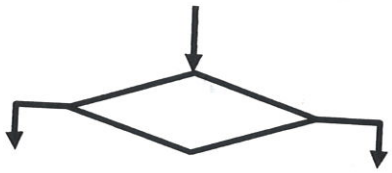
TEMASEK JUNIOR COLLEGE

Temasek Junior College
JC H2 Computing
Problem Solving & Algorithm Design 5 – Decision Tables

5 Decision

- A decision determines an output from a number of inputs by applying some decision logic.
- Decisions can be decomposed into sub-decisions.
  - Top level decisions can be thought of as selecting an answer from a range of possible answers.
  - Lower level decisions often will simply provide input to other decisions.

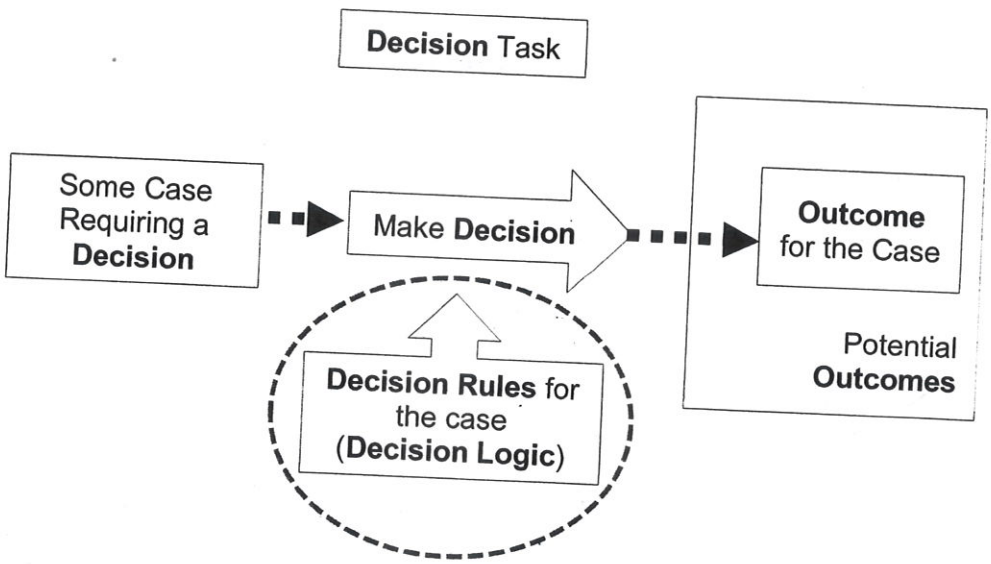


- Two properties should be captured for every decision:
  - **Question**: A natural language statement that represents the decision in the form of a question. This should be specific and detailed.

    e.g. What is the amount of discount allowed on a customer's order?
  - **Allowed Answers**: A natural language description of the possible answers to this question.

    e.g. "Any order of $500 or more received from credit worthy customer attracts discount of 5%, and, similarly, orders of less than $500 attract discount of 3%. Other circumstances must be referred to the supervisor for a decision."
- For action-oriented decisions, the allowed answers represent the responses that the process must handle when the decision model is invoked by a business rule task.

Decision Logic and Decision Task



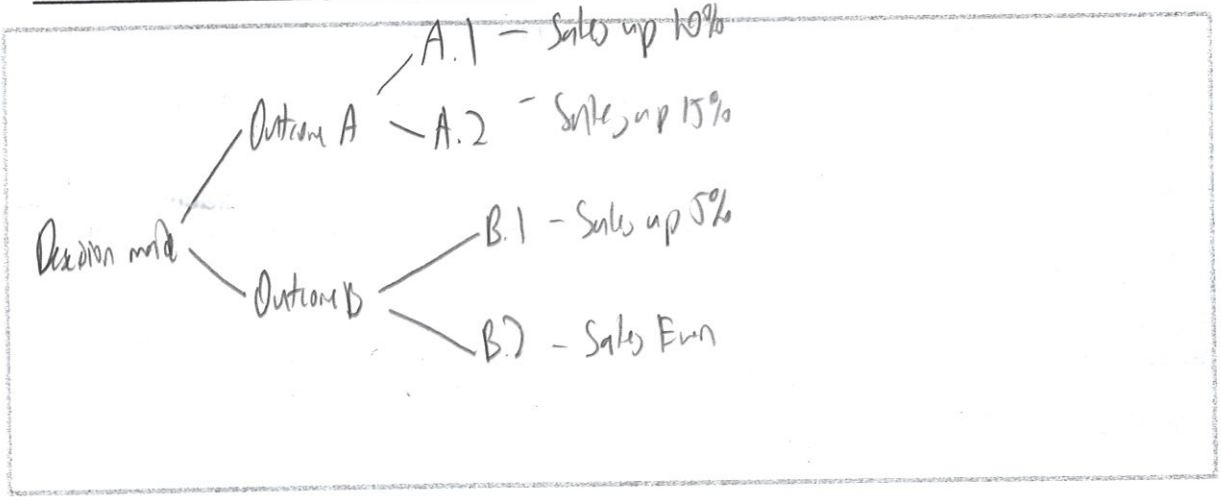
5.1 Decision Trees and Decision Tables

- Often our problem solutions require decisions to be made according to two or more conditions or combinations of conditions.
- Decision trees represent such decision as a **sequence of steps**.
- Decision tables describe **all possible combinations of conditions** and the decision appropriate to each combination.
- **Levels of uncertainty** can also be built into decision trees to account for the relative probabilities of the various outcomes.

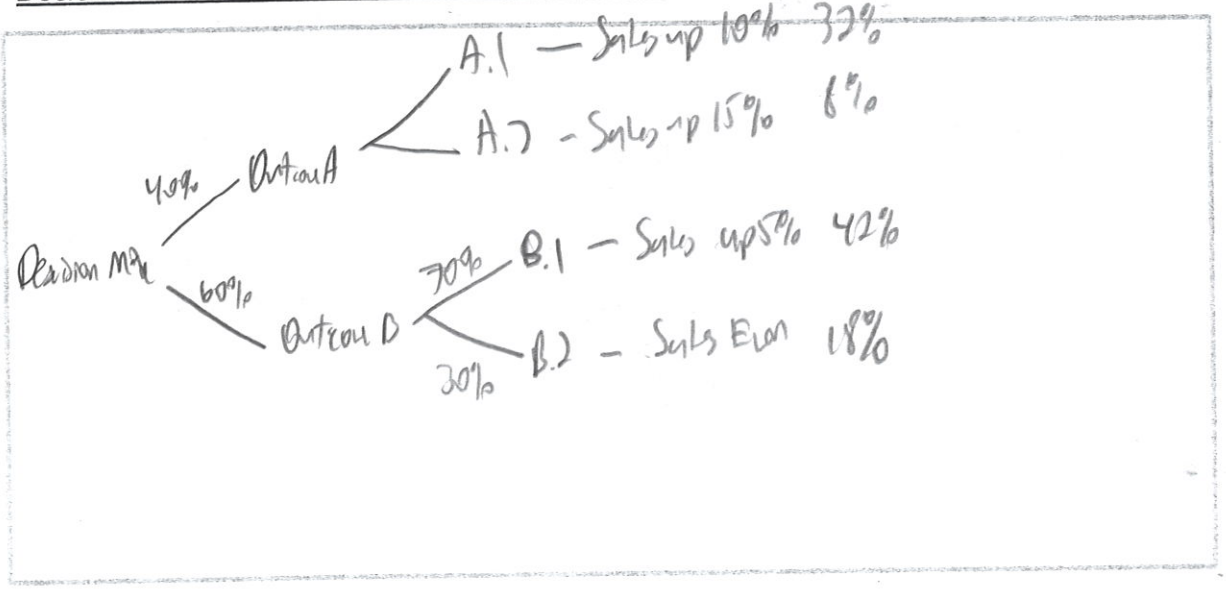
5.2 Decision Trees

- A decision tree shows the possible choices to take in reaching an outcome. It is extremely easy to use.
- Decision trees are a graphical representation of rules.
  - Each inner node corresponds to a decision.
  - Each edge represents an alternative value for the decision.
  - The leaf nodes represent actions or effects.

Decision Tree Showing Possible Outcomes



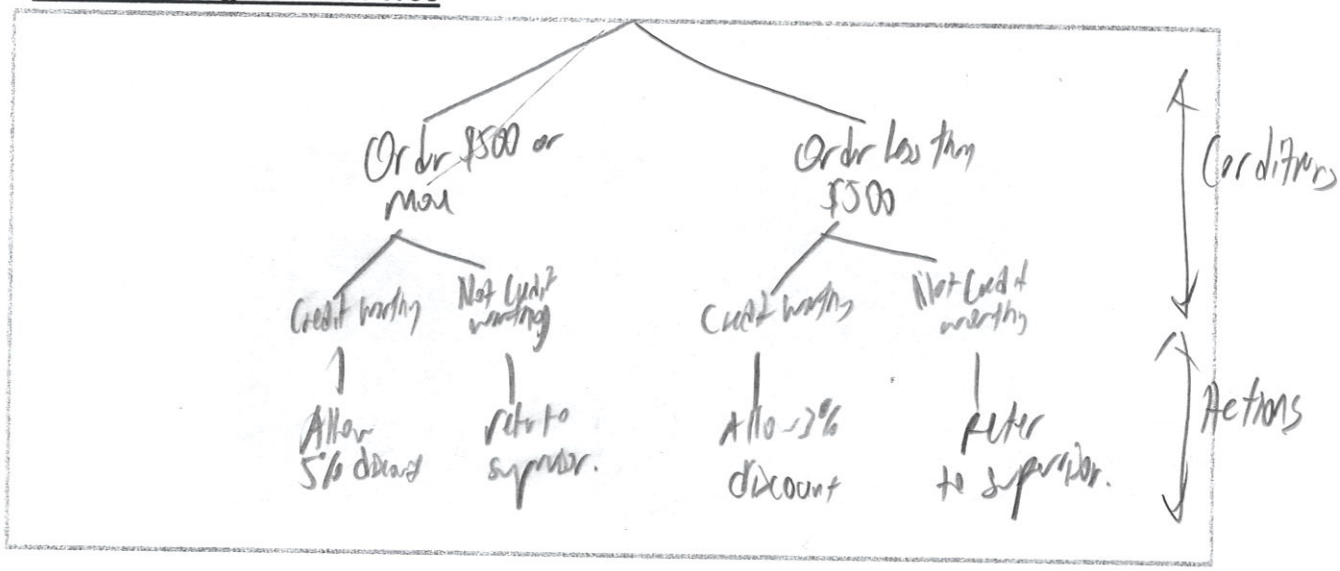
Decision Tree of Outcomes -- Quantifying Uncertainties



Example 1 Credit Worthy Customer
A clerk, in assessing the amount of discount allowed on a customer's order is required to comply with the following policy:

"Any order of $500 or more received from credit worthy customer attracts discount of 5%, and, similarly, orders of less than $500 attract discount of 3%. Other circumstances must be referred to the supervisor for a decision."

Solution using Decision Tree

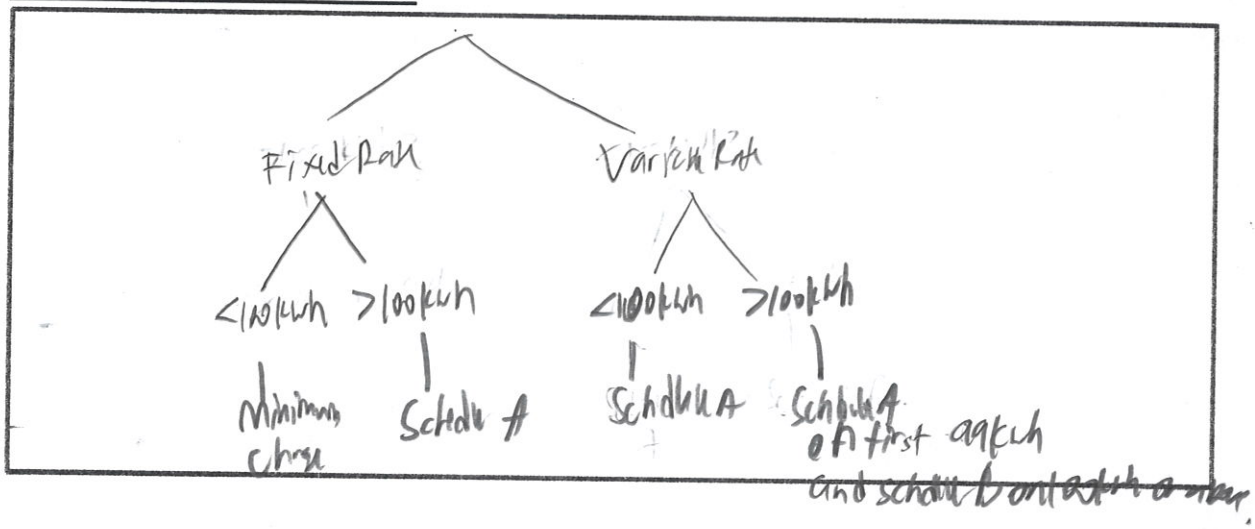


As seen from **Example 1** above, working from the root, we can easily see which condition satisfies any particular situation and then follow the appropriate branch of the tree. Eventually, after all the appropriate conditions have been fulfilled, we will end up at the leaf of the tree where the action to be taken is encountered.

Example 2 Power consumption
Consider the following excerpt from an actual business document:

If the customer account is billed using a fixed rate method, a minimum monthly charge is assessed for consumption of less than 100 kWh. Otherwise, apply a schedule A rate structure. However, if the account is billed using a variable rate method, a schedule A rate structure will apply to consumption below 100 kWh, with additional consumption billed according to schedule B. Draw the decision tree.

Solution using Decision Tree

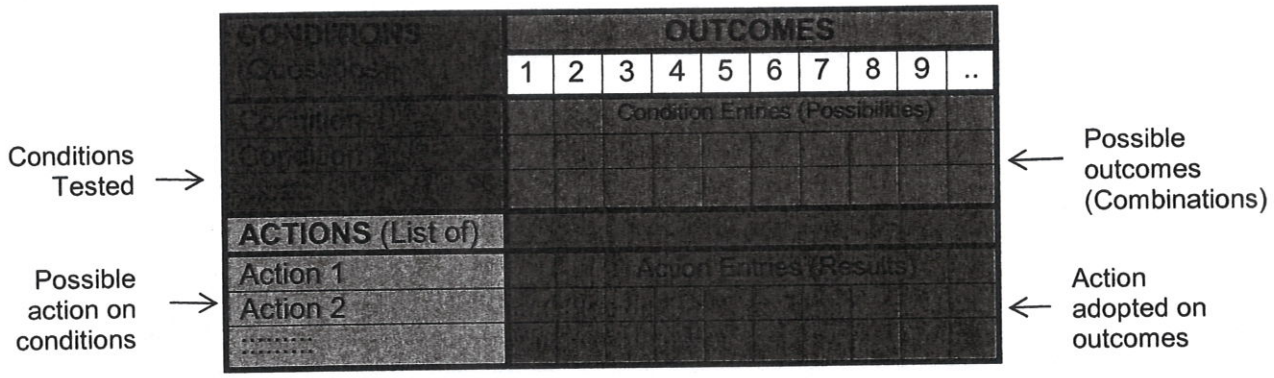


5.3 Decision Tables

A decision table is a precise way of modelling logic. Some programs require multiple decisions to produce the correct output. Managing all possible outcomes of multiple decisions can be a difficult task. A decision table can help to organise the possible decision outcome combinations. Each possible combination of conditions is considered in turn and the action required. It is a method of recording a system in a compact manner. It is more convenient to use than a flowchart when a large number of logical alternatives exist.

5.4 Basic format of a Decision Table

The table can be considered as being divided up into four main quadrants.



In the **top left quadrant**, list all possible **conditions** or questions which are to be involved in the decision-making process.

In the **bottom left quadrant**, list the **actions** that can be taken.

In the **top right quadrant**, **Y** means true and **N** means false, which indicates the conditions being applied in a particular column. The rules which are given the numbers 1, 2, 3 ... is determined by the vertical column underneath each of the numbers. Use a dash '—' to mean that the condition is irrelevant or impossible.

Finally, in the **bottom-right quadrant**, list the actions selected or the results, which relate to the rules indicated in the columns above. For limited action entries, use 'X' to **denote a particular action has been chosen**, or **blank to ignore**. For extended action entries, any statements may be used.

5.5 Types of Decision Tables

1. Limited Condition/Limited Action

- Limited Entry Decision Table
- Limited entry tables are tables where the condition entries are limited to values like "Yes/No", "Y/N", "True/False", etc. Also, limited entry tables use an "X" in the action entries to indicate which actions apply to certain rules.
- List possible outcomes: Y (yes)
  N (no)
  blank (immaterial)

Example 3 Refer to Example 1 (Credit Worthy Customer)
Solution using Limited Entry Decision Table

CONDITIONS	OUTCOMES			
	1	2	3	4
Is order $500 or more?	Y	Y	N	N
Is credit satisfactory?	Y	N	Y	N
ACTIONS				
Allow discount 3%			X	
Allow discount 5%	X			
Refer to supervisor		X		X

2. Extended Condition/Extended Action

- Extended Entry Decision Table
- Both require lesser details in the "Stubs" and any statements may be used in the "Entries".
- In an Extended Entry Table, the conditions are written as more open ended questions such that part of the condition extends itself into the condition entry. For example, instead of having the Condition "Student Under 21?" with the values "Y and "N", you would have the condition "Student Age" and the values "<21" and ">=21".

Example 4 Refer to Example 1 (Credit Worthy Customer)
Solution using Mixed Entry Decision Table

CONDITIONS	OUTCOMES			
	1	2	3	4
Is order $500 or more?	Y	Y	N	N
Is credit satisfactory?	Y	N	Y	N
ACTIONS				
Allow discount of	5%		3%	
Refer to supervisor		X		X

3. Limited Condition/Extended Action

- Mixed Entry Decision Table
- A Mixed Entry table contains a combination of mixed and extended entries. Many of the examples we saw during class were mixed entry.
- List actions adopted on outcomes: X (undertaken)
  blank (ignored)

Example 5 Refer to Example 2 (Power consumption)
Solution using Limited Entry Decision Table

CONDITIONS	OUTCOMES															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Fixed rate acct	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
Variable rate acct	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
Consumption < 100 kwh	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
Consumption >= 100 kwh	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
ACTIONS																
Minimum charge						X										
Schedule A							X			X						
Schedule A on first 99 kwh, Schedule B on kwh 100 +											X					

Final decision table

CONDITIONS	OUTCOMES			
	1	2	3	4
Fixed rate acct	T	T	F	F
Variable rate acct	F	F	T	T
Consumption < 100 kwh	T	F	T	F
Consumption >= 100 kwh	F	T	F	T
ACTIONS				
Minimum charge	X			
Schedule A		X	X	
Schedule A on first 99 kwh, Schedule B on kwh 100 +				X

Example 6 Refer to Example 2 (Power consumption)
Solution using Mixed Entry Decision Table

CONDITIONS	OUTCOMES			
	1	2	3	4
Account type	fixed	fixed	variable	variable
Consumption	< 100	>=100	< 100	>=100
ACTIONS				
Minimum charge	X			
Schedule A		X	X	
Schedule A on first 99 kwh, Schedule B on kwh 100 +				X

5.6 Steps to Create a Decision Table

1. List all the conditions which determine which action to take.

2. Calculate the number of rules required.

   - **Multiply the number of values for each condition by each other.**
     - Example: Condition 1 has 2 values, Condition 2 has 2 values, Condition 3 has 2 values. Thus 2 X 2 X 2 = 8 rules

   - **Decision table rules (number of possible outcomes)**
     - To every decision, there are usually 2 possible outcomes.
     - Therefore for ***n*** decisions, there are $\mathbf{2^n}$ possible outcomes.
     - However, usually several possibilities may result from these related decision.

3. Fill all combinations in the table.

4. Define the action for each rule

5. Analyze column by column to determine which actions are appropriate for each rule.

6. Reduce the table by eliminating redundant columns.

All possible combinations

CONDITIONS	**OUTCOMES**															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Fixed rate acct	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
Variable rate acct	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
Consumption < 100 kwh	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
Consumption >= 100 kwh	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
ACTIONS																
Minimum charge						X										
Schedule A							X			X						
Schedule A on first 99 kwh, Schedule B on kwh 100 +											X					

Actions per combination
(each column represents a different state of affairs)

Example 7 Group testing Part (a)
A manufacturer has an assembly line that produces a particular product. At the end of the assembly process, each product item is comprehensively tested to decide whether that item is acceptable or not.

The tests are split into three groups:

- Group 1: tests to check all dimensions are correct
- Group 2: tests to check strength at various points on the product item
- Group 3: tests to check paint color and coverage

Only if the item passes all three group tests is it accepted. If the Group 1 tests are passed, but exactly one of the other two group tests fails, the item is sent for repair. Otherwise the item is rejected.

(a) Complete the decision table showing all the possible outcomes and results.

CONDITIONS	OUTCOMES							
	1	2	3	4	5	6	7	8
Group 1 tests	0	0	0	0	1	1	1	1
Group 2 tests	0	0	1	1	0	0	1	1
Group 3 tests	0	1	0	1	0	1	0	1
ACTIONS								
Accepted								X
Repair						X	X	
Rejected	X	X	X	X	X			

Possible grading scheme
correct column 1 [1], correct columns 2 and 3 [1], correct column 4 [1], correct columns 5–8 [1]

Example 8 Go to work
You have to decide how to go to work. You only go to work on a weekday. If the time is before 7am, you will take breakfast and walk to work. If the time is between 7am and 8am, you take breakfast and go to work by bus. If the time is after 8am, you skip breakfast and go to work by bus.

Solution using Limited Entry Decision Table

CONDITIONS	OUTCOMES							
	1	2	3	4	5	6	7	8
Is it a weekday?	Y	Y	Y	Y	N	N	N	N
Is it before 7am?	Y	Y	N	N	Y	Y	N	N
Is it after 8am?	Y	N	Y	N	Y	N	Y	N
ACTIONS								
Breakfast	*	X		X	*			
Walk to work	*	X			*			
Go by bus	*		X	X	*			

(* means that no such outcomes exist and these columns have to be omitted.)

5.7 Remove Redundancies

- To find redundancies, we look at each **action** and then check whether the conditions are required.
- If effects (actions) for a several combinations are the same, the combinations can be combined, reducing the number of rules

Example 9 Refer to Example 8 (Go to work)
In **OUTCOMES** 2, ***take breakfast*** and ***walk to work*** only applies if conditions 1 and 2 are true, and condition 3 is false. There is no redundancy here.

In **OUTCOMES** 3, ***go by bus*** only applies if conditions 1 and 3 are true, and condition 2 is false. There is also no redundancy here.

In **OUTCOMES** 4, ***take breakfast*** and ***go by bus*** only applies if condition 1 is true, and conditions 2 and 3 are false. Hence no redundancy.

For **OUTCOMES** 6 to 8, none of the actions need to be taken, which is the same for each of these columns. We can therefore simplify the table by putting a dash in the cells where the condition can be true or false – the action will be the same. The dash is sometimes known as the 'don't care' symbol.

Solution using Limited Entry Decision Table after removing redundancies

CONDITIONS	OUTCOMES							
	1	2	3	4	5	6	7	8
Is it a weekday?	Y	Y	Y	Y	N	N	N	N
Is it before 7am?	Y	Y	N	N	Y	Y	N	N
Is it after 8am?	Y	N	Y	N	Y	N	Y	N
ACTIONS								
Breakfast	*	X		X	*			
Walk to work	*	X			*			
Go by bus	*		X	X	*			

If effects (actions) for a several combinations are the same, the combinations can be combined, reducing the number of rules

CONDITIONS		OUTCOMES			
	after	1	2	3	4
	before	2	3	4	5
Is it a weekday?		Y	Y	Y	N
Is it before 7am?		Y	N	N	-
Is it after 8am?		N	Y	N	-
ACTIONS					
Breakfast		X		X	
Walk to work		X			
Go by bus			X	X	

Why keep this?

[5]

Example 10 Refer to Example 7 Group testing Part (b)
Simplify your solution by removing redundancies.

CONDITIONS	OUTCOMES							
	1	2	3	4	5	6	7	8
Group 1 tests	Y	Y	Y	Y	N	N	N	N
Group 2 tests	Y	Y	N	N	Y	Y	N	N
Group 3 tests	Y	N	Y	N	Y	N	Y	N
ACTIONS								
Accepted	Y							
Repair		Y	Y					
Rejected				Y	Y	Y	Y	Y

Solution: Reduce the table by eliminating redundant columns.

CONDITIONS	OUTCOMES							
	1	2	3	4	5	6	7	8
Group 1 tests	Y	Y	Y	Y	N	N	N	N
Group 2 tests	Y	Y	N	N	Y	Y	N	N
Group 3 tests	Y	N	Y	N	Y	N	Y	N
ACTIONS								
Accepted	Y							
Repair		Y	Y					
Rejected				Y	Y	Y	Y	Y

Note that some columns are identical except for one condition.
Which means that actions are independent from the value of that particular condition.
Hence, the table can be simplified.

CONDITIONS	OUTCOMES							
	1	2	3	4	5	6	7	
Group 1 tests	Y	Y	Y	Y	N	N	N	
Group 2 tests	Y	Y	N	N	Y	Y	N	
Group 3 tests	Y	N	Y	N	Y	N	–	
ACTIONS								
Accepted	Y							
Repair		Y	Y					
Rejected				Y	Y	Y	Y	

First we combine the yellow ones (columns 7 and 8) nullifying the condition.

CONDITIONS	OUTCOMES							
	1	2	3	4	5		7	
Group 1 tests	Y	Y	Y	Y	N		N	
Group 2 tests	Y	Y	N	N	Y		N	
Group 3 tests	Y	N	Y	N	–		–	
ACTIONS								
Accepted	Y							
Repair		Y	Y					
Rejected				Y	Y		Y	

Then the orange ones (columns 5 and 6) nullifying the condition.

CONDITIONS	OUTCOMES							
	1	2	3	4	5			
Group 1 tests	Y	Y	Y	Y	N			
Group 2 tests	Y	Y	N	N	–			
Group 3 tests	Y	N	Y	N	–			
ACTIONS								
Accepted	Y							
Repair		Y	Y					
Rejected				Y	Y			

Notice that yellow and orange columns (5 and 7) are identical but by one condition.
So, we combine them.
Final Solution

correct column 1 [1], correct column 2 [1], correct column 3 [1], correct column 4 [1], correct column 5 [1]

Example 11 Refer to Example 7 Group testing Part (c)

The simplified table produced in part **(b)** is used to design program code. [3]
Three functions are already available: `G1Tests, G2Tests` and `G3Tests`.
These functions return `TRUE` or `FALSE`, indicating the success or otherwise of the group tests.

Write code for a function `Reject` which will return `TRUE` if the product item is to be rejected, otherwise the function will return `FALSE`.

Write code for a function Repair

Example 12 Discount allowable

There are 3 types of customers A, B and C.

- Type A receives a 10% discount on all orders, except for orders above $2000 where the discount is 12%.
- Type B receives a 7% discount on all orders, except for orders above $2000 where the discount is 9%.
- Type C receives a 5% discount on all orders regardless of value.

Solution using Mixed Entry Decision Table

CONDITIONS	**OUTCOMES**							
	1	2	3	4	5	6	7	8
Customer type A?	Y	Y	Y	Y	N	N	N	N
Customer type B?	Y	Y	N	N	Y	Y	N	N
Order > $2000?	Y	N	Y	N	Y	N	Y	N
ACTIONS								
Calculate & print discount	-	-	12%	10%	9%	7%	-	5%
Subtract discount from total	-	-	X	X	X	X	-	X

After Remove Redundancies

CONDITIONS	**OUTCOMES**				
	1	2	3	4	5
Customer type A?	Y	Y	N	N	N
Customer type B?	N	N	Y	Y	N
Order > $2000?	N	Y	N	Y	-
ACTIONS					
Calculate & print discount	10%	12%	7%	9%	5%
Subtract discount from total	X	X	X	X	X

5.8 Advantages of Using Decision Tables

- Possible to check **all** combinations (2^n)
- Show the cause and effect relationship
- Easy to trace from actions to conditions and vice versa
- Use of standardised format (easier to draw and change than flowchart)
- Compact and easier to follow one column – a small table can replace several pages of flowcharts
- Alternatives (actions or conditions) can be grouped to facilitate analysis – provide concise descriptions of logical complex situations

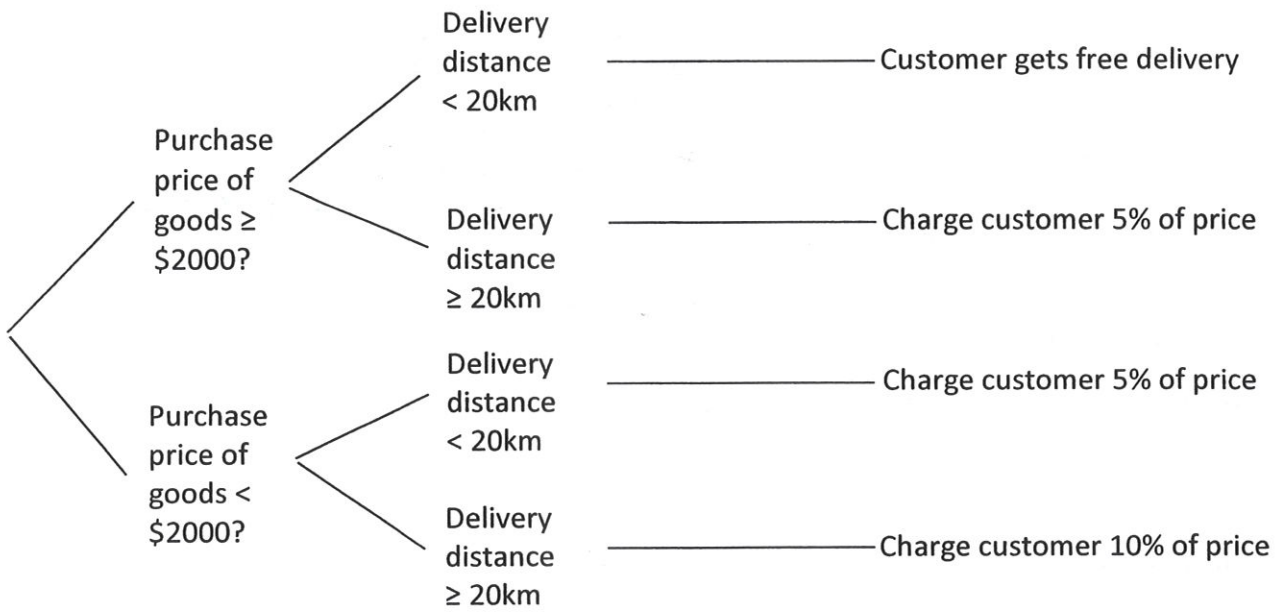
5.9 Limitations of Using Decision Tables

Decision tables are not very popular and are not widely use as flowcharts because:

- Large decision tables can become incomprehensible and difficult to modify
- Flowcharts are better to express the total sequence of events needed to solve a problem
- Flowcharts are more familiar to many programmers and beginners

Tutorial 5 [Decision Table]

1* The following decision tree shows the delivery charges made by a furniture-manufacturing company. Show the same information making use of a decision table.



2* Your company boss sends you a memo outlining a year-end bonus plan with rules attached: "I need a report listing every employee and the bonus I plan to give him or her. Everybody gets at least $100. All the employees in Department 2 get $200, unless they have more than 5 dependents. Anybody with more than 5 dependents gets $1000 unless they're in Department 2. Nobody with an ID number greater than 800 gets more than $100 even if they are in Department 2 or have more than 5 dependents". Create a decision table to manage all the decisions.

3* A systems analyst has been employed to develop a computerised system for a school library which is currently operating manually.

The overdue list is processed in the following manner:

- If a book is overdue then a reminder letter would normally be sent.
- However, if the book is more than five days overdue two further checks are made to see whether the reminder should be replaced by a warning letter:
  (i) If the student has had a previous warning letter, the student will not only receive the warning letter but, in addition, a copy will be sent to the parents of the student.
  (ii) If the student has more than four books overdue, but no previous warning letter, the reminder letter is replaced by a warning letter.

(a) Create a decision table showing all the possible outcomes and results.
(b) Simplify your decision table by removing redundancies.

4 The rules that are used when deciding whether to offer insurance to customers and whether to offer discounts are as follows:

- If the customer has been refused insurance by another company and their car is over 10 years old then insurance is refused.
- If the customer has been refused insurance by another company and their car is not more than 10 years old then insurance without any discount is available.
- If the customer has not been refused insurance by another company and their car is over 10 years old then insurance without any discount is available.
- If the customer has not been refused insurance by another company and their car is less than 10 years old and they have made not more than three claims previously then insurance with a discount is available.

(a) Create a decision table showing all the possible outcomes and results.
(b) Simplify your decision table by removing redundancies.

5* An insurance company calculates the cost of car insurance from a basic price.

The driver may:
- get a discount on the basic price of the insurance
- have to pay an extra charge

The decision is arrived at as follows:
- for a driver aged 25 or over:
  5% discount if no previous accident
  no discount if a previous accident
- for a driver under the age of 25:
  5% discount if no previous accident and licence held for 3 or more years
  no discount if a previous accident but licence held for 3 or more years
  no discount if no previous accident but licence held for less than 3 years
  10% extra charge if a previous accident and licence held for less than 3 years

(a) Complete the decision table. [6]

Conditions	Age under 25	Y	Y	Y	Y	N	N	N	N
	Previous accident	Y	Y	N	N	Y	Y	N	N
	Licence held for 3 or more years more years	Y	N	Y	N	Y	N	Y	N
Actions	10% extra charge								
	No discount								
	5% discount								

(b) Simplify your solution by removing redundancies. [3]

(c) The simplified table produced in **part (b)** is used as a design for program code.
The following identifier table shows the parameters to be passed to the function `CostPercentageChange`. This function returns the percentage change from the basic price as an integer. A discount should be shown as a negative integer. An extra charge should be shown as a positive integer.

Identifier	Data type	Comment
`DriverAge`	INTEGER	Age of driver in years
`HadAccident`	BOOLEAN	Whether driver has had a previous accident
`YearsLicenceHeld`	INTEGER	Number of years the driver has held licence

Write **Python program code** for this function. [6]