Syllabus Content:
4.2 Algorithm design methods

4.2.1: Decision tables

- describe the purpose of a decision table
- construct a decision table for a given problem with a maximum of three conditions
- simplify a decision table by removing redundancies

4.2.1 Decision tables:

A decision table is a precise way of modelling logic. Each possible combination of conditions is considered in turn and what action is required.

A decision table is a tool for documenting complicated logic which is a part of some business problem. The aim is to state all combinations of conditions and outcomes which result.

The decision table should be thought of as two parts

- the conditions, which each have a True or False outcome
- the possible outcomes

The table then forma a grid where the outcome can be shown for all possible combinations of conditions.

Condition(s)		
Are you Hungry	Y	N
Action(s)		
Get a mean	Y	

See further examples

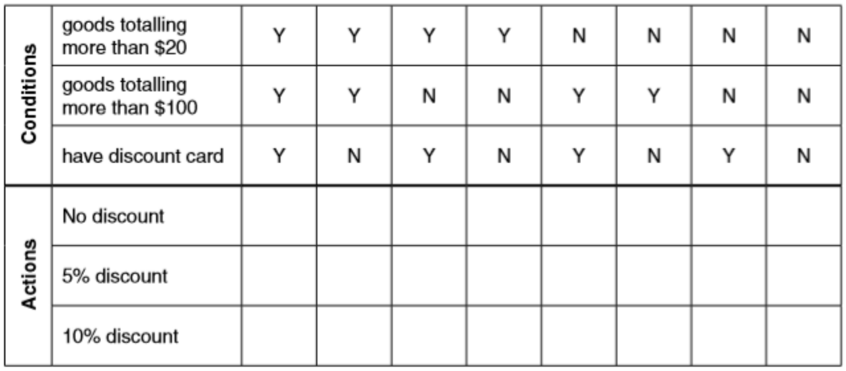
Sample Question

(9608/42/M/J/15)

A shop gives some customers a discount on goods totalling more than $20. The discounts are:

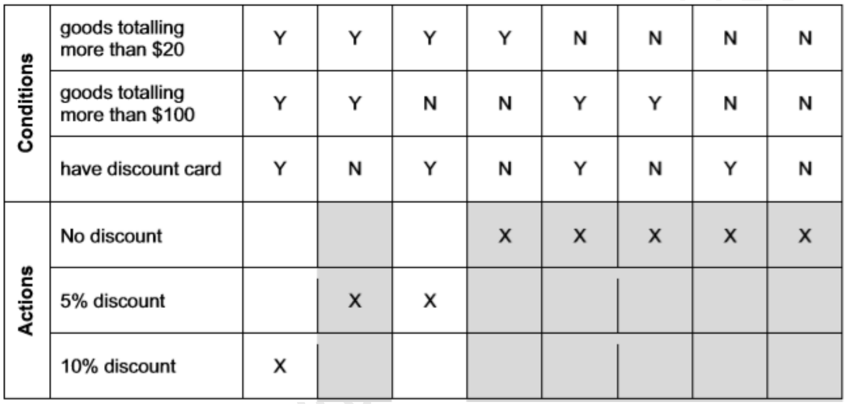
- 5% for goods totalling more than $100
- 5% with a discount card
- 10% with a discount card and goods totalling more than $100

(a) Complete the decision table.



Conditions	goods totalling more than $20	Y	Y	Y	Y	N	N	N	N
	goods totalling more than $100	Y	Y	N	N	Y	Y	N	N
	have discount card	Y	N	Y	N	Y	N	Y	N
Actions	No discount								
	5% discount								
	10% discount								

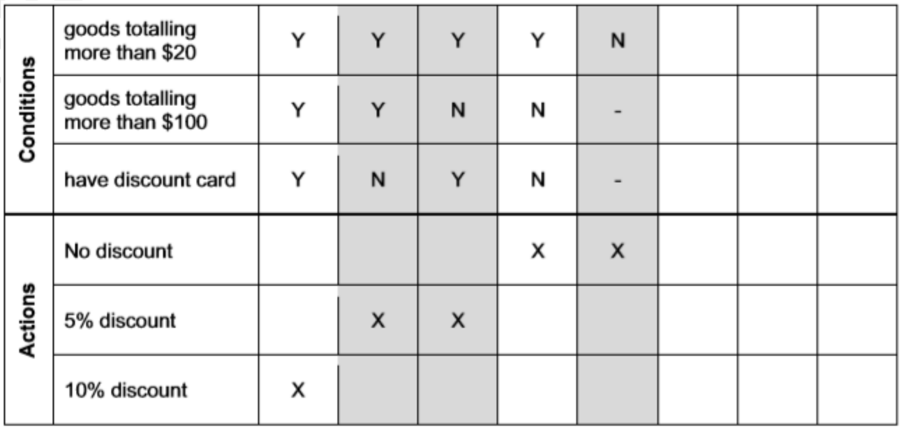
Now we have to complete the actions according to conditions



Conditions	goods totalling more than $20	Y	Y	Y	Y	N	N	N	N
	goods totalling more than $100	Y	Y	N	N	Y	Y	N	N
	have discount card	Y	N	Y	N	Y	N	Y	N
Actions	No discount				X	X	X	X	X
	5% discount		X	X					
	10% discount	X							

- When goods are more than 100$ & customer has discount card: 10% discount
- When goods are more than 100$ but no discount card: 5% discount
- When goods are more than 20$ & has discount card: 5% discount
- When goods are more than 20$ but no discount card: No discount
- When goods are not more than 20$ and more than 100$: Not possible so **X in all other options**

There are redundancies in above diagram of decision table, so we need to simplify our solution.
(b) Simplify your solution by removing redundancies.



Conditions	goods totalling more than $20	Y	Y	Y	Y	N			
	goods totalling more than $100	Y	Y	N	N	-			
	have discount card	Y	N	Y	N	-			
Actions	No discount				X	X			
	5% discount		X	X					
	10% discount	X							

Exercise 24.01 AS & A level Course Book by Sylvia Langfield & Dave Duddell

Produce a decision table for the following scenario:

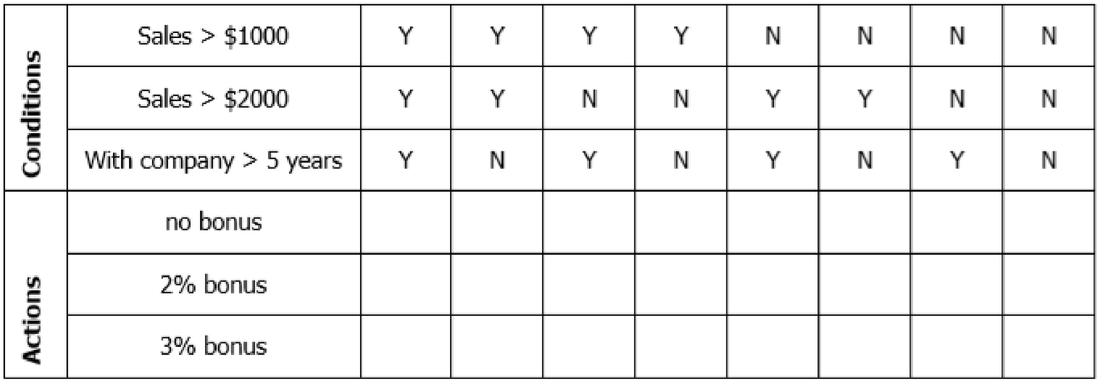
A business pays its sales staff a bonus if they sell more than $1000 worth of goods in a month. The bonus is:

- 1% for selling more than $2000 worth of goods in a month
- 1% if the member of sales staff has been with the company for more than 5 years
- 2% if the member of sales staff has been with the company for more than 5 years and has sold more than $2000 worth of goods in a month

Answer First fill in the conditions.

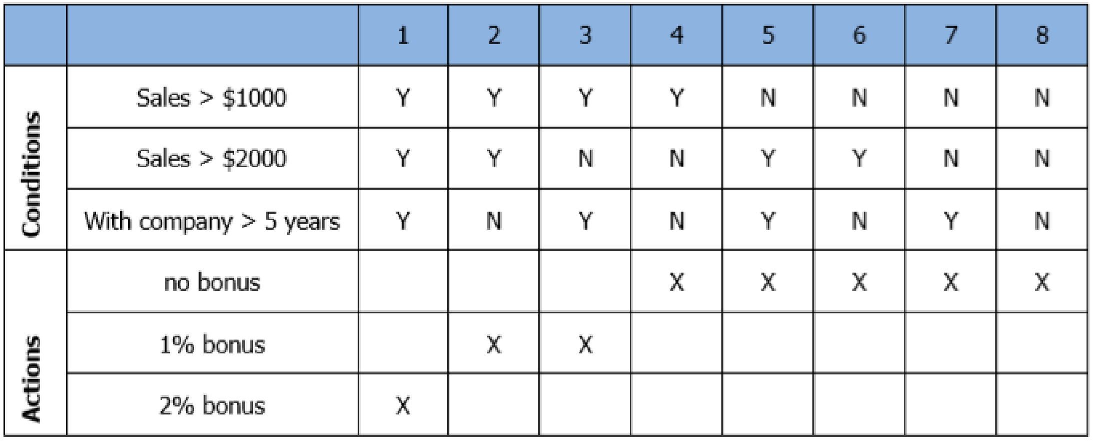
Note that it does not appear to make sense to say **N** for sales more than $1000, but **Y** for sales more than $2000.

However, disregard this in the unsimplified table.



Conditions	Sales > $1000	Y	Y	Y	Y	N	N	N	N
	Sales > $2000	Y	Y	N	N	Y	Y	N	N
	With company > 5 years	Y	N	Y	N	Y	N	Y	N
Actions	no bonus								
	2% bonus								
	3% bonus								

Now work through the bullet points and enter **X** for the action that applies. Remember, one and only one action needs to be chosen for each column. Note that **columns 5 and 6** don't really make sense. **So the action is no bonus.**



		1	2	3	4	5	6	7	8
Conditions	Sales > $1000	Y	Y	Y	Y	N	N	N	N
	Sales > $2000	Y	Y	N	N	Y	Y	N	N
	With company > 5 years	Y	N	Y	N	Y	N	Y	N
Actions	no bonus				X	X	X	X	X
	1% bonus		X	X					
	2% bonus	X							

Now look for redundancies. You only need to look at columns where there is the same action.

Columns 2 and 3: the conditions are mutually exclusive, so no redundancies. Action of 'no bonus': the second and third conditions show every possible combination, but if the first condition is N, then the action is no bonus. In other words, the **second** and **third** conditions can be classed as **'don't care'** and replaced by a **dash.**

So we end up with this simplified table.

		1	2	3	4	5
Conditions	Sales > $1000	Y	Y	Y	Y	N
	Sales > $2000	Y	Y	N	N	-
	With company > 5 years	Y	N	Y	N	-
Actions	no bonus				X	X
	1% bonus		X	X		
	2% bonus	X				

Note that when writing program code from this simplified table, the nested IF statements should reflect the state of the table.

```
IF Sales > 1000
        THEN  IF Sales > 2000
                THEN  IF WithCompany > 5
                        THEN  Bonus · 2
                        ELSE  Bonus · 1
                ENDIF
                ELSE
                        IF WithCompany > 5
                                THEN Bonus · 1
                                ELSE Bonus · 0
                        ENDIF
                ENDIF
                ELSE  Bonus · 0
ENDIF
```

Refrences:

- Computer Science Revision Guide by Tony Piper
- AS & A level Course Book by Sylvia Langfield & Dave Duddell
- A level 9608 Pastpaers