MODULE 3

PROCESS SPECIFICATION

WORKED EXAMPLES

6.1 A bank has the following policy on deposits: On deposits of Rs. 5000 and above and for three years or above the interest is 12%. On the same deposit for a period less than 3 years it is 10%. On deposits below Rs. 5000 the interest is 8% regardless of the period of deposit. Write the above process using

(i) Structured English

(ii) A decision table

(i)

```
for each deposit do
    if deposit >=5000
        then if period >= 3 years
            then interest =12%
            else interest =10%
         end if
        else interest =8%
    end if
 end for
```

(ii)

Deposit >= 5000	Y	Y	N
Period >= 3 years	Y	N	–
Interest	12	10	8

6.2 An organization maintains an employee file in which each record has following data:

{ Employee No., employee name, employee gross pay}.

It has been decided to increase the pay as per the following formula:

For pay of Rs. 1000 or less increase 15%.

For pay of more than Rs. 1000 but up to Rs. 2500 increase 10%.

For pay over Rs. 2500 increase 5%.

(i) Write a structured English processing rule corresponding to the above policies.

(ii) Express the policies as a decision table.

(i)

```
While employee records left in file do
    Read Number, name , gross pay
```

```
            if gross pay <=1000
               then increase = gross pay *  0.15
               else if gross pay <= 2500
                     then increase = gross pay *  0.1
                     else increase = gross pay *  .05
                  end if
            end if
         Gross pay = gross pay + increase
         Write Number, name, gross pay
      end while
```

```
(ii)  While employee records left in file do
         Read Number, name, gross pay
            do    Table

            Gross pay <= 1000       Y     N     N
            Gross pay <= 2500       –     Y     N

      Percent increase             15    10     5

         end table
            Gross pay = gross pay*(1+percent increase/100)
            Write Number, name, gross pay
      end while
```

6.3 An offshore gas company bills its customers according to the following rate schedule:

First 500 litres Rs. 10 (flat)
Next 300 litres Rs. 1.25 per 100 litres
Next 30,000 litres Rs. 1.20 per 100 litres
Next 100,000 litres Rs. 1.10 per 100 litres
Above this Re. 1.00 per 100 litres

The input record has customer identification, name and address, meter reading, past and present. Write a structured English procedure to obtain a bill for the customer.

```
While record left in customer file do
   Read customer id, name, address, past meter reading, new meter reading
   Consumption = new meter reading – old meter reading
   if consumption <= 500 then charge =10
      else if consumption <=800
         then charge = 10 + (consumption – 500) * 0.0125
         else
            if consumption <= 30800
               then charge = 13.75
               +(consumption – 800) * 0.012
```

```
                    else
                        if consumption <=130800
                        then charge = 373.75 + (consumption – 30800) * 0.011
                        else charge = 1473.75 + (consumption – 130800) *  0.01
                        end if
                end if
            end if
    end if
    write customer id, name, address, past meter reading, new meter reading,
    consumption, charge
end while
```

6.4 Refer to users requirements stated in exercise 6.4 of Question bank. Develop the processing rules in structured English for the stores process shown in Fig. 5.8.

```
for each customer requisition do
    if (qty. requested < qty. in stock ) then
        {
            qty. issued = qty. requested.
            qty. in stock = qty. in stock – qty. issued
            send( customer id , qty. issued ) to accounts process}
    end if
if (qty. in stock <= reorder level) then
    send reorder request to purchase.
end if

if (qty. requested = qty. in stock) then
    {
        qty. issued = qty. requested
        qty in stock = 0;
        send (customer id, qty issued) to accounts
        send reorder request to purchase }
end if

if (qty. requested > qty. in stock) then
    if (partial qty. acceptable)
        then {
                qty issued = qty. requested
                qty in stock = 0;
                send (customer id , qty issued) to accounts
                send reorder request to purchase.
                write (customer id, (qty. requested – qty. issued ))
                in file}
        else
            {qty. issued = 0;
                send reorder request to purchase
                write (customer id, qty requested) in back order file }
            end if
        end if
        end for

        for each item (with specified item code and qty. accepted note) received from
                inspection do
            qty. in stock ( item code)
            = qty. in stock ( item code) + qty. accepted
        end for
```

6.5 Refer to Exercise 6.4 in Question Bank. Develop the processing rules in structured English for the purchase process of Fig. 5.9.

```
for each discrepancy note received from the receiving office do
    {   intimate specified vendor
        enter discrepancy in the discrepancy file }
end for

for each goods rejected note received from the inspection office do
    {   intimate specified vendor
        enter rejected items note in items rejected file }
end for

for each reorder advice or new requisition received for an item from stores do
    { Refer to item file to find vendor details and order qty.
        send order to specified vendor
        enter order in order file }
end for
```

6.6. State the processing rules in structured English for the various processes in the data flow diagram for the hostel DFD described in Module 5

Mess Process

```
for each absence note do
    Read student charge record from student charge file (using student number
        as key)
        Add absence days to absence field
    Write student charge record in student charge file
end for
for each extras note do
    Read student charge record
    Append extras code and quantity to record
    Write student charge in student charge file
end for
```

Billing Process

```
While student charge records left in student charge file do
```

```
    Read student charge record from student charge file
    No. of days to charge = no. of days in a month – no. of days absent.
    Monthly charge = no. of days to charge * daily rate
        Total extras charge = 0
        for each extra item do
            extras charge (code) = extra qty. * charge (code)
        end for
    Add extra charge to total extra charge
        Amount to be billed = no. of days to charge * daily rate + total extras charge
        Student bill record = student charge record + daily rate, no. of days
            charged, monthly charge, extras charge (code) , total extras charge ,
            grand total to pay
        Write student bill record
end while
```

(*Note:* It is assumed that the daily rate, no. of days in a month and extras charge for each extras code is stored in the billing process as an internal data record.)

Payment Reconciliation Process

```
for each student Bill record do
    Store student Bill record in Bill file
end for
for each payment received do
        Read bill record of student from Bill file
            if amount paid = grand total to pay
            then balance due = 0
            else balance due = (grand total to pay – amount paid)
            Store bill record in unpaid bill file
        end if
end for
On payment due date, (due date + 10), (due date + 20) do
    While records left in unpaid Bill file do
        if balance due > 0
            then send overdue notice,
        end if
    end while
On (payment due date + 30) do
While records left in unpaid bill file do
    if balance due > 0
        then send message to warden
    end if
end while
```

6.7 The policy followed by a company to process customer orders is given by the following rules:

(i) If the customer order <= that in stock and his credit is OK, supply his requirement.

(ii) If the customer credit is not OK do not supply. Send him an intimation.

(iii) If the customer credit is OK but items in stock are less than his order, supply what is in stock. Enter balance to be sent in back-order file.

Obtain a decision table for above policy

Order <= stock	Y	Y	N	N
Credit OK	Y	N	Y	N
Supply order	X	–	–	–
Credit not OK Do not supply	–	X	–	X
Supply stock	–	–	X	–
Enter (order – stock) In back Order file	–	–	X	–

Observe that the last rule is not specified in the statement of rules. This has been added to complete the table.

6.8 Obtain a decision table to decide whether on a given date an employee in an organization has completed one year's service or not.

Let date of joining be DJ/MJ/YJ (Day/Month/Year)

Let today's date be DT/MT/YT (Day/Month/Year)

If (YT – YJ)	>1	= 1	= 1	E
If (MT – MJ)	–	>0	= 0	L
				S
If (DT – DJ)	–	–	>= 0	E
One year's service	Yes	Yes	Yes	No

6.9 Obtain a decision table corresponding to the structured English procedure given in worked example 6.4of decision table

```
for each customer requisition do decision table
```

decision table					
qty requested < qty in stock	Y	–	–	–	Y
qty requested = qty in stock	–	Y	–	–	–
qty requested > qty in stock	–	–	Y	Y	–
(qty in stock – qty requested) <= reorder level	Y	–	–	–	N
Partial order OK?	–	–	N	Y	–
qty issued = qty requested	X	X	–	–	X
qty issued = qty in stock	–	–	–	X	–
qty in stock = (qty in stock – qty issued)	X	–	–	–	X
qty in stock = 0	–	X	–	X	–
Send customer id,qty issued to account process	X	X	–	X	X
qty issued = 0	–	–	X	–	–
Send reorder request to purchase process	X	X	X	X	–
Write (customer id, qty requested – qty issued) in back order file	–	–	–	X	–
Write (customer id, qty requested) in back order file	–	–	X	–	–

```
for each item with specified item code and qty accepted note
    received from inspection do
    qty in stock ( item code) =
    qty in stock ( item code) + qty. accepted
end for
```

6.10 Obtain a decision table for an automatic stamp vending machine with the following specifications:

(i) To dispense 20, 15, 10, 5 paise stamps

(ii) To accept 50, 25, 10, 5 paise coins

(iii) Do not return change if it is necessary to return more than two coins of the same denomination. In such a case return the customer's coin and turn on "no change" light.

The machine should dispense a stamp, the right amount of change tendered, no stamp available, no change available, etc.

Table 1: Select Stamp

Amount tendered	< Stamp cost	= Stamp cost	> Stamp cost	–
Stamp requested available?	–	Y	Y	N
Amount insufficient	X	–	–	–
Return amount	X	–	–	X
‘No stamp’ light on	–	–	–	X
Dispense stamp	–	X	X	–
Go to Table 2	–	–	X	–
Stop	X	X	–	X

Table 2: Make Change

Amount tendered	50	50	50	50	50	50	50	25	25	25	25	25	25	10	
Stamp cost	20	20	15	15	10	5	5	20	15	15	10	5	5	5	E
25 ps. Available?	>=1	–	>=1	>=1	>=1	>=1	>=1	–	–	–	–	–	–	–	L
10 ps. Available?	–	>=2	>=1	–	>=1	>= 2	>=1	–	>=1	–	>=1	>=2	>=1	–	S
5 ps. Available	>=1	>=2	–	>=2	>=1	–	>=2	>=1	–	>=2	>=1	–	>=2	>=1	E
No of 25 ps. Returned	1	–	1	1	1	1	1	–	–	–	–	–	–	–	–
No of 10 ps. Returned	–	2	1	–	1	2	1	–	1	–	1	2	1	–	–
No of 5 ps. Returned	1	2	–	2	1	–	2	1	–	2	1	–	2	1	–
Dispense stamp	X	X	X	X	X	X	X	X	X	X	X	X	X	X	–
No. change light on	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X
Return amount	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X
Stop	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

6.11 Obtain the decision table to be used by a person to enter the office of a manager. The conditions to be checked are:

Door open? Ring sign on? Enter sign on? Door locked?

The actions a person takes are:

Ring bell, enter, wait, leave.

After obtaining the decision table, ensure that it has no logical errors and that it is in minimal form.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
C1: Door open?	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
C2: Ring sign on?	N	N	N	N	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y
C3: Enter sign on?	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y
C4: Door locked?	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y
Ring bell	–	–	–	–	X	–	X	–	–	?	–	?	X	?	X	?
Enter	–	–	X	–	–	–	X	–	–	?	X	?	–	?	X	?
Wait	–	–	–	–	X	–	–	–	–	?	–	?	X	?	–	?
Leave	X	X	–	X	–	X	–	X	X	?	–	?	–	?	–	?
	A1	A1	A2	A1	A3	A1	A4	A1	A1		A2		A3		A4	

Rules R10, R12, R14, R16 have conditions "Door open" and "Door locked" simultaneously true. These rules are thus impossible. Observe these are four distinct actions. The table is mapped on a K-map (Fig. S6.11)

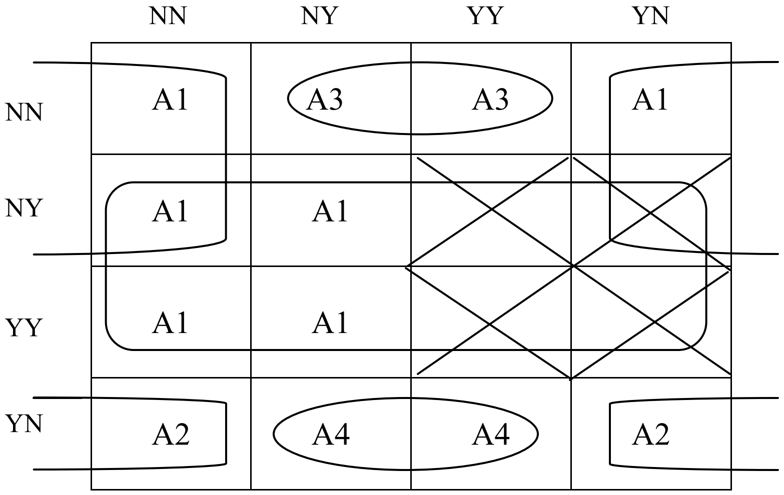


Fig S6.11 k-map for solution 6.11

Decision table with redundancies removed
Decision Table with removed (Condition C1 redundant)

C2: Ring on?	–	N	N	Y	Y
C3: Enter on?	–	N	Y	N	Y
C4: Door locked?	Y	–	N	N	N
Ring bell	–	–	–	X	X
Enter	–	–	X	–	X
Wait	–	–	–	X	–
Leave	X	X	–	–	–
	A1	A1	A2	A3	A4

6.12 In installment buying where payments are made on a weekly basis the action taken on "an account goes into arrears" is a crucial aspect of the operation. Table S6.12 illustrates a simplified arrears procedure. Answer the following questions:

(i) Are all elementary rules logically possible?
(ii) Is the table complete?
(iii) Has the table any ambiguities?
(iv) Are there logical errors in the table? If yes, point them out.
(v) Use reasonable assumptions and correct the table.
(vi) Remove any redundancies in the corrected table (Table S6.12).

Table S6.12 A Simplified Arrears Procedure

C1: This week's cash > weekly rate	Y	Y	N	N	–	–	–	–	–
C2: This week's cash > 0	–	–	Y	Y	–	–	N	N	N
C3: Any cash during last month	–	–	–	–	N	N	Y	Y	Y
C4: Arrears . >2 * weekly rate	–	–	Y	N	–	–	N	Y	–
C5: Arrears >4 * weekly rate	N	Y	N	–	N	Y	–	N	Y
Send arrears letter A	–	X	–	–	–	–	–	–	–
Send arrears letter B	–	–	X	–	–	–	–	–	–
Send arrears letter C	–	–	–	–	X	–	–	–	–
Send arrears letter D	–	–	–	–	–	–	–	X	–
Notify accounts	X	–	–	X	–	–	X	–	–
Take special action	–	–	–	–	–	X	–	–	X
	A1	A2	A3	A1	A4	A5	A1	A6	A5

K-map for the decision table

C1C2 C3C4	NN	NY	YY	YN
NN	A4	A1 A4	A1 A4	☒
NY	A4	A3 A4	A1 A4	☒
YY	A6	A3	A1	☒
YN	A1	A1	A1	☒

C5 = N

C1C2 C3C4	NN	NY	YY	YN
NN				
NY	A5	A5	A2 A5	
YY	A5	?	A2	
YN				

C5=Y

Fig S6.12

Assume weekly rate >0
Impossible rule:C1 = Y C2 = N

Impossible rule:C4 = N C5 = Y

K- map (Fig. S 9.10) has incompleteness marked by ?

C1	C2	C3	C4	C5
N	Y	Y	Y	Y

Logically contradictory rules

C1	C2	C3	C4	C5
N	Y	N	N	N
N	Y	N	Y	N
Y	Y	N	N	N
Y	Y	N	Y	N
Y	Y	N	Y	Y

If in all these cases we take action to notify accounts (Action A1) then the K- map becomes the one shown in Fig. S9.11

$$A1 = C3.\,\overline{C4}\; + C1\,.\,\overline{C5}\; + C2\,.\,\overline{C3}\,.\,\overline{C5} + C1\,.\,\overline{C3}\; + \overline{C1}\,.\,C2\,.\,C3\,.\,C5$$

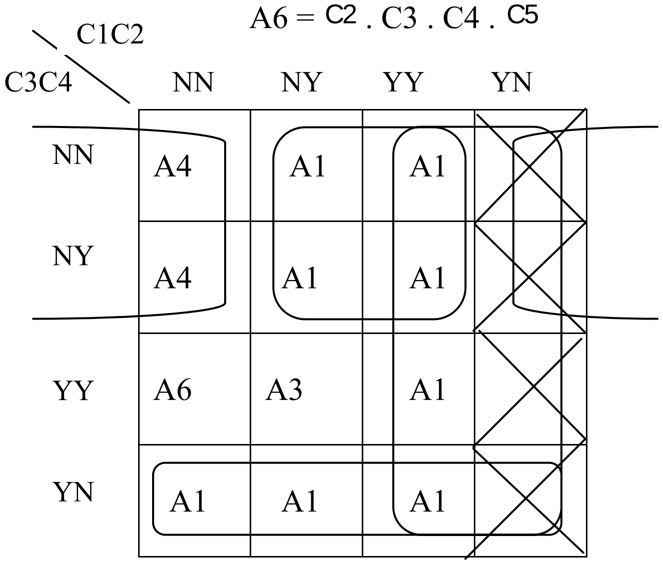
$$A2 = C1\,.\,C3\,.\,C5$$

$$A3 = \overline{C1}.\,C2\,.\,C3\,.\,C4\,.\,\overline{C5}$$

$$A4 = \overline{C2}\,.\,\overline{C3}\,.\,\overline{C5}$$

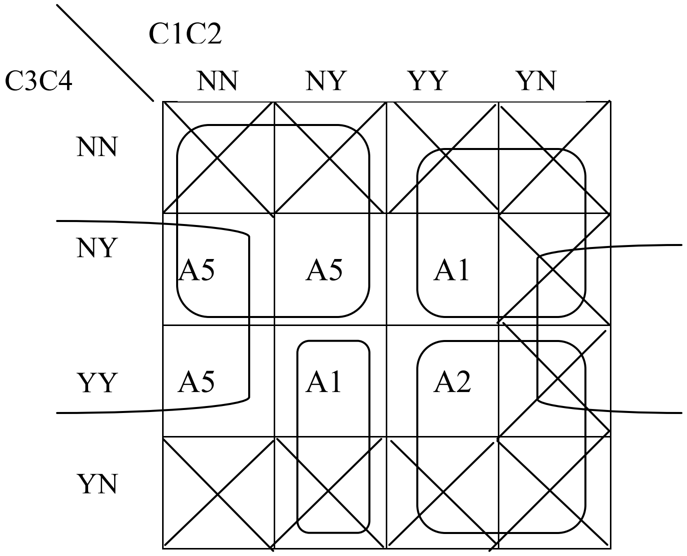
$$A5 = \overline{C1}\,.\,\overline{C3}\,.\,C5 + \overline{C2}\,.\,C4\,.\,C5$$

$$A6 = \overline{C2}\,.\,C3\,.\,C4\,.\,\overline{C5}$$



C3C4 \ C1C2	NN	NY	YY	YN
NN	A4	A1	A1	
NY	A4	A1	A1	
YY	A6	A3	A1	
YN	A1	A1	A1	

C5=N
Fig S.6.13



C5=Y
FigS6.14

C1: This week's cash > weekly rate	–	Y	–	Y	N	Y	N	–	N	–	–
C2: This week's cash > 0	–	–	Y	–	Y	–	Y	N	–	N	N
C3: Any cash during last month	Y	–	N	N	Y	Y	Y	N	N	–	Y
C4: Arrears . > 2 * weekly rate	N	–	–	–	–	–	Y	–	–	Y	Y
C5: Arrears > 4 * weekly rate	–	N	N	–	Y	Y	N	N	Y	Y	N
Send arrears letter A	–	–	–	–	–	X	–	–	–	–	–
Send arrears letter B	–	–	–	–	–	–	X	–	–	–	–
Send arrears letter C	–	–	–	–	–	–	–	X	–	–	–
Send arrears letter D	–	–	–	–	–	–	–	–	–	–	X
Notify accounts	X	X	X	X	X	–	–	–	–	–	–
Take special action	–	–	–	–	–	–	–	–	X	X	–

6.13 The policy followed by a company to process customer orders is given by the following rules:

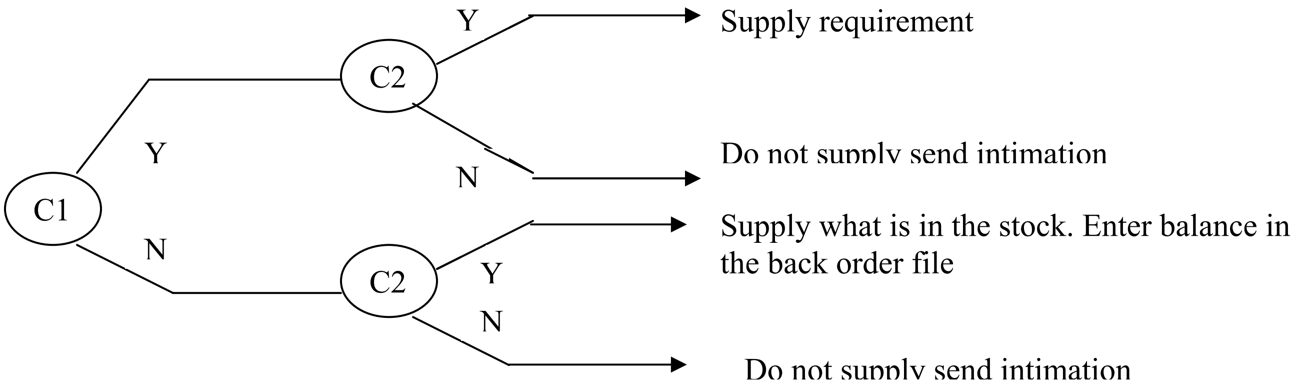
(i) If the customer order ≤ that in stock and his credit is OK, supply his requirement.

(ii) If the customer credit is not OK do not supply. Send him an intimation.

(iii) If the customer credit is OK but items in stock are less than his order, supply what is in stock. Enter the balance to be sent in a back-order file.

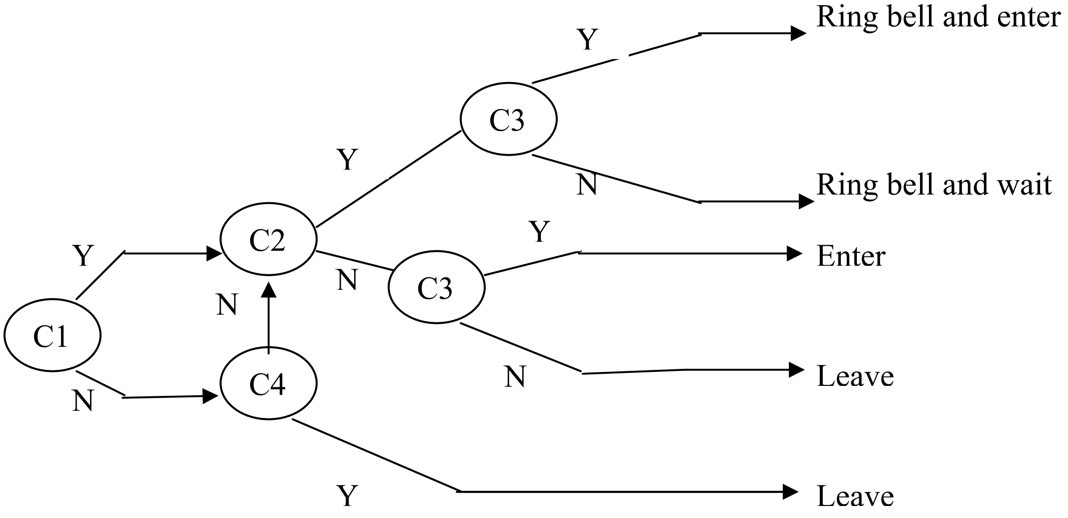
Obtain a decision table for the above policy.

C1 : Customer order <= Item in the stock
C2 : Customer credit ok


C1
Y
N
C2
C2
Y
N
Y
N
Supply requirement
Do not supply send intimation
Supply what is in the stock. Enter balance in the back order file
Do not supply send intimation

6.14 Obtain a decision tree to be used by a person to enter the office of a manager. The conditions to be checked are:
Door open? Ring sign on? Enter sign on? Door locked?
The actions a person takes are:
Ring bell, enter, wait, leave.
After obtaining the decision table, ensure that it has no logical errors and that it is in minimal form.

C1 : Door open
C2 : Ring sign on
C3 : Enter sign on
C4 : Door locked



Observe that C1 is not relevant and not clear in the Decision tree

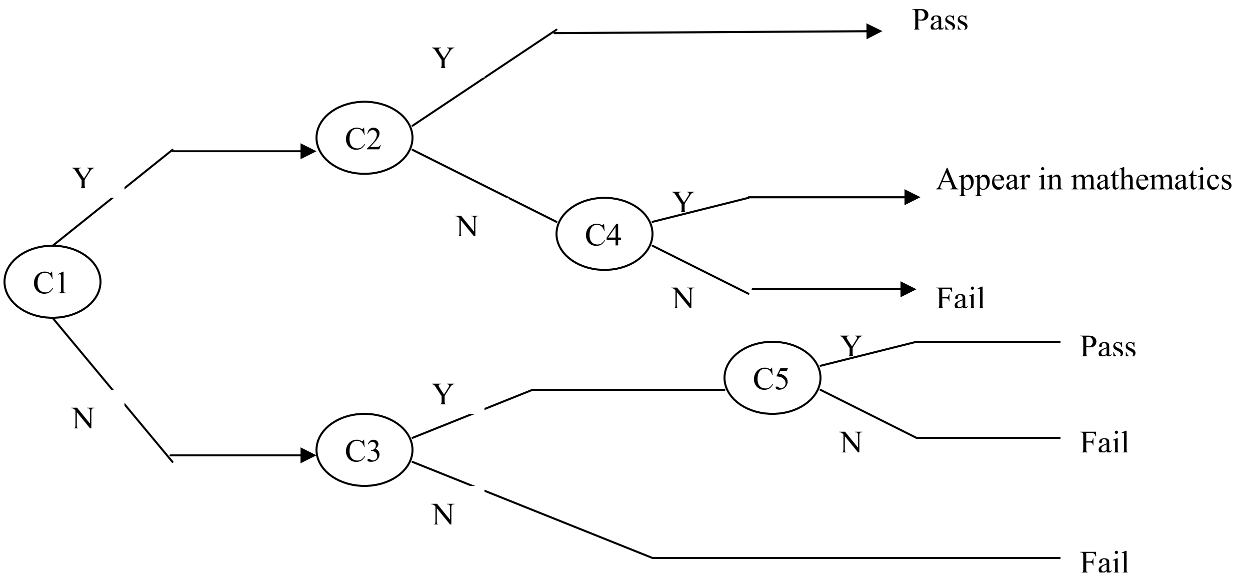
6.15 A University has the following rules for a student to qualify for a degree with Physics as the main subject and Mathematics as the subsidiary subject:

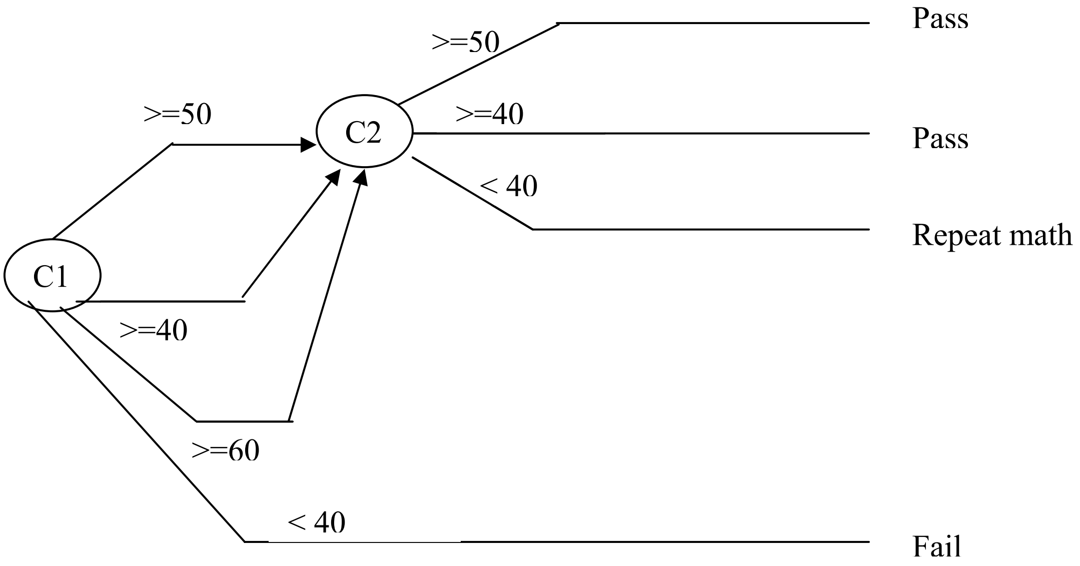
(i) he should get 50% or more marks in Physics and 40% or more marks in Mathematics.

(ii) If he gets < 50% marks in Physics, he should get 50% or more marks in Mathematics. He should, however, get at least 40% marks in Physics.

(iii) If he gets < 40% marks in Mathematics and 60% or more marks in Physics, he is allowed to reappear in Mathematics examination only so that he can qualify.

Obtain a decision tree for the above problem

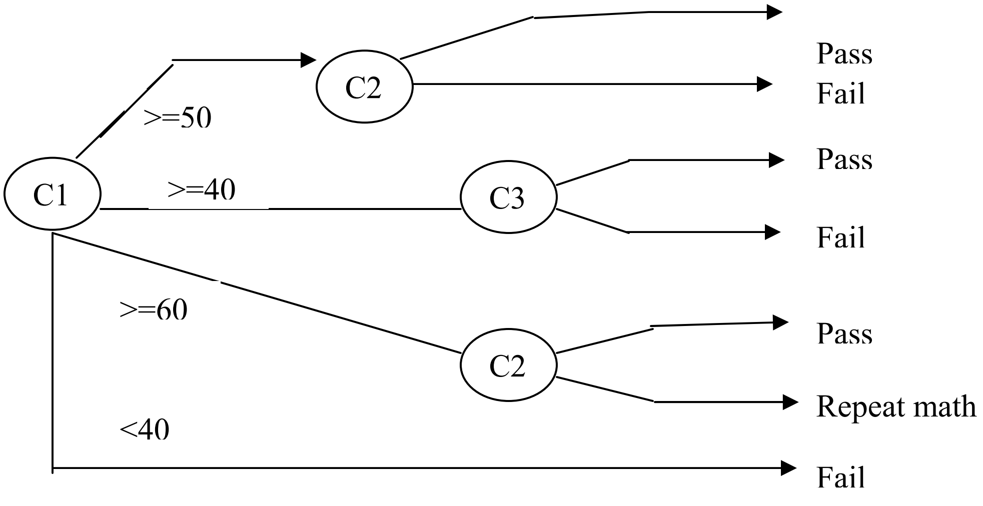




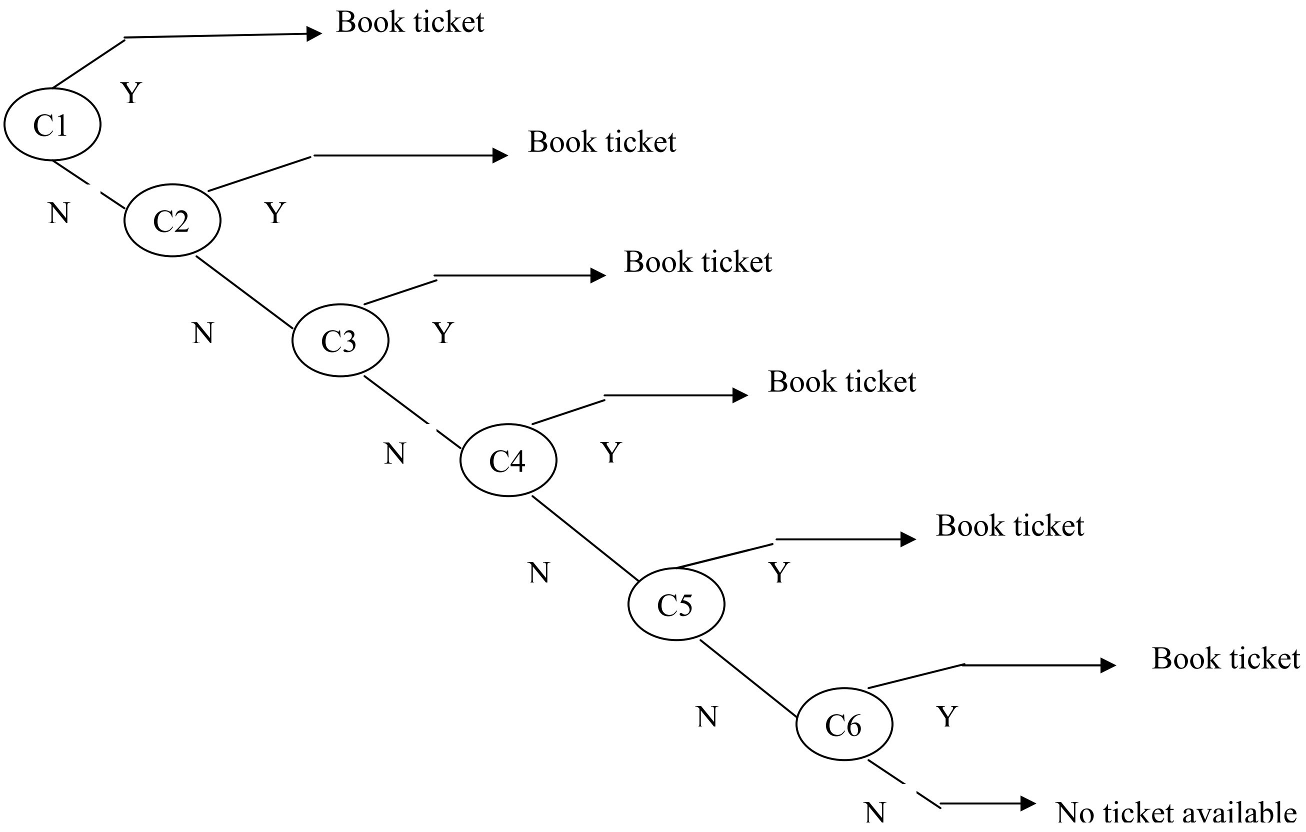
(An incorrect tree. This is not correct as rule ii not correctly interpreted)

C1 : Physics Marks
C2 : Math marks
C3 : Math marks >=50%

CORRECTED DECISION TREE



6.16 You want to go to Delhi from Bangalore. There are three flights per day; early morning, late morning and evening. You would like to go on 21.4.04 by early morning flight. If it is not available you will take the late morning flight or evening flight in that order. If neither is available you are willing to take any flight on 22.4.04 but prefer early and late morning flights., Obtain a decision tree for this word statement. Is decision table suitable for this problem? If not why?



C1 : Is ticket available on early morning flight on 21/4/04
C2 : Is ticket available on late morning flight on 21/4/04
C3 : Is ticket available on evening flight on 21/4/04
C4 : Is ticket available on early morning flight on 22/4/04
C5 : Is ticket available on late morning flight on 22/4/04
C6 : Is ticket available on evening flight on 22/4/04