y \leq 18$, $2x + 4y \leq 24$, $y \geq 2$

The constraints are as in the previous example and the objective function becomes Maximise $4x + 5y - Ma$

We then substitute for a to obtain Maximise $4x + 5y - M(2 - y + s_3)$.

Re-arranging to bring the y values together gives
$$P - 4x - (5+M)y + Ms_3 = -2M$$

The tableau is then solved as per a single-stage problem until M only appears in columns for artificial variables. Since these can then never be basic, they can be removed and the solution obtained as usual.

Summary D2 Topic 2: **Networks - Floyd**

References:
Chapter 2
Pages 42-48

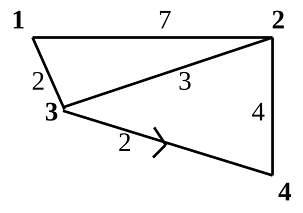
Shortest Paths

One method to find ALL the shortest paths in a network, is to apply Dijkstra's algorithm *n* times, starting at a different vertex each time. Since Dijkstra's algorithm is O(n^2) and we are applying it *n* times, it is O(n^3).

An alternative is to use Floyd's algorithm, which finds all shortest paths in a network by updating *distance* and *route* (*next vertex*) matrices. At each step of the algorithm, a vertex is chosen and each possible route through it is compared with the direct route. The smaller is chosen. Thus if **1** is the chosen vertex and the current shortest distance from **3** to **1** plus the current shortest distance from **1** to **4** is less than the current distance from **3** to **4**, then this distance will be substituted and the *route* matrix updated.

Since Floyd's algorithm considers $(n-1) \times (n-1)$ possible pairings, *n* times, it is O(n^3).
N.B. Alternative notations will use either a dash or zero to represent the distance from **1** to **1**, **2** to **2** etc. in the initial distance matrix. This is not of much importance unless you want to know the shortest distance from (say) **1** to **1** via somewhere else.

Floyds Algorithm Example (for Dijkstra see D1 Topic 3)



distance matrix

	1	**2**	**3**	**4**
1	∞	7	2	∞
2	7	∞	3	4
3	2	3	∞	2
4	∞	4	∞	∞

route matrix

	1	**2**	**3**	**4**
1	1	2	3	4
2	1	2	3	4
3	1	2	3	4
4	1	2	3	4

(Note that **34** is a directed edge so there is no direct connection from **4** to **3**.)

Select vertex **1**, highlighting the column in both matrices, and the row in the first.

∞	7	2	∞
7	∞	3	4
2	3	∞	2
∞	4	∞	∞

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

2 to **2** is 7 + 7 =14 which is less than ∞ so replaces it, and we update the route matrix with the highlighted value from the corresponding row in the previous matrix ("**1**").
2 to **3** is 7 + 2 = 9 so the existing 3 is retained because it is shorter.
The only other change is **3** to **3** becoming 4.

∞	7	2	∞
7	**14**	3	4
2	3	**4**	2
∞	4	∞	∞

1	2	3	4
1	**1**	3	4
1	2	**1**	4
1	2	3	4

Continue to vertex **2**, again highlighting the row and column.

This time there are improvements to **11**, **14**, **41**, **43** and **44**, replacing the current distance with that via **2**.

14	7	2	**11**
7	14	3	4
2	3	**4**	2
11	4	**7**	**15**

2	2	3	**2**
1	1	3	4
1	2	1	4
2	2	**2**	**2**

Exercise 2A
Q. 1(i), 2(i)

Then consider vertex **3**. This gives 7 improved routes.

Note that the improvements for **41** and **44** take a route via **2** since this is the highlighted value in the **4** row of the previous route matrix.

4	**5**	2	**4**
5	**6**	3	4
2	3	**4**	2
9	4	7	**9**

3	**3**	3	**3**
3	**3**	3	4
1	2	1	4
2	2	2	**2**

Finally consider vertex **4**, for which there is no further improvement in this example.

4	5	2	4
5	6	3	4
2	3	4	2
9	4	7	9

3	3	3	3
3	3	3	4
1	2	1	4
2	2	2	2

Interpretation
The shortest path from **1** to **3** is to go direct to **3**, length 2.
For the route **1** to **4**, we should go from **1** to **3** and then find the best route from **3**, which is direct to **4**; 2 + 2.

4 to **3**, which had no direct connection initially, has a route of length 7: **4-2-3.**

Summary D2 Topic 2: ***Networks - The Travelling Salesperson Problem***

References:
Chapter 2
Pages 50-54

Exercise 2C
Q. 3

Travelling salesperson Problem: to find a tour of minimum length which passes through every vertex.

Classical Problem- to find a Hamiltonian Cycle (i.e. a **cycle** which passes through **every** vertex).
A complete network will contain Hamiltonian Cycles.

References:
Chapter 2
Pages 54-59

Upper Bound – Nearest Neighbour Algorithm

- Select any vertex as the starting point
- From vertices not already connected, find the one nearest to the last vertex selected and select it. If two vertices are of equal distance, make an arbitrary selection
- repeat until all the vertices included then return to the starting point to form a cycle.
- The length of the tour is an upper bound for the TSP.
- Select the least upper bound to be an upper bound for the TSP.

This may be repeated starting the algorithm, at a different vertex (which may give a different solution).

Lower Bounds – for each vertex in turn

- Delete the chosen vertex and the edges incident on it from the network.
- Find the length of a minimum connector for the remaining network.
- Choose the two shortest deleted edges and add these to the length of the minimum connector to give a lower bound.
- Select the greatest lower bound to be a lower bound for the network.

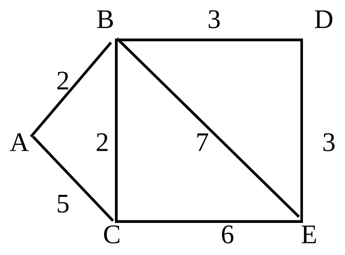
References:
Chapter 2
Pages 59-62

Exercise 2C
Q. 7

Finding a solution

The Nearest Neighbour will always give a suitable tour, but by considering the network a better tour may be obtained using heuristic methods. Alternatively it can be done by systematically exchanging edges in the tour with other edges whose distance is less, or by interchanging towns.

Example: Find the shortest travelling salesperson tour, starting and finishing at A, for the network below.



We construct the distance table for the associated complete graph of shortest distances.

	A	B	C	D	E
A		2	5	5	8
B	2		2	3	6
C	5	2		5	6
D	5	3	5		3
E	8	6	6	3	

Using the Nearest Neighbour Algorithm gives the following sequence of edges
AB 2, BC 2, CD 5, DE 3, EA 8,
Length 20

Interpretation.
This gives the tour: ABC(B)DE(DB)A

deleted vertex	minimum connector for rest	Sum of two shortest deleted adges	lower bound
A	8	7	15
B	14	4	18
C	7	8	15
D	10	6	16
E	7	9	16

The greatest lower bound is 18.
Solution
The result of the upper and lower bound calculations tells us that

$$18 \leq \text{optimal solution} \leq 20$$

The nearest Neighbour provides a suitable tour, but by considering tour on tour improvements we can obtain a better solution.
ABDECBA length 18

Summary D2 Topic 2: Networks - Route Inspection

References:
Chapter 2
Pages 66-69

Route Inspection (Chinese Postman)

In such problems, the solution is a tour of minimum length which includes each edge at least once. There are many equal optimum solutions provided that the graph is Eulerian (every vertex is of even order).

References:
Chapter 2
Page 72

For a network to be traversable it must have zero of two odd nodes only.

Reminder: An odd node is one where an odd number of arcs join.

In any network there is always an even number of odd nodes.
(See also D1 book, page 53)

Exercise 2D
Q. 1(i), 3

To achieve this, consider all possible pairings of odd nodes. For each pairing find the minimum 'cost' of connecting the vertices in their pairs. Choose the pairing with the least cost and add these routes to the network so that their edges are repeated.

The length of the solution will be the total of all the edge lengths (including any duplicates just added).

For $2n$ odd nodes there will be $\frac{(2n)!}{2^n n!}$ pairings to be considered.

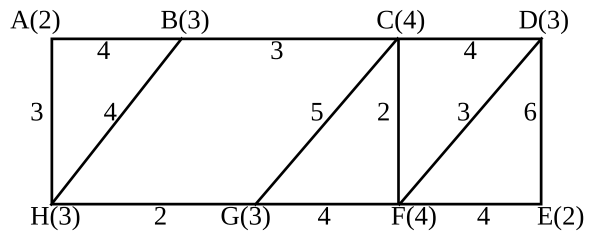
Exercise 2E
Q. 2

Decision Mathematics 2
Version B: page 6
Competence statements N1, 2, 3, 9, 10
© MEI

Route inspection example

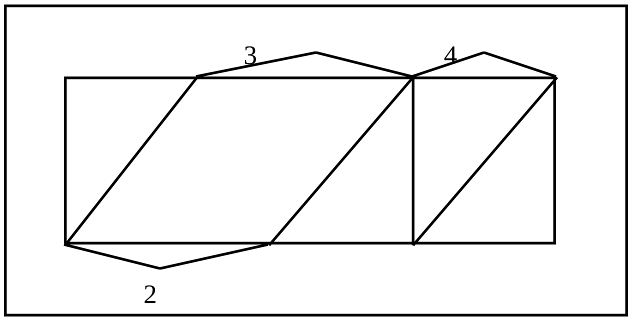
A road gritter needs to cover the network of roads given in the diagram. Both sides of the road can be covered by travelling on it in one direction. The distances, in kilometres, are given on the edges of the diagram. Find the total distance that the road gritter needs to cover to complete the task.

The order of each node is given in brackets.



The total length of the roads is 44 km. However, because there are some odd nodes, some roads will have to be covered twice.
To find the shortest route covering every edge we identify the odd-ordered vertices. These are B, D, G and H. Possible pairings are:
BD + GH = 7 + 2 = 9
BG + DH = 6 + 9 = 15
BH + DG = 4 + 7 = 11

Since the shortest of these is BD + GH, we add edges to these routes. Since the original edges gave a total of 44, our final solution is a route of length 44 + 9 = 53



There are many possible actual routes;
one possible route is ABCDEFDCBHGFCGHA
another possible route is ABHGCBCDCFDEFGHA

Summary D2 Topic 3: ***Decision Analysis***

References:
Chapter 3
Pages 85-88

Exercise 3A
Q. 3, 5

Terminology
Decision trees can be drawn in situations where the outcome of events is uncertain.

EMV (expected monetary value) can be calculated for various courses of action to assist decision-making in situations where probabilities can be assigned to the outcomes.

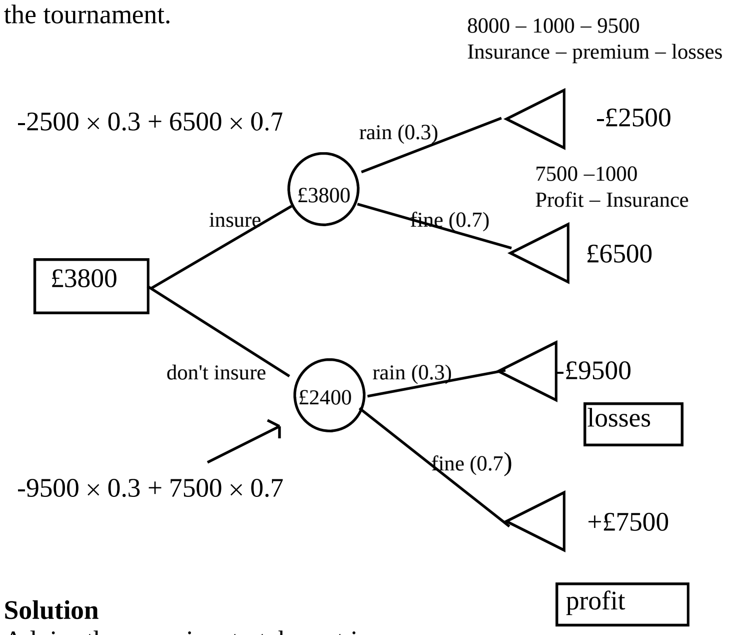
The decision tree has three types of nodes

Decision node	□
Chance node	○
Payoff node	◁

Using Decision Trees

1. Label the pay-off boxes on the right hand side given in the problem.
2. Calculate the EMV from the given probabilities of each event and put in the chance mode.
3. Chose the line out of the decision mode that has the greatest EMV.

Example
The organiser of a tennis tournament has to decide whether to take out insurance against bad weather. For a premium of £1000, she will receive a payout of £8000 if it rains on the day of the tournament. The organiser calculates that if it rains, the losses will total £9500 but if it is fine they will make £7500. The weather forecast estimates that there is a 30% chance of rain on the day of the tournament.



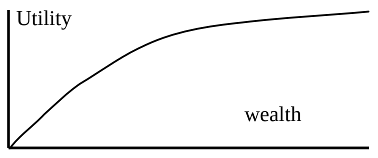
Solution
Advise the organiser to take out insurance
EMV £3800

References:
Chapter 3
Pages 89-90

Exercise 3B
Q. 1

Utility

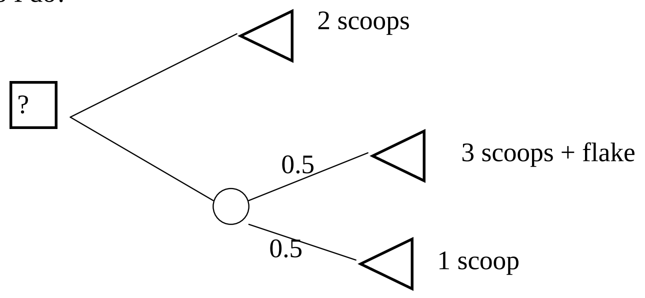
Utility is a measure of the *relative* worth of different courses of actions.
The *utility function* expresses the value of a unit of wealth, £1.00 say, as a function of the decision maker's wealth.
A typical Utility function may look like this:



The utility function is always an increasing function, but the slope usually decreases as wealth increases.

Once a decision maker's utility function is determined, it can be used to help make decisions that will maximise the utility.

Example. A "2 scoop" ice cream cornet costs 80p. Alternatively, I can toss a coin - heads I get 3 scoops and a flake, tails I only get one scoop. If we assume that the flake is worth say 1/2 scoop, then what do I do?



The EMV for the gamble is 2.25 which is better than settling for 2 scoops.
The utility function is a measure of "pleasure" - one interpretation is how much we would be willing to pay for something. If I would be willing to pay 90p for two scoops, 50p for one scoop, and no more than £1.20 for 3 scoops plus flake (a measure of my limited enthusiasm), then the utility is 85p if we gamble, but 90p if we stick with the 2 scoops so our choice will be different.

Summary D2 Topic 4: ***Logic – 1***

References:
Chapter 4
Pages 103-106

Exercise 4A
Q. 1, 6

Logic deals with the structure of systems. These systems can be concerned with:

- language (reasoning and argument).
- Switching circuits
- Combinatorial circuits
- Propositional symbols
- Truth tables

They all represent the same logic.

Propositional Calculus

This is a model of the process of thinking. It can be stated simply using Truth Tables.
Terminology:
p: a statement
~p: the negation of the statement
p∧q: the statement p and the statement q
p∨q: the statement p or the statement q
p⇔q: the statement p if and only if the statement q
p⇒q: the statement p implies the statement q
p⇐ q: the statement p is necessary for the statement q

If the statement is false then it is given the number 0, but if it is true then it is given the number 1.

Example 4.2
Page 109

p	**q**	**p∧q**	**p∨q**
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	1

⇔

p	**q**	**p ⇔ q**
0	0	1
0	1	0
1	0	0
1	1	1

⇐

p	**q**	**p ⇐ q**
0	0	1
0	1	0
1	0	1
1	1	1

⇒

p	**q**	**p ⇒ q**
0	0	1
0	1	1
1	0	0
1	1	1

References:
Chapter 4
Pages 107-108

Exercise 4B
Q. 2, 11

Proof

A proof that one compound proposition is equivalent to another involves showing that their truth values are the same under all circumstances.

Decision Mathematics 2
Version B: page 8
Competence statements p1, 2, 3, 4
© MEI

Example

Represent the statement
"Both a and not b at the same time, or c",
using the different notations.

(a ∧ ~b) ∨ c

Truth table for (a ∧ ~b) ∨ c

a	**b**	**c**	**~b**	**(a ∧ ~b)**	**(a ∧ ~b) ∨ c**
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	1

An alternative table looks like this.

a	**b**	**c**	**~b**	**(a ∧ ~b)**	**∨**	**c**
0	0	0	1	0	**0**	0
0	0	1	1	0	**1**	1
0	1	0	0	0	**0**	0
0	1	1	0	0	**1**	1
1	0	0	1	1	**1**	0
1	0	1	1	1	**1**	1
1	1	0	0	0	**0**	0
1	1	1	0	0	**1**	1

The column in bold gives the final value for this statement. Its values are found taking the columns on either side of it and using the standard truth table for p ∨ q. The final value is not the right hand column as might be expected, but is a rather easier way to construct the column.

E.g. Prove that ~(p ∧ q) = ~p ∨ ~q

The truth table is as follows:

p	**q**	**p ∧ q**	**~(p ∧ q)**	**~p**	**~q**	**~p ∨ ~q**
0	0	0	**1**	1	1	**1**
1	0	0	**1**	0	1	**1**
0	1	0	**1**	1	0	**1**
1	1	1	**0**	0	0	**0**

The two bold columns are the same, proving the proposition.

Summary D2 Topic 4: ***Logic – 2***

References:
Chapter 4
Pages 112-114

Example 4.3
Page 114

Exercise 4C
Q. 3

Boolean Algebra

This is an algebra to manipulate logical expressions.

Identity Rules: $p \wedge 1 = 1$, $p \vee 1 = 1$
$p \wedge 0 = 0$, $p \vee 1 = p$

Associative Rules: $(p \vee q) \vee r = p \vee (q \vee r)$
$(p \wedge q) \wedge r = p \wedge (q \wedge r)$

Commutative Rule: $p \wedge q = q \wedge p$
$p \vee q = q \vee p$

Distributive Rule: $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$
$p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$

De Morgan's Rules: $\sim(p \vee q) = \sim p \wedge \sim q$
$\sim(p \wedge q) = \sim p \vee \sim q$

Double negation: $\sim(\sim p) = p$

Complement Rules: $p \vee \sim p = 1$ $\quad p \wedge \sim p = 0$

Absorption Rules: $p \vee p = p$, $\quad p \wedge p = p$
$p \wedge (p \vee q) = p$ $\quad p \vee (p \wedge q) = p$

Tautology: a proposition which is always true. In a truth table the final column is all 1's; Boolean Algebra simplifies the proposition to a single 1.

Contradiction: a proposition which is never true. In a truth table the final column is all 0's; Boolean Algebra simplifies the proposition to a single 0.

Prove that $(a \vee b) \wedge (a \vee \sim b) = a$

By Boolean Algebra:

$(a \vee b) \wedge (a \vee \sim b)$
$= a \vee (b \wedge \sim b)$ Distributive law
$= a \vee 0$ Complement rule
$= a$ Identity

Truth Table

a	b	~b	a ∨ b	a ∨ ~b	(a ∨ b) ∧ (a ∨ ~b)
0	0	1	0	1	0
0	1	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1

The final column is the same as the first.
As on the last page, this may be written as follows:

a	b	a ∨ b	∧	a ∨ ~b
0	0	0	**0**	1
0	1	1	**0**	0
1	0	1	**1**	1
1	1	1	**1**	1

The bold column is the same as for **a** so they are identical.

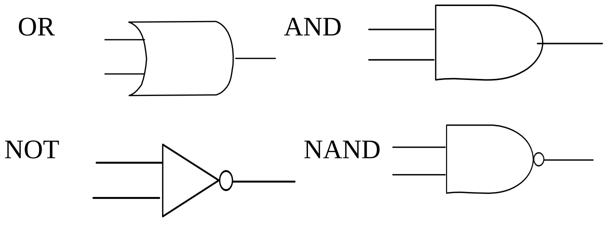
References:
Chapter 4
Pages 115 - 119

Exercise 4D
Q. 2,4

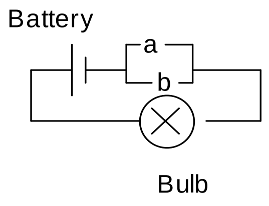
Exercise 4E
Q. 7

Switching and combinatorial circuits

Switching circuits use switches which can either be open or closed (when electrical current can flow).
Combinatorial circuits use logic gates to control the flow of currents.



The following circuit represents a or b



(N.B. The battery and bulb are usually omitted from the diagram.)

Example
The following circuits represent

$(a \wedge \sim b) \vee c$