

Scilab Textbook Companion for
Switching And Finite Automata Theory
by Z. Kohavi¹

Created by
Kota Sumanth Kumar
B.Tech (Pursuing)
Electronics Engineering
NIT Warangal
College Teacher
S. K. L. V. Sai Prakash, NIT Warangal
Cross-Checked by
Giridharan, IITB

July 31, 2019

¹Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website <http://scilab.in>

Book Description

Title: Switching And Finite Automata Theory

Author: Z. Kohavi

Publisher: Tata McGraw - Hill Education

Edition: 2

Year: 2008

ISBN: 0-07-099387-4

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes	4
1 Number System and Codes	5
2 Sets Relations and Lattices	18
3 Switching Algebra And Its Applications	31
4 Minimization Of Switching Functions	41
5 Logical Design	86
6 Functional Decomposition And Symmetric Functions	92
7 Threshold Logic	96
8 Reliable Design And Fault Diagnosis	102
10 Capabilities Minimization and transformation Of Sequential Machines	105
12 Structure Of Sequential Machines	108
13 State Identificaiton And Fault Detection Experiments	115

14 Memory Definiteness Information Losslessness of Finite Automata	117
15 Linear Sequential Machines	125
16 Finite State Recognizers	134

List of Scilab Codes

Exa 1.1	converts no to base 10	5
Exa 1.2	converts binary number to base 10	6
Exa 1.3	convert decimal number to base 8	8
Exa 1.4	Convert decimal value in base 10 to base 8	9
Exa 1.5	convert decimal number to base 2	10
Exa 1.6	Convert Octal number to Base 2	11
Exa 1.7	Adds two binary numbers	14
Exa 1.8	Subtracts Two Binary numbers	15
Exa 1.9	Multiplies two Binary numbers	16
Exa 1.10	Division of Two Binary Numbers	16
Exa 2.1	Set Of All even Numbers	18
Exa 2.2	Subsets Of a Faces of a die	19
Exa 2.4	Relation between two sets	21
Exa 2.5	Equivalence relation	22
Exa 2.6	Equivalence relation 2	24
Exa 2.7	Function Check	25
Exa 2.11	Partial Ordering Divisibility Relation	26
Exa 2.12	Ordering Relation	27
Exa 2.15	Lattice Of subsets	28
Exa 2.16	glb and ulb	29
Exa 3.1	Simplify 1	31
Exa 3.2	Simplify 2	31
Exa 3.3	Prove the identity	32
Exa 3.4	Determine The Output of expression	33
Exa 3.5	Simplify 3	33
Exa 3.6	Simplify 4	34
Exa 3.7	Expand the Expression	34
Exa 3.8	Expand the Expression 2	35

Exa 3.9	POS	36
Exa 3.10	Tabulate the Function of 2 variables	36
Exa 3.11	NOR	37
Exa 3.12	Transmission function	38
Exa 3.13	Air Conditioning System	38
Exa 3.14	DeMorgans Law	39
Exa 4.1	Irredundant expressions	41
Exa 4.2	Irredundant expressions 2	49
Exa 4.3	Reduce Expression	57
Exa 4.4	BCD to Excess 3 Convertor	65
Exa 4.5	5 variable Kmap	66
Exa 4.6	Prime Implicants	83
Exa 4.7	Prime Implicants 2	84
Exa 4.8	Prime Implicants Of a Function	84
Exa 4.9	Prime Implicants Of a Function 2	85
Exa 4.10	Cyclic Prime Implicant map	85
Exa 5.1	Odd Parity Bit Generator	86
Exa 5.2	Serial To Parallel converter	87
Exa 5.3	Transmission function for a network	88
Exa 5.4	4 Input Contact Network	89
Exa 5.5	Minimal contact Network	90
Exa 6.1	Function Decomposition	92
Exa 6.3	Multiplicity	92
Exa 6.6	Symmetric	94
Exa 6.7	Symmetric 2	94
Exa 7.1	weighted Sum	96
Exa 7.2	Inequalities	97
Exa 7.3	unate Functions	99
Exa 7.4	three cube representation	99
Exa 7.5	True Vertex	100
Exa 8.1	NOR Logic Circuit	102
Exa 8.2	Path Sensitizing	103
Exa 8.3	Two level OR AND Network	103
Exa 8.4	Various errors Summarized	104
Exa 10.1	Reducing machine	105
Exa 10.2	Compatibility Graph	106
Exa 12.4	Closed Partitions	108
Exa 12.5	Output Consistent	109

Exa 12.7	Dependence	110
Exa 12.8	input Consistent Matrix	110
Exa 12.9	Implementation of Input Consistent matrix .	111
Exa 12.10	Autonomous Clock	111
Exa 12.11	partition pair	112
Exa 12.12	partition pair 2	112
Exa 12.14	State Consistent Partitions	113
Exa 13.1	Fault DetectionExeriment	115
Exa 14.2	Synchronizing tree	117
Exa 14.3	Contracted State Machine	118
Exa 14.4	Testing Table	121
Exa 14.6	Testing graph	122
Exa 16.7	Testing Table 2	123
Exa 15.3	Transfer function	125
Exa 15.4	Impulse response	126
Exa 15.5	Null Sequence	127
Exa 15.6	Null Sequence 2	127
Exa 15.10	Reducable	128
Exa 15.11	Linear Machine	129
Exa 15.12	Linear Machine 2	130
Exa 15.15	Commutative ring	131
Exa 15.16	Identifying Whether the Ring is a Field Or not	131
Exa 15.17	Finite Field	132
Exa 16.2	Prove the identity	134
Exa 16.3	Prove the identity 2	134

Chapter 1

Number System and Codes

Scilab code Exa 1.1 converts no to base 10

```
1  clc //clears the command window
2  clear //clears all the variables
3
4  i=1;w=1; //flags
5
6  bin=432.2 //given octal number which should be
   expressed in base 10
7  temp1=floor(bin); //separating integer part from the
   given number
8  temp2=modulo(bin,1); //separating decimal part from
   the given number
9  temp2=temp2*10^3; //converting decimal value to
   interger for convenience
10 while(temp1>0) //storing each integer digit in vector
   for convenience
11     p(i)=modulo(temp1,10);
12     temp1=round(temp1/10);
13     i=i+1;
14 end
15 while(temp2>0) //storing each decimal digit in vector
   for convenience
```

```

16     q(w)=modulo(temp2,10);
17     temp2=(temp2/10);
18     temp2=round(temp2);
19     w=w+1;
20 end
21 temp1=0;//clearing temporary variable 'temp1'
22 for i=1:length(p)//checking whether it is a binary
    number or not
23     if(p(i)>8) then
24         disp('not a binary number');
25         abort;
26     end
27 end
28
29 for i=1:length(p)//multiplying the bits of integer
    part with their position values and adding
30     temp1=temp1+(p(i)*8^(i-1));
31 end
32
33 temp2=0;//flag bit
34
35 for z=1:length(q)
36 //multiplying the bits of decimal part with their
    position values and adding
37     temp2=temp2+(q(z)*8^(-1*(4-z)));
38 end
39 temp=temp1+temp2;
40 //finally adding both the integer and decimal parts
    to get total output.
41 disp(temp);//displaying the output

```

Scilab code Exa 1.2 converts binary number to base 10

```

1 clc;//clears the command window
2 clear;//clears all the variables

```

```

3 i=1;w=1;
4 bin=1101.01; //Given binary number which we need to
   be convert into decimal
5 temp1=floor(bin); //separating integer part from the
   given number
6 temp2=modulo(bin,1); //separating decimal part from
   the given number
7 temp2=temp2*10^3; //converting decimal value to
   interger for convenience
8 while(temp1>0) //storing each integer digit in vector
   for convenience
9     p(i)=modulo(temp1,10);
10    temp1=floor(temp1/10);
11    i=i+1;
12 end
13 while(temp2>0) //storing each decimal digit in vector
   for convenience
14    q(w)=modulo(temp2,2);
15    temp2=(temp2/10);
16    temp2=floor(temp2);
17    w=w+1;
18 end
19 temp1=0; //flag bit
20 for i=1:length(p) //checking whether it is a binary
   number or not
21     if(p(i)>1) then
22         disp('not a binary number');
23         abort;
24     end
25 end
26 for i=1:length(p)
27 //multiplying bits of integer part with their
   position values and adding
28     temp1=temp1+(p(i)*2^(i-1));
29 end
30 temp2=0; //flag bit
31 for z=1:length(q)
32 //multiplying bits of decimal part with their

```

```

    position values and adding
33     temp2=temp2+(q(z)*2^(-1*(4-z)));
34 end
35 temp=temp1+temp2;
36 //finally adding both the integer and decimal parts
    to get total output.
37 disp(temp); //displaying the output

```

Scilab code Exa 1.3 convert decimal number to base 8

```

1  clc; //clears the command window
2  clear; //clears all the variables
3  format('v',8); //making the default precision to 8
    significant digits
4  i=1;
5  dec=548; //given decimal number which should be
    expressed in base 8
6  temp=dec;
7  i=1;
8  d=8;
9  while(temp>0) //storing each integer digit in vector
    for convenience
10     p(i)=(modulo(floor(temp),d))
11     temp=floor(temp)/d;
12     i=i+1;
13 end
14 temp2=0;
15 for j=1:length(p)
16 //multiplying bits of integer part with their
    position values and adding
17     temp2=temp2+(p(j)*10^(j-1));
18 end
19 disp(temp2," Octal number");
20 dec=345;
21 //given decimal number which should be expressed in

```

```

        base 8
22 temp=dec;
23 i=1;
24 d=6;
25 while(temp>0)//storing each integer digit in vector
    for convenience
26     p(i)=(modulo(floor(temp),d))
27     temp=floor(temp)/d;
28     i=i+1;
29 end
30 temp2=0;
31 for j=1:length(p)
32 //multiplying bits of integer part with their
    position values and adding
33     temp2=temp2+(p(j)*10^(j-1));
34 end
35 disp(temp2," Base 6");

```

Scilab code Exa 1.4 Convert decimal value in base 10 to base 8

```

1  clc;//clears the command window
2  clear;//clears all the variables
3  format('v',8);//making the default precision to 8
    significant digits
4  i=1;
5  dec=0.3125;//given decimal number which should be
    expressed in base 8
6  temp=modulo(0.3125,1);//separating decimal part from
    the given number
7
8  while(temp~=0) //storing each decimal digit in
    vector for convenience
9     temp=temp*8;
10    p(i)=floor(temp);
11    i=i+1;

```

```

12     temp=modulo(temp,1);
13 end
14
15 temp1=0; //flag bit
16 for j=1:length(p)
17 //multiplying bits of decimal part with their
    position values and adding
18     temp1=temp1+(10^(-1*j)*p(j))
19 end
20 disp(temp1); //displays the final output

```

Scilab code Exa 1.5 convert decimal number to base 2

```

1  clc; //clears the command window
2  clear; //clears all the variables
3  format('v',18); //changing the default precision to 20
    significant digits
4
5  i=1;x=1; //flag bits
6
7  dec=432.354; //given decimal number which should be
    expressed in base 2
8  temp2=floor(dec); //separating integer part from the
    given number
9  temp4=modulo(dec,1); //separating decimal part from
    the given number
10
11 while(temp2>0) //storing each integer digit in vector
    for convenience
12     p(i)=(modulo(floor(temp2),2))
13     temp2=floor(temp2)/2;
14     i=i+1;
15 end
16
17 temp2=0; //clearing temporary variable 'temp2'

```

```

18
19 for j=1:length(p)
20 //multiplying bits of integer part with their
    position values and adding
21     temp2=temp2+(p(j)*10^(j-1));
22 end
23
24 while(temp4~=0) //storing each decimal digit in
    vector for convenience
25     temp4=temp4*2;
26     d(x)=floor(temp4);
27     x=x+1;
28     temp4=modulo(temp4,1);
29 end
30
31 temp5=0;           //clearing temporary variable '
    temp5'
32
33 for j=1:length(d)
34 //multiplying bits of decimal part with their
    position values and adding
35     temp5=temp5+(10^(-1*j)*d(j))
36 end
37
38 temp3=temp2+temp5;
39 //finally adding both the integer and decimal parts
    to get total output.
40 disp(temp3); //displays output

```

Scilab code Exa 1.6 Convert Octal number to Base 2

```

1 clc; //clears the command window
2 clear; //clears all the variables
3 format('v',8); //setting the default precision to 8
4

```

```

5 i=1;w=1;
6
7 bin=123.4; //Given octal number which we need to be
      convert into binary
8 temp1=floor(bin); //separating integer part from the
      given number
9 temp2=modulo(bin,1); //separating decimal part from
      the given number
10 temp2=temp2*10^3; //converting decimal value to
      interger for convenience
11
12 while(temp1>0) //storing each integer digit in
      vector for convenience
13     p(i)=modulo(temp1,10);
14     temp1=round(temp1/10);
15     i=i+1;
16 end
17
18 while(temp2>0) //storing each decimal digit in
      vector for convenience
19     q(w)=modulo(temp2,10);
20     temp2=(temp2/10);
21     temp2=round(temp2);
22     w=w+1;
23 end
24
25 temp1=0; //clearing temporary variable 'temp1'
26
27 for i=1:length(p) //checking whether it is a binary
      number or not
28     if(p(i)>8) then
29         disp('not a binary number');
30         abort;
31     end
32 end
33
34 for i=1:length(p)
35 //multipliying bits of decimal part with their

```



```

        position values and adding
36     temp1=temp1+(p(i)*8^(i-1));
37 end
38
39 temp2=0;//clearing temporary variable 'temp2'
40
41 for z=1:length(q)
42 //multiplying bits of decimal part with their
    position values and adding
43     temp2=temp2+(q(z)*8^(-1*(4-z)));
44 end
45
46 temp=temp1+temp2;
47 //adding both integer and decimal parts to get total
    deciaml value.
48 dec=temp;
49
50 temp2=floor(dec); //separating integer part from the
    given number
51 temp3=modulo(dec,1);//separating decimal part from
    the given number
52 format('v',18);//setting the default precision to 8
53
54 i=1;x=1;//flag bits
55
56 while(temp2>0)//storing each integer digit in vector
    for convenience
57     p(i)=(modulo(floor(temp2),2))
58     temp2=floor(temp2)/2;
59     i=i+1;
60 end
61
62 temp2=0; //clears temporary variable 'temp2'
63
64 for j=1:length(p)
65 //multiplying bits of integer part with their
    position values and adding
66     temp2=temp2+(p(j)*10^(j-1));

```

```

67 end
68
69 temp4=modulo(temp3,1);
70
71 while(temp4~=0) //storing each decimal digit in
    vector for convenience
72     temp4=temp4*2;
73     d(x)=floor(temp4);
74     x=x+1;
75     temp4=modulo(temp4,1);
76 end
77
78 temp5=0; //clears temporary variable 'temp2'
79
80 for j=1:length(d)
81 //multiplying bits of decimal part with their
    position values and adding
82     temp5=temp5+(10^(-1*j)*d(j))
83 end
84
85 temp=temp2+temp5;
86 //finally adding both the integer and decimal parts
    to get total output.
87 disp(temp); //displaying the output

```

check Appendix [AP 9](#) for dependency:

bin21dec.sci

check Appendix [AP 6](#) for dependency:

dec21bin.sci

Scilab code Exa 1.7 Adds two binary numbers

```

1 clc;
2 a=0111.10;           // first number

```

```

3 b=1111.01;           //second number
4 A=bin21dec(a);      //converting a in to decimal number
5 B=bin21dec(b);      //converting b in to decimal number
6 S=A+B;              //adding the two decimal numbers
7 temp=dec21bin(S);   //converting the decimal sum back
    to binary
8 format('v',10);     //changing the default precision
    to 8
9 disp(temp);         //displaying the final output

```

check Appendix [AP 9](#) for dependency:

bin21dec.sci

check Appendix [AP 6](#) for dependency:

dec21bin.sci

Scilab code Exa 1.8 Subtracts Two Binary numbers

```

1 clc;
2 format('v',8);      //changing the default precision to
    8
3 a=10010.11;        //first number
4 b=01100.10;        //second number
5 A=bin21dec(a);     //converting a in to decimal number
6 B=bin21dec(b);     //converting b in to decimal number
7 S=A-B;             //subtracting the two decimal
    numbers
8 temp=dec21bin(S);   //converting the decimal number
    back to binary
9 disp(temp);         //displaying the final output

```

check Appendix [AP 9](#) for dependency:

bin21dec.sci

check Appendix [AP 6](#) for dependency:

dec21bin.sci

Scilab code Exa 1.9 Multiplies two Binary numbers

```
1 clc ;
2 format('v',8); //changing the default precision to
   8
3 a=11001.1; //first number
4 b=110.1; //second number
5 A=bin21dec(a); //converting a in to decimal number
6 B=bin21dec(b); //converting b in to decimal number
7 S=A*B; //multiply the two decimal numbers
8 temp=dec21bin(S); //converting the decimal product
   back to binary
9 disp(temp); //displaying the final output
```

check Appendix [AP 9](#) for dependency:

bin21dec.sci

check Appendix [AP 6](#) for dependency:

dec21bin.sci

Scilab code Exa 1.10 Division of Two Binary Numbers

```
1 clc ;
2 format('v',8); //changing the default precision to 8
3 a=1000100110; //first number
4 b=11001; //second number
5 A=bin21dec(a); //converting a in to decimal number
6 B=bin21dec(b); //converting b in to decimal number
7 S=A/B; //multiply the two decimal numbers
8 temp=dec21bin(S); //converting the decimal product
   back to binary
```

```
9 disp("quotient");  
10 disp(temp); //displaying the final output
```

Chapter 2

Sets Relations and Lattices

Scilab code Exa 2.1 Set Of All even Numbers

```
1 clear;
2 clc;
3 //lower=input("input the lower limit of the set");
4 //upper=input("input the upper limit of the set");
5 lower=1; //lower limit of the set
6 upper=10; //upper limit of the set
7 temp=lower;
8 h=1;i=1;
9 while(temp<=upper)
10     if(modulo(temp,2)==0) //checking whether a
11         e(h)=temp;
12         h=h+1;
13     else
14         o(i)=temp; //odd number
15         i=i+1;
16     end
17     temp=temp+1;
18 end;
19 disp("the set of even number between the limits")
20 disp(e);
```

```
21 disp("the set of odd number between the limits")
22 disp(o);
```

Scilab code Exa 2.2 Subsets Of a Faces of a die

```
1 clear;
2 clc;
3 //N=input("enter the elements in the set");
4 //n=1;
5 //while(n<=N)
6 //     i(n)=input("enter the elements of first set");
7 //     n=n+1;
8 //end;
9 N=6;
10 i(1)='f1';i(2)='f2';i(3)='f3';i(4)='f4';i(5)='f5';i
    (6)='f6';
11 disp("null set"); //since null set is subset of
    any set.
12 for a=1:1:N //set of single elements
13     disp(i(a));
14 end
15 c=1;
16 for a=1:1:N-1 //set of two elements
17     for b=a:1:N
18         if(a~=b)
19             m(c,1)=i(a);
20             m(c,2)=i(b);
21             c=c+1;
22         end
23     end;
24 end;
25 disp(m);
26 d=1;
27 for a=1:1:N //set of three elements
28     for b=a:1:N
```

```

29         for c=b:1:N
30             if((a~=b)&(b~=c)&(c~=a))
31                 p(d,1)=i(a);
32                 p(d,2)=i(b);
33                 p(d,3)=i(c);
34                 d=d+1;
35             end
36         end
37     end
38 end
39 disp(p);
40 e=1;
41 for a=1:1:N //set of four elements
42     for b=a:1:N
43         for c=b:1:N
44             for d=c:1:N
45                 if((a~=b)&(b~=c)&(c~=d)&(d~=a)&(b~=d)
46                    )&(a~=c))
47                     q(e,1)=i(a);
48                     q(e,2)=i(b);
49                     q(e,3)=i(c);
50                     q(e,4)=i(d);
51                     e=e+1;
52                 end
53             end
54         end
55     end
56 disp(q);
57 f=1;
58 for a=1:1:N //set of five elements
59     for b=a:1:N
60         for c=b:1:N
61             for d=c:1:N
62                 for e=d:1:N
63                     if((a~=b)&(b~=c)&(c~=d)&(d~=e)&(
64                        e~=a)&(a~=c)&(a~=d)&(b~=d)&(c
65                        ~=e)&(b~=e))

```



```

64             r(f,1)=i(a);
65             r(f,2)=i(b);
66             r(f,3)=i(c);
67             r(f,4)=i(d);
68             r(f,5)=i(e);
69             f=f+1;
70             end
71         end
72     end
73 end
74 end
75 end
76 disp(r);
77 for a=1:1:N           //set of six elements
78     s(1,1)=i(1);
79     s(1,2)=i(2);
80     s(1,3)=i(3);
81     s(1,4)=i(4);
82     s(1,5)=i(5);
83     s(1,6)=i(6);
84 end
85 disp(s);

```

Scilab code Exa 2.4 Relation between two sets

```

1  clear;
2  clc;
3  N1=2;
4  N2=3;
5  //n=1;
6  //while(n<=N1)
7  //     i(n)=input("enter the elements of first set");
8  //     n=n+1;
9  //end;
10 i=['p' 'q']; //set A

```

```

11 //n=1;
12 //while(n<=N2)
13 //    j(n)=input("enter the elements of second set")
14 //    ;
15 //    n=n+1;
16 //end;
17 j=['r' 's' 't']; //set B
18 c=1;d=1;
19 for a=1:1:N1 //realtion between sets A and B
20     for b=1:1:N2
21         m(c,d)=i(a);
22         m(c,d+1)=j(b);
23         c=c+1;
24     end;
25 end;
26 disp(m);

```

Scilab code Exa 2.5 Equivalence relation

```

1 clear;
2 clc;
3 //N=input("enter the no of elements in the set :");
4 //for i=1:1:N
5 //    s(1,i)=input("enter the elements in the set
6 //    :");
7 //end;
8 //n=input("enter the number of pairs in the relation
9 //    :");
10 //for j=1:1:n
11 //    for k=1:1:2
12 //        r(j,k)=input("enter the elements in the
13 //        relation :");
14 //    end
15 //end
16 N=2;

```

```

14 s=['a' 'b'];           //elements in the set
15 n=3;
16 r=['a' 'a';'b' 'b';'a' 'b']; //realtion between
    the elements in the above set.
17 ref=zeros(1,N);
18 for a=1:1:N
19     for b=1:1:n
20         if(r(b,1)==s(1,a)&r(b,2)==s(1,a))
21             ref(1,a)=1;
22         end
23     end
24 end
25 for i=1:1:N           //checking whether above relation
    is reflexive or not
26     if(ref(1,i)==1)
27         disp("the above relation is reflexive with
                elements ");
28         disp(s(1,i));
29         disp(" ");
30     end
31 end
32 sym=zeros(1,(N*N-1)/2);
33 s(1,N+1)=s(1,1);
34 for a=1:1:N           //checking whether above
    relation is symmetric or not
35     for b=1:1:n
36         if(r(b,1)==s(1,a)&r(b,2)==s(1,a+1))
37             for d=1:1:n
38                 if(r(d,1)==s(1,a+1)&r(d,2)==s(1,a))
39                     sym(1,a)=1;
40                     disp("the above relation is
                            symmetric for these pairs :");
41                     disp(")",s(1,a+1),s(1,a),"(");
42                 end
43             end
44         end
45     end
46 end

```

```

47 for a=1:1:n           //checking whether it is
    transtive or not.
48     u=r(a,1);
49     v=r(a,2);
50     for b=a:1:n
51         if(r(b,1)==v)
52             w=r(b,2);
53             for c=b:1:n
54                 if(r(c,1)==w&r(c,2)==u)
55                     disp("satisfies
                            transtitive property")
                            ;
56                         abort;
57                     end
58                 end
59             end
60         end
61 end

```

Scilab code Exa 2.6 Equivalence relation 2

```

1 clear;
2 clc;
3 //N=input("enter the elements in the set :");
4 //for i=1:1:N
5 //     s(i)=input("enter the elements :");
6 //end
7 N=2;
8 s=['a' 'b'];
9 disp("The equivalence relation of above set is :");
10 h=1;
11 for i=1:1:N           //to satisfy reflexive property
12     ref(h,1)=s(i);
13     ref(h,2)=s(i);
14     h=h+1;

```

```

15 end
16 for i=1:1:N           //to satisfy symmetric property
17     for j=i:1:N
18         if(i~=j)
19             ref(h,1)=s(i);
20             ref(h,2)=s(j);
21             h=h+1;
22         end
23     end
24 end
25 m=1;
26 for i=1:max(size(ref(:,1)))           //to satisfy
    transtive property
27     if(ref(i,1)~=ref(i,2))
28         ref1(m,1)=ref(i,2);
29         ref1(m,2)=ref(i,1);
30         m=m+1;
31     end
32 end
33 disp(ref);
34 disp(ref1);

```

Scilab code Exa 2.7 Function Check

```

1 clc;
2 clear;
3 //n1=input("enter the no of elements in the 1st set
    ");
4 //for i=1:1:n1
5 //     s1(i)=input("enter the elements of 1st set:");
6 //end
7 n1=3;
8 s1=['a1' 'a2' 'a3'];           //set A
9 //n2=input("enter the no of elements in the 2nd set
    ");

```

```

10 //for i=1:1:n2
11 //    s2(i)=input("enter the elements of 2ns set:");
12 //end
13 n2=2;
14 s1=['b1' 'b2'];           //set B
15 //N=input("enter the pairs in the relation which you
    want to check whether is a function");
16 //for i=1:1:N
17 //    for j=1:1:2
18 //        r(i,j)=input("enter the elements in the
    relation:");
19 //    end
20 //end
21 N=3;
22 r=['a1' 'b1'; 'a2' 'b2'; 'a2' 'b1'];           //Realtion
    r
23 for i=1:1:N           //checks whether the relation
    is function or not
24     for j=i:1:N
25         if(r(i,1)==r(j,1) & i~=j)
26             disp("the relation is not a function");
27             abort;
28         end
29     end
30 end
31 disp("the realtion is a fuction");

```

Scilab code Exa 2.11 Partial Ordering Divisibilty Relation

```

1 clear;
2 clc;
3 //N=input("enter the number for which divisibilty ")
    ;
4 N=45;           //divident
5 h=1;

```

```

6 for i=1:N           //finding all the divisors of 45
7     if(modulo(N,i)==0)
8         r(h)=i;
9         h=h+1;
10    end
11 end
12 n=max(size(r));
13 disp("Hasse Diagram")           //displaying in the form
    of hasse diagram
14 disp("_____")
15 disp(r(n));
16 h=2;
17 for i=n-1:-2:3
18     disp("_____");
19     disp(r(i),r(i-1));
20     h=h+1;
21 end
22 disp("_____")
23 disp(r(1));

```

Scilab code Exa 2.12 Ordering Relation

```

1 clear
2 clc
3 N=4;
4 //defining all the partial ordered sets
5 s(1,:)= [1 0];
6 s(2,:)= [0 1];
7 s(3,:)= [0 0];
8 s(4,:)= [1 1];
9 //Finding
10 a=1;b=1;
11 for i=1:1:N           //sorting based on the level
12     for j=i:1:N
13         if(i~=j)

```

```

14         u=s(i,1)+s(i,2);
15         v=s(j,1)+s(j,2);
16         if(u<v)
17             temp(1)=s(i,1);
18             temp(2)=s(i,2);
19             s(i,1)=s(j,1);
20             s(i,2)=s(j,2);
21             s(j,1)=temp(1);
22             s(j,2)=temp(2);
23         end
24     end
25 end
26 end
27 // displaying in the form of hasse graph form
28 disp("1st stage of Hasse diagram");
29 disp(s(1,:));
30 disp("2nd stage of Hasse diagram");
31 disp(s(2,:));
32 disp(s(3,:));
33 disp("3rd stage of Hasse diagram");
34 disp(s(4,:));

```

Scilab code Exa 2.15 Lattice Of subsets

```

1 clear
2 clc
3 N=3;
4 s=['a' 'b' 'c']; //set for which lattice
                    should be defined
5 for i=2:2^3
6     s(i,:)=s(1,:);
7 end
8 //defining 2nd level vertices of the lattice
9 for i=2:4
10    s(i,i-1)='0';

```



```

11 end
12 //defining 3rd level vertices of the lattice
13 for i=5:6
14     s(i,i-4)='0';
15     s(i,i-3)='0';
16 end
17 s(7,1)='0';s(7,3)='0';
18 //defining the final level of vertices of the
    lattice
19 s(8,:)=['0' '0' '0'];
20 disp("1st level");
21 disp(s(1,:));
22 disp("2nd level");
23 disp(s(2,:));
24 disp(s(3,:));
25 disp(s(4,:));
26 disp("3rd level");
27 disp(s(5,:));
28 disp(s(6,:));
29 disp(s(7,:));
30 disp("4th level");
31 disp(s(8,:));

```

Scilab code Exa 2.16 glb and ulb

```

1 clear;
2 clc;
3 par1=['ab','cde','fh','gi']; //initial partitions
    par1
4 par2=['abc','de','fg','hi']; //partition 2
5 //par=par1+par2;
6 //lub-lower upper bound
7 par_lub=['abcde','fghi'];
8 disp(par_lub);
9 //par=par1.par2

```

```
10 //glb-greatest lower bound
11 par_glb=['ab','c','de','f','g','h','i'];
12 disp(par_glb);
```

Chapter 3

Switching Algebra And Its Applications

Scilab code Exa 3.1 Simplify 1

```
1 clear
2 clc;
3 disp("T(x,y,z)=x^y^z+yz+xz");
4 disp("**Minimise the given expression**");
5 disp("**Since z is common in every term taking z
   common**");
6 disp("T(x,y,z)=z(x^y^+y+x)");
7 disp("**From the property a+a^b=a+b **");
8 disp("T(x,y,z)=z(x^+y+x)");
9 disp("**Since we know that a+a^=1 **");
10 disp("T(x,y,z)=z(1+y)");
11 disp("**we know that 1+a=1 **");
12 disp("T(x,y,z)=z.1");
13 disp("T(x,y,z)=z");
```

Scilab code Exa 3.2 Simplify 2

```

1 clear
2 clc
3 disp("T(x,y,z)=(x+y)[x^(y+z)]^x^y+x^z");
4 disp("From the properties 1. (ab)^=a^+b^ 2. (a+b)^=a
    ^b");
5 disp("T(x,y,z)=((x+y)(x+yz))+x^y+x^z");
6 disp("Multiplying the first 2 terms");
7 disp("T(x,y,z)=(x+xyz+xy+yz)+x^y+x^z");
8 disp("T(x,y,z)=(x(1+y+yz)+yz)+x^y+x^z");
9 disp("T(x,y,z)=x+yz+x^y+x^z");
10 disp("we know a+a^b=a+b");
11 disp("T(x,y,z)=x+y+yz+x^z");
12 disp("T(x,y,z)=x+z+y+yz");
13 disp("T(x,y,z)=x+z+y+z");
14 disp("since z+z=1");
15 disp("T(x,y,z)=x+1+y");
16 disp("T(x,y,z)=1")

```

Scilab code Exa 3.3 Prove the identity

```

1 clear
2 clc;
3 disp("L.H.S = xy+x^y+yz");
4 disp("R.H.S = xy+x^y+x^z");
5 disp("Based on consensus theorem")
6 disp("we can write x^y+yz as x^y+yz+x^z bcoz the
    two expressions are equal");
7 disp("(x^y+yz+x^z(y+y))=x^y+yz+x^yz+x^y^z");
8 disp("x^y+yz+x^yz+x^y^z=x^y(1+z)+yz(1+x)");
9 disp("x^y+yz=x^y+yz+x^z");
10 disp("so L.H.S=xy+x^y+yz=xy+x^y+yz+x^z");
11 disp("In the similar way xy+yz+x^z can be simplified
    as xy+x^z");
12 disp("so L.H.S becomes xy+x^z+x^y");
13 disp("thus L.H.S= R.H.S");

```

14 `disp("hence proved")`

Scilab code Exa 3.4 Determine The Output of expression

```
1 clear;
2 clc;
3 //function definition
4 x=[0;0;0;0;1;1;1;1];
5 y=[0;0;1;1;0;0;1;1];
6 z=[0;1;0;1;0;1;0;1];
7 f=[1;0;1;1;0;0;1;1];
8 g=[0;1;0;1;1;0;1;0];
9 //calculating the values of expressions given
10 forg=bitor(f,g);
11 fandg=bitand(f,g);
12 fcmp=bitcmp(f,1);
13 s(:,1)=x;
14 s(:,2)=y;
15 s(:,3)=z;
16 s(:,4)=f;
17 s(:,5)=g;
18 s(:,6)=forg;
19 s(:,7)=fandg;
20 s(:,8)=fcmp;
21 p=[ ' x ', ' y ', ' z ', ' f ', ' g ', ' f+g ', ' fg '
      , ' f^ '];
22 disp(p);
23 disp(s);
```

Scilab code Exa 3.5 Simplify 3

```
1 clear;
2 clc;
```

```

3 disp("T(A,B,C,D)=A^C^+ABD+BC^D+AB^D^+ABCD^");
4 disp("Assume A^=x , C^=y , BD=z");
5 disp("Now from consensus theorem for the first three
      terms");
6 disp("BC^D is the redundant term so it can be
      removed");
7 disp("T(A,B,C,D)=A^C^+ABD+AB^D^+ABCD^");
8 disp("T(A,B,C,D)=A^C^+ABD+AD^(B^+BC)");
9 disp("we know that a+a^b=a+b");
10 disp("T(A,B,C,D)=A^C^+ABD+AD^(B^+C)");
11 disp("T(A,B,C,D)=A^C^+A(BD+D^(B^+C))");

```

Scilab code Exa 3.6 Simplify 4

```

1 clear
2 clc
3 disp("T(A,B,C,D)=A^B+ABD+AB^CD^+BC");
4 disp("T(A,B,C,D)=B(A^+AD)+C(AD^B^+B)");
5 disp("T(A,B,C,D)=B(A^+D)+C(AD^+B)");
6 disp("T(A,B,C,D)=A^B+BD+ACD^+BC");
7 disp("T(A,B,C,D)=A^B+BD+ACD^+BC(A+A^)");
8 disp("T(A,B,C,D)=A^B+A^BC+ABC+BD+ACD^");
9 disp("T(A,B,C,D)=A^B(1+C)+ABC+BD+ACD^");
10 disp("T(A,B,C,D)=A^B+ABC+BD+ACD^");
11 disp("**Now apply consensus theorem for 2nd 3rd and
      4th terms**");
12 disp("let x=D,y=B,z=AC");
13 disp("T(A,B,C,D)=A^B+BD+ACD^");

```

Scilab code Exa 3.7 Expand the Expression

```

1 clear;
2 clc;

```

```

3 disp("T(x,y,z)=x^y+z^+xyz");
4 disp("**To determine the canonical sum of products
we have to check for a product which is not a min
term and then multiply with the missing variable
such that the expression value doesnt change**")
;
5 disp("T(x,y,z)=x^y(z+z^)+(x+x^)(y+y^)z^+xyz");
6 disp("T(x,y,z)=x^yz+x^yz^+xyz^+xy^z^+x^yz^+x^y^z^+
xyz");
7 disp("T(x,y,z)=x^yz+x^yz^+xyz^+xy^z^+x^y^z^+xyz");

```

Scilab code Exa 3.8 Expand the Expression 2

```

1 clear;
2 clc;
3 disp("T(x,y,z)=x^(y^+z)");
4 disp("**To determine the canonical product of sums
form you need to check for a product which is not
a max term and then add it with the missing
terms such that the expression value is not
altered**");
5 disp("T(x,y,z)=(x^+yy^+zz^)(xx^+y^+z)");
6 disp("a+b+cc^ can be written as product of 2 max
terms (a+b+c)(a+b+c^)");
7 disp("a+bb^+cc^ can be written as (a+b+c)(a+b+c^)(a+
b^+c)(a+b^+c^)");
8 disp("from the above two properties we can write the
T(x,y,z) as");
9 disp("T(x,y,z)=(x^+y+z)(x^+y+z^)(x^+y^+z)(x^+y^+z^)(
x+y^+z)(x^+y^+z)");
10 disp("T(x,y,z)=(x^+y+z)(x^+y+z^)(x^+y^+z)(x^+y^+z^)(
x+y^+z)");

```

Scilab code Exa 3.9 POS

```
1 clear;
2 clc;
3 disp("T(x,y,z)=x^y^z^+x^y^z+x^yz+xyz+xy^z+xy^z^");
4 disp("the complement T^ consists of those minterms
      which are not contained in the expression for T")
      ;
5 disp("T=[x^yz^+xyz^]^");
6 disp("(x+y^+z)(x^+y^+z)");
```

Scilab code Exa 3.10 Tabulate the Function of 2 variables

```
1 clear;
2 clc;
3 a1=0;a2=a1;a3=a1;
4 //all combinations of 2 variable inputs
5 f(:,1)=[0;0;1;1];
6 f(:,2)=[0;1;0;1];
7 disp("The turth table of f for all the combinations
      of a0,a1,a2,a3 are shown below")
8 //determining the values of f for all combinations
      of a0,a1,a2,a3
9 for a3=0:1
10     for a2=0:1
11         for a1=0:1
12             for a0=0:1
13                 disp(['  a3' '  a2' '  a1' '  a0'])
14                 ;
15                 disp([a3 a2 a1 a0]);
16                 i=1;
17                 for x=0:1
18                     for y=0:1
19                         f0=bitand(a0,bitand(bitcmp(x
20                                     ,1),bitcmp(y,1)));
```



```

19         f1=bitand(a1,bitand(bitcmp(x
20             ,1),y));
21         f2=bitand(a2,bitand(x,bitcmp
22             (y,1)));
23         f3=bitand(a3,bitand(x,y));
24         f4=bitor(f0,f1);
25         f5=bitor(f2,f3);
26         f(i,3)=bitor(f4,f5);
27         i=i+1;
28     end
29 end
30 disp([' x' ' y' ' f']);
31 disp(f);
32 disp('*-----*');
33 end
34 end

```

Scilab code Exa 3.11 NOR

```

1 clear;
2 clc;
3 disp("x NOR x=(x+x)^");
4 disp("NOT Gate");
5 disp("x NOR x=x^x^=x^");
6 disp("OR Gate");
7 disp("(x NOR y) NOR (x NOR y) = (x^ NOR y^)^ = x+y")
8 ;
9 disp("AND Gate");
10 disp("(x NOR x) NOR (y NOR y) = x^ NOR y^ = xy");
11 disp("NAND Gate");
12 disp("NOT ((x NOR x) NOR (y NOR y))= NOT (x^ NOR y
    ^) = NOT(xy) = (xy)^");
13 disp("XOR Gate");

```

```

13 disp("(x^ NOR y^ ) NOR (x NOR y) = x^y+xy^");
14 disp("XNOR Gate");
15 disp("(x^ NOR y) NOR (x NOR y^ ) = xy+x^y^");
16 disp("Since every other gate can be implemented
      using NOR gate it is said to be functionally
      complete")

```

Scilab code Exa 3.12 Transmission function

```

1 clear;
2 clc;
3 disp("T=xy^+(x+y^)z");
4 disp("from the identity a+a^b=a+b");
5 disp("T=xy^+z");

```

Scilab code Exa 3.13 Air Conditioning System

```

1 clear;
2 clc;
3 disp("Air conditioning system of a storage warehouse
      will be turned on if and only if it satisfies
      these conditions");
4 disp("let W denotes weight of 100 tons or more");
5 disp("H denotes relative humidity of atleast 60
      percent");
6 disp("T denotes temperature above 60 degrees");
7 disp("P denotes barometric pressure of 30 or more");
8 disp("*-first condition-*");
9 disp("W<100 tons => W^ ,H>=60 => H , T>60 ");
10 disp("A1=W^HT");
11 disp("*-second condition-*");
12 disp("W>100 tons => W , T>60 => T");
13 disp("A2=WT");

```

```

14 disp("*-third condition -*");
15 disp("W<100 tons => W^ ,P>30 => P");
16 disp("A3=W^P");
17 disp("since Air condtioning system should be
    activated if any one of the above is satisfied so
    ");
18 disp("A=A1+A2+A3");
19 disp("A=W^HT+WT+W^P");
20 disp("A=T(W^H+W)+W^P");
21 disp("A=T(W+H)+W^P");
22 disp("Thus a combinational system with above
    expression makes the air conditioning system on
    when required")

```

Scilab code Exa 3.14 DeMorgans Law

```

1 clear;
2 clc;
3 i=1;
4 //all combinations of 2 variable inputs
5 f(:,1)=[0;0;1;1];
6 f(:,2)=[0;1;0;1];
7 //verifying D'morgan first law
8 for a=0:1
9     for b=0:1
10        f(i,3)=bitcmp(bitxor(a,b),1);
11        f(i,4)=bitand(bitcmp(a,1),bitcmp(b,1));
12        i=i+1;
13    end
14 end
15 disp("    a    b    (a+b)^    a^b^");
16 disp(f);
17 disp("Therefore (a+b)^=a^ b^ ");
18 //verfying D'morgan 2nd law
19 i=1;

```

```

20 for a=0:1
21     for b=0:1
22         f(i,3)=bitcmp(bitand(a,b),1);
23         f(i,4)=bitor(bitcmp(a,1),bitcmp(b,1));
24         i=i+1;
25     end
26 end
27 disp("  a      b      (ab)^      a^+b^");
28 disp(f);
29 //proving D'morgans laws theoretically
30 disp("(a+b)^=a^.b^");
31 disp("(a.b)^=a^+b^");
32 disp("we have show that (a+b)(a+b)^=0 and (a+b)+a^.b
      ^=1");
33 disp("(a+b)a^b^=aa^b^+ba^b^=0+a^bb^=0+0=0");
34 disp("(a+b)+a^b^=a+b+a^b^=a+b+a^=b+a+a^=b+1=1");
35 disp("This proves the first Dmorgan law and in the
      similar way 2nd law can also be proved");

```

Chapter 4

Minimization Of Switching Functions

check Appendix [AP 2](#) for dependency:

```
number_of.sci
```

Scilab code Exa 4.1 Irredundant expressions

```
1  clc;  
2  n=4;           //four variable kmap  
3  k=[1 1 0 1;  
4     0 1 1 1;  
5     0 1 1 0;  
6     0 0 0 0];  
7  k(:, :, 2)=zeros(n,n);    //temporary matrix to know  
8     whether a element is paired or not  
9  //declaring notations to display output  
10 var=['y' 'z' 'w' 'x'];  
11 p1=['y' 'z' '' 'y' 'z' 'yz' 'yz'];  
12 p2=['w' 'x' '' 'w' 'x' 'wx' 'wx'];  
13 //minimum redundant elements accepted while pairing  
14 cmn4=4;  
15 cmn2=2;
```

```

15 temp=1;
16 disp("The minimal expression of the given Kmap ");
17 disp(k(:,:,1));
18 disp(" is :");
19 disp(" ")
20 //16 cells
21 for i=1:n
22     for j=1:n
23         if(k(i,j)~=1)
24             temp=0;
25             break;
26         end
27     end
28 end
29 printf('f=');
30 if(temp==1)
31     printf("1");
32     abort;
33 end
34 //8 cells
35 z1=ones(2,4);
36 temp1=['00' '01' '11' '10'];
37 temp2=temp1';
38 for i=1:n
39     if(i==4)
40         t=1;
41     else
42         t=i+1;
43     end
44     z=[k(i,:,1);k(t,:,1)];
45     if(z==z1)
46         k(i,:,2)=[1 1 1 1];
47         k(t,:,2)=[1 1 1 1];
48         a=strsplit(temp2(i,1));
49         b=strsplit(temp2(t,1));
50         c=strcmp(a,b);
51         for in=1:max(size(c))
52             if(c(in)==0 & a(in)=='0')

```

```

53         printf('%s',var(in));
54         printf('+');
55         break;
56     else
57         if(c(in)==0 & a(in)=='1')
58             printf(var(in));
59             printf('+');
60             break;
61         end
62     end
63 end
64 end
65 end
66 z2=ones(4,2);
67 for j=1:n
68     if(j==4)
69         t=1;
70     else
71         t=j+1;
72     end
73     z=[k(:,j,1) k(:,t,1)];
74     if(z==z2)
75         k(:,j,2)=[1;1;1;1];
76         k(:,t,2)=[1;1;1;1];
77         a=strsplit(temp1(1,j));
78         b=strsplit(temp1(1,t));
79         c=strcmp(a,b);
80         for in=1:max(size(c))
81             if(c(in)==0 & a(in)=='0')
82                 printf('%s',var(2+in));
83                 printf('+');
84                 break;
85             else
86                 if(c(in)==0 & a(in)=='1')
87                     printf(var(2+in));
88                     printf('+');
89                     break;
90             end

```

```

91         end
92     end
93 end
94 end
95 //4 cells
96 z1=ones(1,4);
97 z2=ones(4,1);
98 z3=ones(2,2);
99 temp1=['00' '01' '11' '10'];
100 temp2=temp1';
101 for t=1:n
102     z=k(t, :, 1);
103     no=number_of(k(t, :, 2), 1);
104     if(z==z1 & no<cmn4)
105         k(t, :, 2)=z1;
106         a=strsplit(temp1(1,t));
107         for in=1:max(size(a))
108             if(a(in)=='0')
109                 printf('%s', var(in));
110             end
111             if(a(in)=='1')
112                 printf(var(in));
113             end
114         end
115         printf("+");
116     end
117 end
118 for t=1:n
119     z=k(:, t, 1);
120     no=number_of(k(:, t, 2), 1);
121     if(z==z2 & no<cmn4)
122         k(:, t, 2)=z2;
123         a=strsplit(temp2(t,1));
124         for in=1:max(size(a))
125             if(a(in)=='0')
126                 printf('%s', var(2+in));
127             end
128             if(a(in)=='1')

```



```

129             printf(var(2+in));
130         end
131     end
132         printf("+");
133     end
134 end
135 for i=1:n
136     for j=1:n
137         if(i==n)
138             t1=1;
139         else
140             t1=i+1;
141         end
142         if(j==n)
143             t2=1;
144         else
145             t2=j+1;
146         end
147         z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2,1)
148             ];
149         z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2,2)
150             ];
151         no=number_of(z5,1);
152         if(z4==z3 & no<cmn4)
153             k(i,j,2)=1;
154             k(i,t2,2)=1;
155             k(t1,j,2)=1;
156             k(t1,t2,2)=1;
157             a=strsplit(temp2(i,1));
158             b=strsplit(temp2(t1,1));
159             c=strcmp(a,b);
160             for in=1:max(size(c))
161                 if(c(in)==0 & a(in)=='0')
162                     printf('%s''',var(in));
163                 end
164                 if(c(in)==0 & a(in)=='1')
165                     printf(var(in));
166                 end
167             end
168         end
169     end
170 end

```

```

165         end
166         a=strsplit(temp1(1,j));
167         b=strsplit(temp1(1,t2));
168         c=strcmp(a,b);
169         for in=1:max(size(c))
170             if(c(in)==0 & a(in)=='0')
171                 printf('%s''',var(2+in));
172             end
173             if(c(in)==0 & a(in)=='1')
174                 printf(var(2+in));
175             end
176         end
177         printf("+"");
178     end
179 end
180 end
181 //2 cells
182 z6=[1 1];
183 z7=z6';
184 for i=1:n
185     for j=1:n
186         if(i==n)
187             t1=1;
188         else
189             t1=i+1;
190         end
191         if(j==n)
192             t2=1;
193         else
194             t2=j+1;
195         end
196         z8=[k(i,j,1) k(i,t2,1)];
197         z9=[k(i,j,2) k(i,t2,2)];
198         no1=number_of(z9,1);
199         if(z8==z6 & no1<cmn2 & i+j~=2)
200             k(i,j,2)=1;
201             k(i,t2,2)=1;
202             a=strsplit(temp1(1,j));

```

```

203         b=strsplit(temp1(1,t2));
204         c=strcmp(a,b);
205         for in=1:max(size(c))
206             if(c(in)==0 & a(in)=='0')
207                 printf(p1(1,i));
208                 printf('%s',var(2+in));
209                 printf("+");
210             end
211             if(c(in)==0 & a(in)=='1')
212                 printf(p1(1,i));
213                 printf(var(2+in));
214                 printf("+");
215             end
216         end
217     end
218 end
219 end
220 for i=1:n
221     for j=1:n
222         if(i==n)
223             t1=1;
224         else
225             t1=i+1;
226         end
227         if(j==n)
228             t2=1;
229         else
230             t2=j+1;
231         end
232         z10=[k(i,j,1);k(t1,j,1)];
233         z11=[k(i,j,2);k(t1,j,2)];
234         no2=number_of(z11,1);
235         if(z10==z7 & no2<cmn2)
236             k(i,j,2)=1;
237             k(t1,j,2)=1;
238             a=strsplit(temp2(i,1));
239             b=strsplit(temp2(t1,1));
240             c=strcmp(a,b);

```

```

241         for in=1:max(size(c))
242             if(c(in)==0 & a(in)=='0')
243                 printf(p2(j,1));
244                 printf('%s''',var(in));
245                 printf("+");
246             end
247             if(c(in)==0 & a(in)=='1')
248                 printf(p2(j,1));
249                 printf(var(in));
250                 printf("+");
251             end
252         end
253     end
254 end
255 end
256 //single cell
257 for i=1:n
258     for j=1:n
259         if(k(i,j,2)==0 & k(i,j,1)==1)
260             a=strsplit(temp1(1,j));
261             b=strsplit(temp2(i,1));
262             for in=1:max(size(a(:,1)))
263                 if(a(in,1)=='1')
264                     printf(var(in+2));
265                 else
266                     if(a(in,1)=='0')
267                         printf('%s''',var(2+in));
268                     end
269                 end
270             end
271         for in=1:max(size(b(:,1)))
272             if(b(in,1)=='1')
273                 printf(var(in));
274             else
275                 if(b(in,1)=='0')
276                     printf('%s''',var(in));
277                 end
278             end

```

```

279             end
280             if(i~=4 & j~=4)
281                 printf("+");
282             end
283         end
284     end
285 end
286 printf("0");

```

check Appendix [AP 2](#) for dependency:

number_of.sci

Scilab code Exa 4.2 Irredundant expressions 2

```

1  clc;
2  n=4;           //four variable kmap
3  k=[1 1 0 1;
4     0 1 1 1;
5     0 1 1 0;
6     0 0 0 0];
7  k(:,:,2)=zeros(n,n); //temporary matrix to know
   whether a element is paired or not
8  //declaring notations to display output
9  var=['y' 'z' 'w' 'x'];
10 p1=['y^z^' 'y' 'z' 'yz' 'yz'''];
11 p2=['w^x^' 'w' 'x' 'wx' 'wx'''];
12 //minimum redundant elements accepted while pairing
13 cmn4=1;
14 cmn2=1;
15 temp=1;
16 disp("The minimal epression of the given Kmap ");
17 disp(k(:,:,1));
18 disp(" is :");
19 disp(" ")
20 //16 cells

```

```

21 for i=1:n
22     for j=1:n
23         if(k(i,j)~=1)
24             temp=0;
25             break;
26         end
27     end
28 end
29 printf('f=');
30 if(temp==1)
31     printf("1");
32     abort;
33 end
34 //8 cells
35 z1=ones(2,4);
36 z2=ones(4,2);
37 temp1=['00' '01' '11' '10'];
38 temp2=temp1';
39 for i=1:n
40     if(i==4)
41         t=1;
42     else
43         t=i+1;
44     end
45     z=[k(i,:,1);k(t,:,1)];
46     if(z==z1)
47         k(i,:,2)=[1 1 1 1];
48         k(t,:,2)=[1 1 1 1];
49         a=strsplit(temp2(i,1));
50         b=strsplit(temp2(t,1));
51         c=strcmp(a,b);
52         for in=1:max(size(c))
53             if(c(in)==0 & a(in)=='0')
54                 printf('%s''',var(in));
55                 printf('+');
56                 break;
57             else
58                 if(c(in)==0 & a(in)=='1')

```

```

59             printf(var(in));
60             printf('+');
61             break;
62         end
63     end
64 end
65     end
66 end
67 for j=1:n
68     if(j==4)
69         t=1;
70     else
71         t=j+1;
72     end
73     z=[k(:,j,1) k(:,t,1)];
74     if(z==z2)
75         k(:,j,2)=[1;1;1;1];
76         k(:,t,2)=[1;1;1;1];
77         a=strsplit(temp1(1,j));
78         b=strsplit(temp1(1,t));
79         c=strcmp(a,b);
80         for in=1:max(size(c))
81             if(c(in)==0 & a(in)=='0')
82                 printf('%s',var(2+in));
83                 printf('+');
84                 break;
85             else
86                 if(c(in)==0 & a(in)=='1')
87                     printf(var(2+in));
88                     printf('+');
89                     break;
90                 end
91             end
92         end
93     end
94 end
95 //4 cells
96 z1=ones(1,4);

```

```

97 z2=ones(4,1);
98 z3=ones(2,2);
99 temp1=['00' '01' '11' '10'];
100 temp2=temp1';
101 for t=1:n
102     z=k(t, :, 1);
103     no=number_of(k(t, :, 2), 1);
104     if(z==z1 & no<cmn4)
105         k(t, :, 2)=z1;
106         a=strsplit(temp1(1,t));
107         for in=1:max(size(a))
108             if(a(in)=='0')
109                 printf('%s' , var(in));
110             end
111             if(a(in)=='1')
112                 printf(var(in));
113             end
114         end
115         printf("+");
116     end
117 end
118 for t=1:n
119     z=k(:, t, 1);
120     no=number_of(k(:, t, 2), 1);
121     if(z==z2 & no<cmn4)
122         k(:, t, 2)=z2;
123         a=strsplit(temp2(t,1));
124         for in=1:max(size(a))
125             if(a(in)=='0')
126                 printf('%s' , var(2+in));
127             end
128             if(a(in)=='1')
129                 printf(var(2+in));
130             end
131         end
132         printf("+");
133     end
134 end

```



```

135 for i=1:n
136     for j=1:n
137         if(i==n)
138             t1=1;
139         else
140             t1=i+1;
141         end
142         if(j==n)
143             t2=1;
144         else
145             t2=j+1;
146         end
147         z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2,1)
148             ];
149         z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2,2)
150             ];
151         no=number_of(z5,1);
152         if(z4==z3 & no<cmn4)
153             k(i,j,2)=1;
154             k(i,t2,2)=1;
155             k(t1,j,2)=1;
156             k(t1,t2,2)=1;
157             a=strsplit(temp2(i,1));
158             b=strsplit(temp2(t1,1));
159             c=strcmp(a,b);
160             for in=1:max(size(c))
161                 if(c(in)==0 & a(in)=='0')
162                     printf('%s''',var(in));
163                 end
164                 if(c(in)==0 & a(in)=='1')
165                     printf(var(in));
166                 end
167             end
168             a=strsplit(temp1(1,j));
169             b=strsplit(temp1(1,t2));
170             c=strcmp(a,b);
171             for in=1:max(size(c))
172                 if(c(in)==0 & a(in)=='0')

```

```

171         printf( '%s' , var(2+in));
172     end
173     if(c(in)==0 & a(in)=='1')
174         printf(var(2+in));
175     end
176 end
177 printf("+");
178 end
179 end
180 end
181 //2 cells
182 z6=[1 1];
183 z7=z6';
184 for i=1:n
185     for j=1:n
186         if(i==n)
187             t1=1;
188         else
189             t1=i+1;
190         end
191         if(j==n)
192             t2=1;
193         else
194             t2=j+1;
195         end
196         z8=[k(i,j,1) k(i,t2,1)];
197         z9=[k(i,j,2) k(i,t2,2)];
198         no1=number_of(z9,1);
199         if(z8==z6 & no1<cmn2)
200             k(i,j,2)=1;
201             k(i,t2,2)=1;
202             a=strsplit(temp1(1,j));
203             b=strsplit(temp1(1,t2));
204             c=strcmp(a,b);
205             for in=1:max(size(c))
206                 if(c(in)==0 & a(in)=='0')
207                     printf(p1(1,i));
208                     printf( '%s' , var(2+in));

```

```

209             printf("+");
210         end
211         if(c(in)==0 & a(in)=='1')
212             printf(p1(1,i));
213             printf(var(2+in));
214             printf("+");
215         end
216     end
217 end
218 end
219 end
220 for i=1:n
221     for j=1:n
222         if(i==n)
223             t1=1;
224         else
225             t1=i+1;
226         end
227         if(j==n)
228             t2=1;
229         else
230             t2=j+1;
231         end
232         z10=[k(i,j,1);k(t1,j,1)];
233         z11=[k(i,j,2);k(t1,j,2)];
234         no2=number_of(z11,1);
235         if(z10==z7 & no2<cmn2)
236             k(i,j,2)=1;
237             k(t1,j,2)=1;
238             a=strsplit(temp2(i,1));
239             b=strsplit(temp2(t1,1));
240             c=strcmp(a,b);
241             for in=1:max(size(c))
242                 if(c(in)==0 & a(in)=='0')
243                     printf(p2(j,1));
244                     printf(' %s ',var(in));
245                     printf("+");
246                 end

```

```

247             if(c(in)==0 & a(in)=='1')
248                 printf(p2(j,1));
249                 printf(var(in));
250                 printf("+");
251             end
252         end
253     end
254 end
255 //single cell
256 for i=1:n
257     for j=1:n
258         if(k(i,j,2)==0 & k(i,j,1)==1)
259             a=strsplit(temp1(1,j));
260             b=strsplit(temp2(i,1));
261             for in=1:max(size(a(:,1)))
262                 if(a(in,1)=='1')
263                     printf(var(in+2));
264                 else
265                     if(a(in,1)=='0')
266                         printf('%s',var(2+in));
267                     end
268                 end
269             end
270         end
271         for in=1:max(size(b(:,1)))
272             if(b(in,1)=='1')
273                 printf(var(in));
274             else
275                 if(b(in,1)=='0')
276                     printf('%s',var(in));
277                 end
278             end
279         end
280     end
281 end
282 printf("+");
283 end
284 printf("0");

```

check Appendix [AP 2](#) for dependency:

number_of.sci

Scilab code Exa 4.3 Reduce Expression

```
1  clc;
2  n=4;          //four variable kmap
3  k=[0 0 1 0;
4     1 1 1 0;
5     0 1 1 1;
6     0 1 0 0];
7  k(:,:,2)=zeros(n,n);    //temporary matrix to know
   whether a element is paired or not
8  //declaring notations to display output
9  var=['y' 'z' 'w' 'x'];
10 p1=['y^z^' 'y''z' 'yz' 'yz'''];
11 p2=['w^x^' 'w''x' 'wx' 'wx'''];
12 //minimum redundant elements accepted while pairing
13 cmn4=1;
14 cmn2=1;
15 temp=1;
16 disp("The minimal epression of the given Kmap ");
17 disp(k(:,:,1));
18 disp(" is :");
19 disp(" ")
20 //16 cells
21 for i=1:n
22     for j=1:n
23         if(k(i,j)~=1)
24             temp=0;
25             break;
26         end
27     end
28 end
```

```

29 printf('f=');
30 if(temp==1)
31     printf("1");
32     abort;
33 end
34 //8 cells
35 z1=ones(2,4);
36 z2=ones(4,2);
37 temp1=['00' '01' '11' '10'];
38 temp2=temp1';
39 for i=1:n
40     if(i==4)
41         t=1;
42     else
43         t=i+1;
44     end
45     z=[k(i, :, 1);k(t, :, 1)];
46     if(z==z1)
47         k(i, :, 2)=[1 1 1 1];
48         k(t, :, 2)=[1 1 1 1];
49         a=strsplit(temp2(i,1));
50         b=strsplit(temp2(t,1));
51         c=strcmp(a,b);
52         for in=1:max(size(c))
53             if(c(in)==0 & a(in)=='0')
54                 printf('%s' ',var(in));
55                 printf('+');
56                 break;
57             else
58                 if(c(in)==0 & a(in)=='1')
59                     printf(var(in));
60                     printf('+');
61                     break;
62                 end
63             end
64         end
65     end
66 end

```

```

67 for j=1:n
68     if(j==4)
69         t=1;
70     else
71         t=j+1;
72     end
73     z=[k(:,j,1) k(:,t,1)];
74     if(z==z2)
75         k(:,j,2)=[1;1;1;1];
76         k(:,t,2)=[1;1;1;1];
77         a=strsplit(temp1(1,j));
78         b=strsplit(temp1(1,t));
79         c=strcmp(a,b);
80         for in=1:max(size(c))
81             if(c(in)==0 & a(in)=='0')
82                 printf('%s''',var(2+in));
83                 printf('+');
84                 break;
85             else
86                 if(c(in)==0 & a(in)=='1')
87                     printf(var(2+in));
88                     printf('+');
89                     break;
90                 end
91             end
92         end
93     end
94 end
95 //2 cells
96 z6=[1 1];
97 z7=z6';
98 for i=1:n
99     for j=1:n
100         if(i==n)
101             t1=1;
102         else
103             t1=i+1;
104         end

```

```

105         if (j==n)
106             t2=1;
107         else
108             t2=j+1;
109         end
110         z8=[k(i,j,1) k(i,t2,1)];
111         z9=[k(i,j,2) k(i,t2,2)];
112         no1=number_of(z9,1);
113         if(z8==z6 & no1<cmn2)
114             if(i==3 & i+j==5)
115                 ;
116             else
117                 k(i,j,2)=1;
118                 k(i,t2,2)=1;
119                 a=strsplit(temp1(1,j));
120                 b=strsplit(temp1(1,t2));
121                 c=strcmp(a,b);
122                 for in=1:max(size(c))
123                     if(c(in)==0 & a(in)=='0')
124                         printf(p1(1,i));
125                         printf('%s' ',var(2+in));
126                         printf("+");
127                     end
128                     if(c(in)==0 & a(in)=='1')
129                         printf(p1(1,i));
130                         printf(var(2+in));
131                         printf("+");
132                     end
133                 end
134             end
135         end
136     end
137 end
138 for i=1:n
139     for j=1:n
140         if(i==n)
141             t1=1;
142         else

```



```

143         t1=i+1;
144     end
145     if(j==n)
146         t2=1;
147     else
148         t2=j+1;
149     end
150     z10=[k(i,j,1);k(t1,j,1)];
151     z11=[k(i,j,2);k(t1,j,2)];
152     no2=number_of(z11,1);
153     if(z10==z7 & no2<cmn2)
154         k(i,j,2)=1;
155         k(t1,j,2)=1;
156         a=strsplit(temp2(i,1));
157         b=strsplit(temp2(t1,1));
158         c=strcmp(a,b);
159         for in=1:max(size(c))
160             if(c(in)==0 & a(in)=='0')
161                 printf(p2(j,1));
162                 printf('%s''',var(in));
163                 printf("+"");
164             end
165             if(c(in)==0 & a(in)=='1')
166                 printf(p2(j,1));
167                 printf(var(in));
168                 printf("+"");
169             end
170         end
171     end
172 end
173 end
174 //4 cells
175 z1=ones(1,4);
176 z2=ones(4,1);
177 z3=ones(2,2);
178 temp1=['00' '01' '11' '10'];
179 temp2=temp1';
180 for t=1:n

```

```

181         z=k(t, :, 1);
182         no=number_of(k(t, :, 2), 1);
183         if(z==z1 & no<cmn4)
184             k(t, :, 2)=z1;
185             a=strsplit(temp1(1,t));
186             for in=1:max(size(a))
187                 if(a(in)=='0')
188                     printf('%s''',var(in));
189                 end
190                 if(a(in)=='1')
191                     printf(var(in));
192                 end
193             end
194             printf("+");
195         end
196     end
197     for t=1:n
198         z=k(:, t, 1);
199         no=number_of(k(:, t, 2), 1);
200         if(z==z2 & no<cmn4)
201             k(:, t, 2)=z2;
202             a=strsplit(temp2(t,1));
203             for in=1:max(size(a))
204                 if(a(in)=='0')
205                     printf('%s''',var(2+in));
206                 end
207                 if(a(in)=='1')
208                     printf(var(2+in));
209                 end
210             end
211             printf("+");
212         end
213     end
214     for i=1:n
215         for j=1:n
216             if(i==n)
217                 t1=1;
218             else

```

```

219         t1=i+1;
220     end
221     if (j==n)
222         t2=1;
223     else
224         t2=j+1;
225     end
226     z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2,1)
        ];
227     z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2,2)
        ];
228     no=number_of(z5,1);
229     if(z4==z3 & no<cmn4)
230         k(i,j,2)=1;
231         k(i,t2,2)=1;
232         k(t1,j,2)=1;
233         k(t1,t2,2)=1;
234         a=strsplit(temp2(i,1));
235         b=strsplit(temp2(t1,1));
236         c=strcmp(a,b);
237         for in=1:max(size(c))
238             if(c(in)==0 & a(in)=='0')
239                 printf('%s''',var(in));
240             end
241             if(c(in)==0 & a(in)=='1')
242                 printf(var(in));
243             end
244         end
245         a=strsplit(temp1(1,j));
246         b=strsplit(temp1(1,t2));
247         c=strcmp(a,b);
248         for in=1:max(size(c))
249             if(c(in)==0 & a(in)=='0')
250                 printf('%s''',var(2+in));
251             end
252             if(c(in)==0 & a(in)=='1')
253                 printf(var(2+in));
254             end

```

```

255         end
256         printf("+");
257     end
258 end
259 end
260 //single cell
261 for i=1:n
262     for j=1:n
263         if(k(i,j,2)==0 & k(i,j,1)==1)
264             a=strsplit(temp1(1,j));
265             b=strsplit(temp2(i,1));
266             for in=1:max(size(a(:,1)))
267                 if(a(in,1)=='1')
268                     printf(var(in+2));
269                 else
270                     if(a(in,1)=='0')
271                         printf('%s',var(2+in));
272                     end
273                 end
274             end
275             for in=1:max(size(b(:,1)))
276                 if(b(in,1)=='1')
277                     printf(var(in));
278                 else
279                     if(b(in,1)=='0')
280                         printf('%s',var(in));
281                     end
282                 end
283             end
284             printf("+");
285         end
286     end
287 end
288 printf("0");

```

check Appendix [AP 6](#) for dependency:

dec21bin.sci

check Appendix [AP 7](#) for dependency:

donkmap.sci

check Appendix [AP 2](#) for dependency:

number_of.sci

Scilab code Exa 4.4 BCD to Excess 3 Convertor

```
1  clc;
2  n=10;
3  //coverting decimal numbers into excess 3 values
4  for i=0:n-1
5      c(i+1,1)=dec21bin(i+3);
6  end
7  a=c;
8  b=zeros(10,4);
9  //placing excess 3 outputs in matrix for convenience
10 for i=1:n
11     j=4;
12     while(a(i,1)>=1)
13         b(i,j)=round(modulo(a(i,1),10));
14         a(i,1)=a(i,1)/10;
15         j=j-1;
16     end
17 end
18 //dont care is represented by a 2 since scilab
    doesnt allow a matrix to contain string and a
    number.
19 for i=n+1:16
20     b(i,:)=[2 2 2 2];
21 end
22 //map of each output variable
23 z=[b(1,1) b(5,1) b(13,1) b(9,1);b(2,1) b(6,1) b
    (14,1) b(10,1);
```

```

24     b(3,1) b(7,1) b(15,1) b(11,1);b(4,1) b(8,1) b
        (16,1) b(12,1)];
25 y=[b(1,2) b(5,2) b(13,2) b(9,2);b(2,2) b(6,2) b
        (14,2) b(10,2);
26     b(3,2) b(7,2) b(15,2) b(11,2);b(4,2) b(8,2) b
        (16,2) b(12,2)];
27 w=[b(1,3) b(5,3) b(13,3) b(9,3);b(2,3) b(6,3) b
        (14,3) b(10,3);
28     b(3,3) b(7,3) b(15,3) b(11,3);b(4,3) b(8,3) b
        (16,3) b(12,3)];
29 x=[b(1,4) b(5,4) b(13,4) b(9,4);b(2,4) b(6,4) b
        (14,4) b(10,4);
30     b(3,4) b(7,4) b(15,4) b(11,4);b(4,4) b(8,4) b
        (16,4) b(12,4)];
31 donkmap(w,1);
32 donkmap(x,2);
33 donkmap(y,3);
34 donkmap(z,4);

```

check Appendix [AP 5](#) for dependency:

check.sci

check Appendix [AP 4](#) for dependency:

noof3.sci

check Appendix [AP 2](#) for dependency:

number_of.sci

Scilab code Exa 4.5 5 variable Kmap

```

1 // f=x^y^z+wxz+xy+v^w^yz^
2 clc;
3 n=4;
4 k=[0 0 0 0;

```

```

5     1 0 1 1;
6     0 1 1 0;
7     1 1 1 0];
8 k(:,:,2)=[0 0 0 0;
9           1 0 1 1;
10          0 1 1 0;
11          0 1 1 0];
12 //k=[1 0 0 0;
13 //   0 0 0 0;
14 //   0 0 0 0;
15 //   0 0 1 0];
16 //k(:,:,2)=[1 0 0 0;
17 //           0 0 0 0;
18 //           0 0 0 0;
19 //           1 0 0 0];
20 k(:,:,3)=zeros(n,n);
21 k(:,:,4)=zeros(n,n);
22 var=['y' 'z' 'v' 'w' 'x'];
23 p1=['y' 'z' 'y' 'z' 'yz' 'yz'];
24 p2=['v' 'w' 'x' 'v' 'w' 'x' 'v' 'wx' 'v' 'wx';
25     'vw' 'x' 'vw' 'x' 'vwx' 'vwx'];
26 cmn16=9;
27 cmn8=5;
28 cmn4=3;
29 cmn2=2;
30 temp=1;
31     printf('The minimal expression of the given Kmap
32           ');
33     disp(k(:,:,1));
34     disp(k(:,:,2));
35     disp(" is ");
36     printf('f');
37     printf("=");
38 //32 cells
39 for i=1:n
40     for j=1:n
41         for l=1:2
42             if(k(i,j,l)~=1 & k(i,j,l)~=2)

```

```

42             temp=0;
43             break;
44         end
45     end
46 end
47 end
48 if(temp==1)
49     printf("1");
50     abort;
51 end
52 //16 cells
53 //8+8 row cells
54 z1=ones(2,4,2);
55 z2=ones(4,2,2);
56 temp1=['00' '01' '11' '10'];
57 temp2=['000' '001' '011' '010' '100' '101' '111' '
        110'];
58 for i=1:n
59     if(i==4)
60         t=1;
61     else
62         t=i+1;
63     end
64     z=[k(i,:,1:2);k(t,:,1:2)];
65     z1=[k(i,:,3:4);k(t,:,3:4)];
66     if(noof3(z,0)==0 & noof3(z1,1)<cmn16)
67         k(i,:,3:4)=ones(4,2);
68         k(t,:,3:4)=ones(4,2);
69         a=strsplit(temp1(1,i));
70         b=strsplit(temp1(1,t));
71         c=strcmp(a,b);
72         for in=1:max(size(c))
73             if(c(in)==0 & a(in)=='0')
74                 printf('%s' ',var(in));
75             else
76                 if(c(in)==0 & a(in)=='1')
77                     printf(var(in));
78                 end

```



```

79             end
80         end
81         k(i, :, 3:4) = ones(1, 4, 2);
82         k(t, :, 3:4) = ones(1, 4, 2);
83     end
84 end
85 //8+8 column cells
86 for j=1:n
87     if(j==4)
88         t=1;
89     else
90         t=j+1;
91     end
92     z=[k(:, j, 1:2) k(:, t, 1:2)];
93     z1=[k(:, j, 3:4) k(:, t, 3:4)];
94     if(noof3(z,0)==0 & noof3(z1,1)<cmn16)
95         k(:, j, 3:4) = ones(4, 2);
96         k(:, t, 3:4) = ones(4, 2);
97         a=strsplit(temp2(1, j));
98         b=strsplit(temp2(1, t));
99         c=strsplit(temp2(1, j+4));
100        d=strsplit(temp2(1, t+4));
101        c1=check(a, b, c, d);
102        for in=1:max(size(c1))
103            if(c1(in)==0 & a(in)=='0')
104                printf(' %s ', var(2+in));
105            else
106                if(c1(in)==0 & a(in)=='1')
107                    printf(var(2+in));
108                end
109            end
110        end
111        printf(" + ");
112        k(:, j, 3:4) = ones(1, 4, 2);
113        k(:, t, 3:4) = ones(1, 4, 2);
114    end
115 end
116 //4x4 front matrix

```

```

117 if(number_of(k(:,:,1),0)==0 & number_of(k(:,:,3),1)<
    cmn16)
118     printf(var(3));
119     printf('');
120     k(:,:,3)=ones(4,4);
121 end
122 //4x4 rear matrix
123 if(number_of(k(:,:,2),0)==0 & number_of(k(:,:,4),1)<
    cmn16)
124     printf(var(3));
125     k(:,:,4)=ones(4,4);
126 end
127 //8 cells
128 //2x2 front and rear cells
129 for i=1:n
130     for j=1:n
131         if(i==4)
132             t=1;
133         else
134             t=i+1;
135         end
136         if(j==4)
137             u=1;
138         else
139             u=j+1;
140         end
141         z=k(i,j,1:2);
142         z(1,2,:)=k(i,u,1:2);
143         z(2,1,:)=k(t,j,1:2);
144         z(2,2,:)=k(t,u,1:2);
145         z1=k(i,j,3:4);
146         z1(1,2,:)=k(i,u,3:4);
147         z1(2,1,:)=k(t,j,3:4);
148         z1(2,2,:)=k(t,u,3:4);
149         if(noof3(z,0)==0 & noof3(z1,1)<cmn8)
150             a=strsplit(temp1(1,i));
151             b=strsplit(temp1(1,t));
152             c=strcmp(a,b);

```

```

153         for in=1:max(size(c))
154             if(c(in)==0 & a(in)=='0')
155                 printf('%s''',var(in));
156             else
157                 if(c(in)==0 & a(in)=='1')
158                     printf(var(in));
159                 end
160             end
161         end
162         a=strsplit(temp2(1,j));
163         b=strsplit(temp2(1,u));
164         c=strsplit(temp2(1,4+j));
165         d=strsplit(temp2(1,4+u));
166         c1=check(a,b,c,d);
167         for in=1:max(size(c1))
168             if(c1(in)==0 & a(in)=='0')
169                 printf('%s''',var(2+in));
170             else
171                 if(c1(in)==0 & a(in)=='1')
172                     printf(var(2+in));
173                 end
174             end
175         end
176         k(i,j,3:4)=ones(1,1,2);
177         k(i,u,3:4)=ones(1,1,2);
178         k(t,j,3:4)=ones(1,1,2);
179         k(t,u,3:4)=ones(1,1,2);
180         printf("++");
181     end
182 end
183 end
184 //1x4 front and rear cells
185 for i=1:n
186     z=k(i,:,1:2);
187     z1=k(i,:,3:4);
188     if(noof3(z,0)==0 & noof3(z1,1)<cmn8)
189         printf(p1(i));
190         printf("++");

```

```

191         k(i, :, 3:4) = ones(1, 4, 2);
192     end
193 end
194 //4x1 front and rear cells
195 for j=1:n
196     z=k(:, j, 1:2);
197     z1=k(:, j, 3:4);
198     if(noof3(z,0)==0 & noof3(z1,1)<cmn8)
199         a=strsplit(temp2(1,j));
200         b=strsplit(temp2(1,u));
201         c=strcmp(a,b);
202         for in=1:max(size(c))
203             if(c(in)==0 & a(in)=='0')
204                 printf('%s', var(2+in));
205             else
206                 if(c(in)==0 & a(in)=='1')
207                     printf(var(2+in));
208                 end
209             end
210         end
211         printf(" ");
212         k(:, j, 3:4) = ones(1, 2, 4);
213     end
214 end
215 //2x4 front cells
216 for i=1:n
217     if(i==4)
218         t=1;
219     else
220         t=i+1;
221     end
222     z=k(i, :, 1);
223     z(2, :, 1)=k(t, :, 1);
224     z1=k(i, :, 3);
225     z1(2, :, 1)=k(t, :, 3);
226     if(number_of(z,0)==0 & number_of(z1,1)<cmn8)
227         a=strsplit(temp1(1,i));
228         b=strsplit(temp1(1,t));

```

```

229         c=strcmp(a,b);
230         for in=1:max(size(c))
231             if(c(in)==0 & a(in)=='0')
232                 printf('%s''',var(in));
233             else
234                 if(c(in)==0 & a(in)=='1')
235                     printf(var(in));
236                 end
237             end
238         end
239         printf('%s''',var(3));
240         printf("++");
241         k(i,:,3)=ones(1,4);
242         k(t,:,3)=ones(1,4);
243     end
244 end
245 //2x4 rear cells
246 for i=1:n
247     if(i==4)
248         t=1;
249     else
250         t=i+1;
251     end
252     z=k(i,:,2);
253     z(2,:,1)=k(t,:,2);
254     z1=k(i,:,4);
255     z1(2,:,1)=k(t,:,4);
256     if(number_of(z,0)==0 & number_of(z1,1)<cmn8)
257         a=strsplit(temp1(1,i));
258         b=strsplit(temp1(1,t));
259         c=strcmp(a,b);
260         for in=1:max(size(c))
261             if(c(in)==0 & a(in)=='0')
262                 printf('%s''',var(in));
263             else
264                 if(c(in)==0 & a(in)=='1')
265                     printf(var(in));
266                 end

```

```

267         end
268     end
269     printf(var(3));
270     printf("+");
271     k(i, :, 4) = ones(1, 4);
272     k(t, :, 4) = ones(1, 4);
273 end
274 end
275 //4x2 front cells
276 for j=1:n
277     if(j==4)
278         u=1;
279     else
280         u=j+1;
281     end
282     z=k(:, j, 1);
283     z(:, 2, 1)=k(:, u, 1);
284     z1=k(:, j, 3);
285     z1(:, 2, 1)=k(:, u, 3);
286     if(number_of(z, 0)==0 & number_of(z1, 1)<cmn8)
287         a=strsplit(temp2(1, i));
288         b=strsplit(temp2(1, t));
289         c=strcmp(a, b);
290         for in=1:max(size(c))
291             if(c(in)==0 & a(in)=='0')
292                 printf('%s', var(in));
293             else
294                 if(c(in)==0 & a(in)=='1')
295                     printf(var(in));
296                 end
297             end
298         end
299     printf('%s', var(3));
300     printf("+");
301     k(:, j, 3) = ones(4, 1);
302     k(:, u, 3) = ones(4, 1);
303 end
304 end

```

```

305 //4x2 rear cells
306 for j=1:n
307     if(j==4)
308         u=1;
309     else
310         u=j+1;
311     end
312     z=k(:,j,2);
313     z(:,2,1)=k(:,u,2);
314     z1=k(:,j,4);
315     z1(:,2,1)=k(:,u,4);
316     if(number_of(z,0)==0 & number_of(z1,1)<cmn8)
317         a=strsplit(temp2(1,i));
318         b=strsplit(temp2(1,t));
319         c=strcmp(a,b);
320         for in=1:max(size(c))
321             if(c(in)==0 & a(in)=='0')
322                 printf('%s''',var(4+in));
323             else
324                 if(c(in)==0 & a(in)=='1')
325                     printf(var(4+in));
326                 end
327             end
328         end
329         printf(var(3));
330         printf("++");
331         k(:,j,4)=ones(4,1);
332         k(:,u,4)=ones(4,1);
333     end
334 end
335 //4 cells
336 //1x4 front cells
337 for i=1:n
338     z=k(i,,:,1);
339     z1=k(i,,:,3);
340     if(number_of(z,0)==0 & number_of(z1,1)<cmn4)
341         printf(p1(1,i));
342         printf('%s''',var(3));

```

```

343         printf("+");
344         k(i, :, 3) = ones(1, 4);
345     end
346 end
347 //1x4 rear cells
348 for i=1:n
349     z=k(i, :, 2);
350     z1=k(i, :, 4);
351     if(number_of(z,0)==0 & number_of(z1,1)<cmn4)
352         printf(p1(1,i));
353         printf(var(3));
354         printf("+");
355         k(i, :, 4) = ones(1, 4);
356     end
357 end
358 //4x1 front cells
359 for j=1:n
360     z=k(:, j, 1);
361     z1=k(:, j, 3);
362     if(number_of(z,0)==0 & number_of(z1,1)<cmn4)
363         printf(p2(j,1));
364         printf("+");
365         k(:, j, 3) = ones(4, 1);
366     end
367 end
368 //4x1 rear cells
369 for j=1:n
370     z=k(:, j, 2);
371     z1=k(:, j, 4);
372     if(number_of(z,0)==0 & number_of(z1,1)<cmn4)
373         printf(p2(4+j,1));
374         printf("+");
375         k(:, j, 4) = ones(4, 1);
376     end
377 end
378 //2x1 front and rear matrix
379 for i=1:n
380     for j=1:n

```



```

381     if(i==4)
382         t=1;
383     else
384         t=i+1;
385     end
386     z=[k(i,j,1);k(t,j,1)];
387     z(:,:,2)=[k(i,j,2) k(t,j,2)];
388     z1=[k(i,j,3);k(t,j,3)];
389     z1(:,:,2)=[k(i,j,4) k(t,j,4)];
390     if(noof3(z,0)==0 & noof3(z1,1)<cmn4)
391         a=strsplit(temp1(1,i));
392         b=strsplit(temp1(1,t));
393         c=strcmp(a,b);
394         for in=1:max(size(c))
395             if(c(in)==0 & a(in)=='0')
396                 printf('%s''',var(in));
397             else
398                 if(c(in)==0 & a(in)=='1')
399                     printf(var(in));
400                 end
401             end
402         end
403         a=strsplit(temp2(1,j));
404         b=strsplit(temp2(1,4+j));
405         c=strcmp(a,b);
406         for in=1:max(size(c))
407             if(c(in)==0 & a(in)=='0')
408                 printf('%s''',var(2+in));
409             else
410                 if(c(in)==0 & a(in)=='1')
411                     printf(var(2+in));
412                 end
413             end
414         end
415         printf('+');
416         k(i,j,3)=1;k(t,j,3)=1;
417         k(i,j,4)=1; k(t,j,4)=1;
418     end

```

```

419     end
420 end
421 //1x2 front and rear matrix
422 for i=1:n
423     for j=1:n
424         if(j==4)
425             u=1;
426         else
427             u=j+1;
428         end
429         z=[k(i,j,1) k(i,u,1)];
430         z(:,:,2)=[k(i,j,2) k(i,u,2)];
431         z1=[k(i,j,3) k(i,u,3)];
432         z1(:,:,2)=[k(i,j,4) k(i,u,4)];
433         if(noof3(z,0)==0 & noof3(z1,1)<1)
434             printf(p1(i));
435             a=strsplit(temp2(1,j));
436             b=strsplit(temp2(1,u));
437             c=strsplit(temp2(1,4+j));
438             d=strsplit(temp2(1,4+j));
439             c1=check(a,b,c,d);
440             for in=1:max(size(c1))
441                 if(c1(in)==0 & a(in)=='0')
442                     printf('%s''',var(2+in));
443                 else
444                     if(c1(in)==0 & a(in)=='1')
445                         printf(var(2+in));
446                     end
447                 end
448             end
449             printf('+');
450             k(i,j,3)=1; k(i,u,3)=1;
451             k(i,j,4)=1; k(i,u,4)=1;
452         end
453     end
454 end
455 //2 cells
456 //1x2 front cells

```

```

457 for i=1:n
458     for j=1:n
459         if(j==4)
460             u=1;
461         else
462             u=j+1;
463         end
464         z=[k(i,j,1) k(i,u,1)];
465         z1=[k(i,j,3) k(i,u,3)];
466         if(number_of(z,0)==0 & number_of(z1,1)<cmn2)
467             printf(p1(1,i));
468             a=strsplit(temp2(1,j));
469             b=strsplit(temp2(1,u));
470             c=strcmp(a,b);
471             for in=1:max(size(c))
472                 if(c(in)==0 & a(in)=='0')
473                     printf('%s''',var(2+in));
474                 else
475                     if(c(in)==0 & a(in)=='1')
476                         printf(var(2+in));
477                     end
478                 end
479             end
480             printf('+');
481             k(i,j,3)=1; k(i,u,3)=1;
482         end
483     end
484 end
485 //1x2 rear cells
486 for i=1:n
487     for j=1:n
488         if(j==4)
489             u=1;
490         else
491             u=j+1;
492         end
493         z=[k(i,j,2) k(i,u,2)];
494         z1=[k(i,j,4) k(i,u,4)];

```

```

495         if(number_of(z,0)==0 & number_of(z1,1)<cmn2)
496             printf(p1(1,i));
497             a=strsplit(temp2(1,4+j));
498             b=strsplit(temp2(1,4+u));
499             c=strcmp(a,b);
500             for in=1:max(size(c))
501                 if(c(in)==0 & a(in)=='0')
502                     printf('%s''',var(2+in));
503                 else
504                     if(c(in)==0 & a(in)=='1')
505                         printf(var(2+in));
506                     end
507                 end
508             end
509             printf('+');
510             k(i,j,4)=1; k(i,u,4)=1;
511         end
512     end
513 end
514 //2x1 front cells
515 for i=1:n
516     for j=1:n
517         if(i==4)
518             t=1;
519         else
520             t=i+1;
521         end
522         z=[k(i,j,1);k(t,j,1)];
523         z1=[k(i,j,3) k(t,j,3)];
524         if(number_of(z,0)==0 & number_of(z1,1)<cmn2)
525             a=strsplit(temp1(1,i));
526             b=strsplit(temp1(1,t));
527             c=strcmp(a,b);
528             for in=1:max(size(c))
529                 if(c(in)==0 & a(in)=='0')
530                     printf('%s''',var(in));
531                 else
532                     if(c(in)==0 & a(in)=='1')

```

```

533                                     printf(var(in));
534                                     end
535                                 end
536                             end
537                             printf(p2(j,1))
538                             printf('+');
539                             k(i,j,3)=1; k(i,u,3)=1;
540                         end
541                     end
542                 end
543                 //2x1 rear cells
544                 for i=1:n
545                     for j=1:n
546                         if(i==4)
547                             t=1;
548                         else
549                             t=i+1;
550                         end
551                         z=[k(i,j,2);k(t,j,2)];
552                         z1=[k(i,j,4) k(t,j,4)];
553                         if(number_of(z,0)==0 & number_of(z1,1)<cmn2)
554                             a=strsplit(temp1(1,i));
555                             b=strsplit(temp1(1,t));
556                             c=strcmp(a,b);
557                             for in=1:max(size(c))
558                                 if(c(in)==0 & a(in)=='0')
559                                     printf('%s''',var(in));
560                                 else
561                                     if(c(in)==0 & a(in)=='1')
562                                         printf(var(in));
563                                     end
564                                 end
565                             end
566                             printf(p2(4+j,1))
567                             printf('+');
568                             k(i,j,4)=1; k(i,u,4)=1;
569                         end
570                 end

```

```

571 end
572 //1 cell front and rear matrix
573 for i=1:n
574     for j=1:n
575         z=k(i,j,1:2);
576         z1=k(i,j,3:4);
577         if(noof3(z,0)==0 & noof3(z1,1)<cmn2)
578             printf(p1(1,i));
579             a=strsplit(temp2(1,j));
580             b=strsplit(temp2(1,4+j));
581             c=strcmp(a,b);
582             for in=2:max(size(c))
583                 if(a(in)=='0' & c(in)==0)
584                     printf('%s''',var(2+in));
585                 else
586                     if(a(in)=='1' & c(in)==0)
587                         printf(var(2+in));
588                     end
589                 end
590             end
591             printf('+');
592             k(i,j,3:4)=ones(1,1,2);
593         end
594     end
595 end
596 //single cell
597 for i=1:n
598     for j=1:n
599         for z=1:2
600             if(k(i,j,z)==1 & k(i,j,z+2)==0)
601                 printf(p2(j,1));
602                 printf(p1(1,i));
603                 printf('+');
604             end
605         end
606     end
607 end
608 printf('0');

```

Scilab code Exa 4.6 Prime Implicants

```
1  clc;
2  clear;
3  z=1;
4  //evaluating first expression
5  for i=0:1
6      for j=0:1
7          for k=0:1
8              for l=0:1
9                  f1(z,1)=bitor(bitand(i,j),bitand(k,l
10                     ));
11                 z=z+1;
12             end
13         end
14     end
15     z=1;
16     //evaluating 2nd expression
17     for i=0:1
18         for j=0:1
19             for k=0:1
20                 for l=0:1
21                     f2(z,1)=bitand(bitand(i,j),bitcmp(k
22                         ,1));
23                     z=z+1;
24                 end
25             end
26         end
27     //determining whether f covers h or not.
28     for i=1:16
29         if(f2(i,1)==1)
30             if(f2(i,1)==f1(i,1))
```

```

31         ;
32         else
33         disp("f doesnt cover h");
34         abort;
35     end
36 end
37 end
38 disp("f covers h and h implies f");

```

Scilab code Exa 4.7 Prime Implicants 2

```

1  clc;
2  clear;
3  disp("f=x^y+xz+y^z^");
4  disp("since x^y is a prime implicant neither x^ nor
      y alone implies f");
5  disp("only x^y are covered by f");

```

check Appendix [AP 1](#) for dependency:

karmap.sci

check Appendix [AP 3](#) for dependency:

karmap1.sci

check Appendix [AP 2](#) for dependency:

number_of.sci

Scilab code Exa 4.8 Prime Implicants Of a Function

```

1  clc;
2  k=[1 1 0 1;
3     0 1 1 1;

```



```
4     0 1 1 0;  
5     0 0 0 0];  
6 disp("The prime implicants of function f");  
7 karmap(k);  
8 karmap1(k);
```

check Appendix [AP 1](#) for dependency:

karmap.sci

check Appendix [AP 2](#) for dependency:

number_of.sci

Scilab code Exa 4.9 Prime Implicants Of a Function 2

```
1 clc;  
2 k=[0 1 1 1;  
3     0 1 1 0;  
4     0 0 1 0;  
5     0 0 1 0];  
6 disp("The prime implicants of the function f");  
7 karmap(k);
```

check Appendix [AP 8](#) for dependency:

karmap3.sci

check Appendix [AP 2](#) for dependency:

number_of.sci

Scilab code Exa 4.10 Cyclic Prime Implicant map

```
1 clc;  
2 k=[1 1 0 1;  
3     0 1 1 1];  
4 karmap3(k);
```

Chapter 5

Logical Design

Scilab code Exa 5.1 Odd Parity Bit Generator

```
1  clc;
2  clear;
3  //Takes x input and check whether it is valid logic
   level or not.
4  x=input("x = ");
5  while(x~=0 & x~=1)
6      disp("enter a valid logical level");
7      x=input("x = ");
8  end
9  //Takes y input and check whether it is valid logic
   level or not.
10 y=input("y = ");
11 while(y~=0 & y~=1)
12     disp("enter a valid logical level");
13     y=input("y = ");
14 end
15 //Takes z input and check whether it is valid logic
   level or not.
16 z=input("z = ");
17 while(z~=0 & z~=1)
18     disp("enter a valid logical level");
```

```

19     z=input("z = ");
20 end
21 p1=bitand(bitand(bitcmp(x,1),bitcmp(y,1)),z);
22 p2=bitand(bitand(bitcmp(x,1),y),bitcmp(z,1));
23 p3=bitand(bitand(bitcmp(y,1),x),bitcmp(z,1));
24 p4=bitand(bitand(x,y),z);
25 p=bitor(bitor(p1,p2),bitor(p3,p4));
26 disp(p,"The output of the odd parity generator
      circuit is");
27 disp("p= x^y^z+x^yz^+xy^z^+xyz");

```

Scilab code Exa 5.2 Serial To Parallel converter

```

1  clc;
2  clear;
3  //Takes x input and check whether it is valid logic
   level or not.
4  disp("enter the vaLue of input Line & controL
      signaLs C1 and C2");
5  x=input("x = ");
6  while(x~=0 & x~=1)
7      disp("enter a vaLid LogicaL Level");
8      x=input("x = ");1
9  end
10 //Takes C1 input and check whether it is valid logic
    level or not.
11 C1=input("C1 = ");
12 while(C1~=0 & C1~=1)
13     disp("enter a vaLid LogicaL Level");
14     C1=input("C1 = ");
15 end
16 //Takes C2 input and check whether it is valid logic
    level or not.
17 C2=input("C2 = ");
18 while(C2~=0 & C2~=1)

```

```

19     disp("enter a valid Logical Level");
20     C2=input("C2 = ");
21 end
22 disp("-----OUTPUT-----");
23 disp(L1=bitand(bitand(bitcmp(C1,1),bitcmp(C2,1)),x),
      "L1 = ");
24 disp(L2=bitand(bitand(bitcmp(C1,1),C2),x),"L2 = ");
25 disp(L3=bitand(bitand(C1,bitcmp(C2,1)),x),"L3 = ");
26 disp(L4=bitand(bitand(C1,C2),x),"L4 = ");
27 disp("L1=xC1^C2^");
28 disp("L2=xC1^C2");
29 disp("L3=xC1C2^");
30 disp("L4=xC1^C2^");

```

Scilab code Exa 5.3 Transmission function for a network

```

1  clc;
2  clear;
3  //Takes input and check whether it is valid logic
   level or not.
4  w=input("w = ");
5  while(w~=0 & w~=1)
6     disp("enter a valid logical level");
7     w=input("w = ");
8  end
9  x=input("x = ");
10 while(x~=0 & x~=1)
11    disp("enter a valid logical level");
12    x=input("x = ");
13 end
14 y=input("y = ");
15 while(y~=0 & y~=1)
16    disp("enter a valid logical level");
17    y=input("y = ");
18 end

```

```

19 z=input("z = ");
20 while(z~=0 & z~=1)
21     disp("enter a valid logical level");
22     z=input("z = ");
23 end
24 disp("original network  $x^{\wedge} [((y^{\wedge} z+z^{\wedge} y)w^{\wedge})+w+y^{\wedge}+x^{\wedge} z^{\wedge}]$ ")
    ;
25 disp("x^{\wedge} [w^{\wedge} y^{\wedge} z+w^{\wedge} yz^{\wedge}+w+y^{\wedge}+x^{\wedge} z^{\wedge}]");
26 disp("x^{\wedge} [y^{\wedge} (w^{\wedge} z+1)+w^{\wedge} yz^{\wedge}+w+x^{\wedge} z^{\wedge}]");
27 disp("x^{\wedge} [y^{\wedge}+w+yz^{\wedge}+x^{\wedge} z^{\wedge}]");
28 disp("x^{\wedge} [y^{\wedge}+yz^{\wedge}+w+x^{\wedge} z^{\wedge}]");
29 disp("x^{\wedge} [y^{\wedge}+z^{\wedge}+x^{\wedge} z^{\wedge}+w]");
30 disp("x^{\wedge} [y^{\wedge}+z^{\wedge}+w]");
31 //output of the relay network for the inputs given
32 disp(p=bitand(bitcmp(x,1),bitor(bitor(bitcmp(y,1),
    bitcmp(z,1)),w)),"output = ");

```

Scilab code Exa 5.4 4 Input Contact Network

```

1 clc;
2 clear;
3 //Takes input and check whether it is valid logic
    level or not.
4 w=input("w = ");
5 while(w~=0 & w~=1)
6     disp("enter a valid logical level");
7     w=input("w = ");
8 end
9 x=input("x = ");
10 while(x~=0 & x~=1)
11     disp("enter a valid logical level");
12     x=input("x = ");
13 end
14 y=input("y = ");
15 while(y~=0 & y~=1)

```

```

16     disp("enter a valid logical level");
17     y=input("y = ");
18 end
19 z=input("z = ");
20 while(z~=0 & z~=1)
21     disp("enter a valid logical level");
22     z=input("z = ");
23 end
24 disp("We have 4 relays W,X,Y,Z which takes BCD
      number as input");
25 disp("so the max value value we can get is 1001");
26 disp("By drawing the Karnaugh map(consider all terms
      above 1001 as dont cares ) the minimised
      expression is");
27 disp("T(w,x,y,z)=wz+xyz'+x'yz");
28 disp("T(w,x,y,z)=xyz'+x'yz");
29 disp("Draw the series parallel realization of T");
30 disp("We can identify the redundancy of y contact at
      the right end");
31 disp("which can be removed");
32 disp("The cutset realisation of the above minimized
      series parallel network is");
33 disp("T(w,x,y,z)=(w+y)(x+z)(x'+z')");

```

Scilab code Exa 5.5 Minimal contact Network

```

1  clc;
2  clear;
3  disp("Given minimal contact network is T(w,x,y,z)=
      wxy+wxz+w'x'y'z'+w'x'yz");
4  disp("T(w,x,y,z)=wx(y+z)+w'x'(y'z'+yz)");
5  disp("Draw the equivalent series parallel circuit of
      T");
6  disp("Interchange the locations of contacts of w'
      and x' and connect the nodes");

```

```
7 disp("By connecting in the above manner there is no
    logical effect since the connection path is not
    at all used");
8 disp("now the lower branch of  $yz+y^{\wedge}z^{\wedge}$  can be written
    as  $(y+z^{\wedge})(y^{\wedge}+z)$  so transfer contacts can be used
    ");
9 disp("This parallel connection enables us to combine
    two parallel z contacts and thus the minimum
    spring connection is obtained");
```

Chapter 6

Functional Decomposition And Symmetric Functions

Scilab code Exa 6.1 Function Decomposition

```
1 clc;
2 clear;
3 disp("Given function f(w,x,y,z) can be written as
      follows");
4 disp("f(w,x,y,z)=w^x^z^+wx^z+w^yz+wyz^");
5 disp("f(w,x,y,z)=((w^z^+wz)x^+(w^z+wz^)^y)");
6 disp("let Q=w^z^+wz");
7 disp("then we can rewrite f as f(w,x,y,z)=Qx^+Q^y");
8 disp("f(w,x,y,z)=Qx^+Q^y=F[Q(w,z),x,y]");
```

Scilab code Exa 6.3 Multiplicity

```
1 clc;
2 clear;
3 //determines the row and column multiplicity of a
  mtraix
```



```

4 //m=input(" enter the number of variables in the
   function=");
5 m=4;
6 for i=1:m
7     for j=1:m
8         x(i,j)=0;
9     end
10 end
11 //k='y';
12 //disp("Enter the cells index whose values are 1 in
   the matrix representing your function");
13 //while(k~='n')
14 //     i=input("x index =");
15 //     j=input("y index =");
16 //     x(i,j)=1;
17 //     k=input("If you want to enter more enter y(in
   quotes) else n(in quotes) :");
18 //end
19 x=[1 0 1 1;
20     0 0 1 0;
21     0 1 1 1;
22     1 1 0 1];
23 l=0;
24 //checks every row and find the max no of ones in a
   row.
25 for i=1:m
26     k=0;
27     for j=1:m
28         if(x(i,j)==1)
29             k=k+1;
30         end
31     end
32     if(k>1)
33         l=k;
34     end
35 end
36 disp(1,"Row multiplicity =");
37 l=0;

```

```

38 //checks every column and find the max no of ones in
    a column.
39 for j=1:m
40     k=0;
41     for i=1:m
42         if(x(i,j)==1)
43             k=k+1;
44         end
45     end
46     if(k>1)
47         l=k;
48     end
49 end
50 disp(1,"Column multiplicity =");

```

Scilab code Exa 6.6 Symmetric

```

1 clc;
2 clear;
3 disp("f(x,y,z)=x^y^z+xy^z^x+x^yz^");
4 disp("since interchanging any two variables gives us
    the same equation");
5 disp("for example interchange x and y");
6 disp("f(y,x,z)=y^x^z+yx^z^y+y^xz^");
7 disp("f(x,y,z)=f(y,x,z)");
8 disp("So the function f(x,y,z) is symmetric");

```

Scilab code Exa 6.7 Symmetric 2

```

1 clc;
2 clear;
3 disp("f(x1,x2,x3)=x1^x2^x3^+x1x2^x3+x1^x2x3");

```

```
4 disp("you can check that above equatioin is not
      symmetric in x1,x2,x3");
5 disp("But it is symmetric in x1,x2,x3^");
6 disp("let us verify it with interchanging x1 and x3^
      ");
7 disp("f(x3^,x2,x1)=x3x2^x1+x3^x2^x1^+x3x2x1^");
8 disp("you can identify that f(x1,x2,x3)=f(x3^,x2,x1)
      ");
```

Chapter 7

Threshold Logic

Scilab code Exa 7.1 weighted Sum

```
1  clc;
2  clear;
3  //takes the input and check whether it is valid or
   not
4  x1=input("x1 = ");
5  while(x1~=0 & x1~=1)
6      disp("enter a valid logical level");
7      x1=input("x1 = ");
8  end
9  x2=input("x2 = ");
10 while(x2~=0 & x2~=1)
11     disp("enter a valid logical level");
12     x2=input("x = ");
13 end
14 x3=input("x3 = ");
15 while(x3~=0 & x3~=1)
16     disp("enter a valid logical level");
17     x3=input("x3 = ");
18 end
19 f=-x1+(2*x2)+x3;
20 if(f>0.5) then
```

```

21     f=1;
22 else
23     f=0;
24 end
25 disp(f,"output y is");
26 m=1;
27 //displays the output of the above expression for
    all the combinations of inputs.
28 for x=0:1
29     for y=0:1
30         for z=0:1
31             f1(m,1)=x;
32             f1(m,2)=y;
33             f2(m,3)=z;
34             f1(m,4)=-x+(2*y)+z;
35             if(f1(m,4)>0.5) then
36                 f1(m,5)=1;
37             else
38                 f1(m,5)=0;
39             end
40             m=m+1;
41         end
42     end
43 end
44 disp("    x1    x2    x3    sum    y");
45 disp(f1)

```

Scilab code Exa 7.2 Inequalities

```

1  clc;
2  clear;
3  y='y';
4  i=1;
5  //Takes the equivalent decimal value of the min
    terms for eg:  $x^y z = 011 = 3$ 

```

```

6 while(y=='y')
7     disp("enter the minterm of a 3 variable function
           ");
8     x(i)=input(": ");
9     while(x(i)>7)
10        disp("enter a valid minterm");
11    end
12    disp("press y if you want to enter more min
           terms else n :");
13    y=input("");
14    i=i+1;
15 end
16 a=1;
17 //Generating truth table for determining the
    inequalities
18 for i=0:1
19     for j=0:1
20         for k=0:1
21             for z=1:length(x)
22                 if(x(z)==a-1);
23                     f(a,4)=1;
24                 end
25             end
26             f(a,1)=i;
27             f(a,2)=j;
28             f(a,3)=k;
29             a=a+1;
30         end
31     end
32 end
33 //displaying the truth table
34 disp("    x1    x2    x3    f");
35 disp(f);
36 disp("");
37 a=1;
38 //generating inequalities
39 for i=0:1
40     for j=0:1

```

```

41         for k=0:1
42             if(f(a,4)==1)
43                 printf('%3d * w1 + %3d * w2 + %3d *
44                     w3>=T',f(a,1),f(a,2),f(a,3))
45                 disp("")
46             else
47                 printf('%3d * w1 + %3d * w2 + %3d *
48                     w3<T',f(a,1),f(a,2),f(a,3))
49                 disp("")
50             end
51         end
52     end
53     disp("By solving the above inequalities we can get
54         the values of weights and T");

```

Scilab code Exa 7.3 Unate Functions

```

1  clc;
2  clear;
3  disp("Given function is f=x1x2^+x2x3^");
4  disp("Since x1 has no complemented form in the above
5      function f,f is positive in x1");
6  disp("x2 has both complemented and uncomplemented
7      forms in f so f is not unate in x2");
8  disp("x3 is only in complemented form so f is
9      negative in x3");

```

Scilab code Exa 7.4 three cube representation

```

1  clc;
2  clear;

```

```

3 disp("given function is f=x1^x2+x2x3^");
4 disp("Since the variables x1 and x3 are only in their
    complemented form f is negative and unate in x1
    and x3");
5 disp("even x2 is only in its uncomplemented form so
    f is positive in x2");

```

Scilab code Exa 7.5 True Vertex

```

1 clc;
2 clear;
3 n=input("Enter the no of input variables :");
4 //Input the true minimal vertices
5 v=input("Enter the no of minimal true vertices :");
6 disp("vertex will be in the form of 101 if it is 3
    variable");
7 for i=1:v
8     printf('Vertex %3d :',i)
9     s(i)=input(" ");
10 end
11 tv=input("enter a vertex which you want find whether
    true vertex or not");
12 //determines whether the vertex is a true or not by
    comparing it with the true minimal vertices
13 for i=1:v
14     if(tv>s(i))
15         disp("It is a true vertex");
16         break;
17     else
18         if(i==v)
19             disp("It is not a true vertex since it
                is not > than any of the min
                vertices");
20         end
21     end

```


22 end

Chapter 8

Reliable Design And Fault Diagnosis

Scilab code Exa 8.1 NOR Logic Circuit

```
1 clc;
2 clear;
3 disp("To find whether h is s-a-0 or not");
4 disp("First we have to express f as function of h
   and x1,x2,x3,x4");
5 disp("Simplifying the given logical circuit we
   obtain the expression of output as");
6 disp("f(X,h) = x1x2x3x4 + h(x2x3 + x1^x2^x3^x4^)");
7 disp("Here h = x2^x3^");
8 disp("let G = x1x2x3x4 & H = h(x2x3 + x1^x2^x3^x4^)"
   );
9 disp("Taking derivative on both sides");
10 disp("df/dh=G^dH/dh");
11 disp("Applying the formula d[f(X)+g(X)]/dx=f^(X)dg(
   X)/dx exor g^(X)df(X)/dx exor df(X)/dx.dg(X)/dx")
   ;
12 disp("df/dh=G^dH/dh=(x1^+x2^+x3^+x4^)(x2x3+x1^x2^x3^
   x4^)");
13 disp("df/dh=x1^x2x3+x2x3x4^+x1^x2^x3^x4^");
```

```

14 disp("To find tests for h s-a-0 we compute the
      values for which h(df/dh)=1");
15 disp(" hdf/dh=x2^x3^(x1^x2x3+x2x3x4^+x1^x2^x3^x4^)");
16 disp(" hdf/dh=x1^x2^x3^x4^");
17 disp("Thus the test set for h s-a-0 is X(0,0,0,0)");

```

Scilab code Exa 8.2 Path Sensitizing

```

1  clc;
2  clear;
3  disp("To prove whether there possibily of
      identifying the fault by sensitising just a single
      path");
4  disp("Let us sensitize the path G3 G6 G9");
5  disp("This requires G6=1 , G10=0 , G11=0 ,G8=0 ");
6  disp("Which inturn requires x2=0 and x3=0 (since G6
      =1)");
7  disp("G10=0 impiles that x4=1 regardless of whether
      there is a fault or not");
8  disp("G11=0 implies G7=1(since x3=0) which in turn
      implies x4=0");
9  disp("Evidently to satisfy both G10=0 and G11=0 we
      must set conflicting requirements on x4 and thus
      have a contradiction");
10 disp("By symmetry of the circuit it is obvious that
      an attempt to sensitize the path through G10 will
      also fail");
11 disp("Hence the method of one dimensional path
      sensitizing fails to generate the test inputs to
      detect the fault");

```

Scilab code Exa 8.3 Two level OR AND Network

```

1 clc;
2 clear;
3 disp("BY following the procedure discussed in
      Section 8.6");
4 disp("we find that {a}={0 or 2 or 3,9,15}");
5 disp("and {b}={7,8,11,13,14}");
6 disp("Thus the minimal sets of tests for the above
      network are ");
7 disp("{T}={0 or 2 or 3,7,8,9,11,13,14,15}");
8 disp("In this case the s-tests and b-tests
      correspond to s-a-0 and s-a-1 respectively");

```

Scilab code Exa 8.4 Various errors Summarized

```

1 clc;
2 clear;
3 disp("Various errors associated with some commonly
      used gates are summarised below");
4 disp("A 0-to-1 error in an AND gate with R identical
      inputs is subcritical");
5 disp("since the output depends on the other input so
      it is subcritical");
6 disp("If the same 0-to-1 error is in a OR gate with
      R identical inputs then it is critical error");
7 disp("since an error input struck at 1 will make the
      output of OR gate to 1 all the time");
8 disp("Similarly the critical error of AND gate is
      subcritical error of OR gate");
9 disp("Similarly the error classification in various
      gates is mentioned in Table 8-2");

```

Chapter 10

Capabilities Minimization and transformation Of Sequential Machines

Scilab code Exa 10.1 Reducing machine

```
1  clc;
2  clear;
3  disp("      ...                NS                ");
4  disp("      PS                x=0                x=1                ");
5  disp("      A                E,0                C,0                ");
6  disp("      B                C,0                A,0                ");
7  disp("      C                B,0                G,0                ");
8  disp("      D                G,0                A,0                ");
9  disp("      E                F,1                B,0                ");
10 disp("      F                E,0                D,0                ");
11 disp("      G                D,0                G,0                ");
12 disp("Initial partition consists of all the states")
    ;
13 disp("P0=(ABCDEFG)");
14 disp("Since E only has 1 equivalent different from
    others ,it can be partitioned from others");
15 disp("P1=(ABCDFG)E");
```

```

16 disp("now check for 2-equivalent i.e. 1-equivalent
    and their Ii sucesors for all possible Ii are
    also 1 equivalent");
17 disp("A,F dont satisfy the 2-equivalent so they can
    be partitioned from others");
18 disp("P2=(AF) (BCDG)E");
19 disp("In the similar manner P3 can be obtained by
    splitting BD and CG since they dont have the same
    3 equivalent");
20 disp("P3=(AF) (BD) (CG)E");
21 disp("In the same way P4 can be obtained by
    splitting A and F");
22 disp("P4=(A) (F) (BD) (CG)E");
23 disp("P4 cant be splitted more so P5 will be the
    same and the partition stops");
24 disp("P5=(A) (F) (BD) (CG) (E)");
25 disp("so the reduced machine is");
26 disp("    ...                NS                ");
27 disp("    .. PS                x=0                x=1 ");
28 disp("    A-----a                e,0                c,0 ");
29 disp("    F-----f                e,0                b,0 ");
30 disp("    .(BD)---b                c,0                a,0 ");
31 disp("    (CG)---c                b,0                c,0 ");
32 disp("    E-----e                f,1                b,0 ");

```

Scilab code Exa 10.2 Compaitability Graph

```

1 clc;
2 clear;
3 disp("Considering the merger graph of machine M6 in
    PG 339");
4 disp("A set of compaitabilities is said to be closed
    if and only if for every compaitable contained
    in the set and all its implied compaitabilities
    are also in the same set");

```

```
5 disp("A closed set of compaitables which contains
      all the states of M is called a closed covering")
  ;
6 disp("By observing the merger graph we can find that
      {(AD)(BE)(CD)} is a closed set ");
7 disp("And the set {(AB)(CD)(EF)} appears to be a
      closed covering");
```

Chapter 12

Structure Of Sequential Machines

Scilab code Exa 12.4 Closed Partitions

```
1  clc;
2  clear;
3  disp("we know that a partition pi on the set of
      states of a Sequential machine M is said to be
      closed if ,for every two states Si and Sj which
      are in the same block of pi and any Ith input
      successor of Si and Sj are also in the same group
      ");
4  disp("Based on the above definition we can make
      seven closed partitions as below");
5  disp("**Parition 1**");
6  disp("{A,B,C,D,E,F,G,H}")
7  disp("**Parition 1**");
8  disp("{(ABCD) (EFGH)}");
9  disp("**Parition 2**");
10 disp("{(ADEH) (BCFG)}")
11 disp("**Parition 3**");
12 disp("{(AD) (BCFG) (EH)}");
13 disp("**Parition 4**");
```



```

14 disp(" {(ADEH) (BC) (FG)}")
15 disp("** Parition 5**");
16 disp(" {(AD) (BC) (EH) (FG)}")
17 disp("** Parition 6**");
18 disp(" {(ABCCDEFGH)}");
19 disp("By assigning values from 000 to 111 to all the
      states from A to H and obtaining the functions
      for Y1,Y2,Y3 and z will result in this equations"
      );
20 disp("Y1=x^y1^");
21 disp("Y2=x^y2+xy2^");
22 disp("Y3=xy2+x^y1^y2y3^+y3^y2^y3+y1y2y3+x^y1y2^y3^")
      ;
23 disp("z=y1^y2^y3");

```

Scilab code Exa 12.5 Output Consistent

```

1 clc;
2 clear;
3 disp("For the 4 state machine M4 in PG 397");
4 disp("IF the four states A,B,C,D are assigned as
      00,01,10,11(ta={ (AC) (BD) }) and 00,01,11,10(tb={ (
      AD) (BC) })");
5 disp("And if the output ,next state functions of both
      the assignments are calculated we can find that
      ");
6 disp("For first assignment ");
7 disp("Y1=x^y1+xy1^");
8 disp("Y2=x^y2^+y1^y2^+xy1y2");
9 disp("z=x^y1^y2^+x^y1y2+xy1^y2+xy1y2^");
10 disp("Second Assignment ");
11 disp("Y1=x^y1+x^y1");
12 disp("Y2=x^y2^+xy1^y2+y1y2^");
13 disp("z=x^y2^+xy2");
14 disp("Since the second assignment results in reduced

```

output expression this partition is called as
output-consistent partition”)

Scilab code Exa 12.7 Dependence

```
1 clc;
2 clear;
3 disp("A partition pi0 on the states of a machine M
      is said to be output consistent if for every
      block pi0 and every input, all the states
      contained in the block have the same outputs");
4 disp("Considering the above definition It can be
      understood that");
5 disp("pi0={(AD) ,(BC)} is an output consistent
      partition of machine M4");
6 disp("since the outputs of A,D states for any inputs
      are same and similarly the outputs of B,C are
      same.");
7 disp("So the partition {(AD) ,(BC)} are said to be
      output consistent");
```

Scilab code Exa 12.8 input Consistent Matrix

```
1 clc;
2 clear;
3 disp("For the Machine M5 in PG 400");
4 disp("State A implies the identification of states C
      and D");
5 disp("similarly C implies the identification of E
      and F states");
6 disp("In the same way C implies the identification
      of A and B");
```

```

7 disp("Thus the smallest input consistent partition
    for M5 is ");
8 disp("pi={{(AB) ,(CD) ,(EF)}}");
9 disp("Any other partition that contains pi is also
    input consistent")

```

Scilab code Exa 12.9 Implementation of Input Consistent matrix

```

1 clc;
2 clear;
3 disp("In previously problems we have determined the
    input and output consistent partitions for the
    Machine M5");
4 disp("Input consistent partition {{(AB) ,(CD) ,(EF)}}");
5 disp("Output consistent partition {{(ACE) ,(BDF)}}");
6 disp("By assigning 000 to 101 to all the states from
    A to F");
7 disp("we can find the expressions for the next state
    and the output");
8 disp("Y1=y2");
9 disp("Y2=y1^y2^");
10 disp("Y3=xy3+xy2+x^y2^y3^+y2y3");
11 disp("z=xy3^");

```

Scilab code Exa 12.10 Autonomous Clock

```

1 clc;
2 clear;
3 disp("Partition pi of the machine M5 is {{(AB) ,(CD) ,(
    EF)}}");
4 disp("If M5 posses a closed partition pi such that
    pi> i .if the autonomous clock has #pi states

```

```

    the period p will be less than or equal to #pi ")
;
5 disp("since in the above closed partition of M5 we
    have 3 states i.e. (AB) as one state and (CD) and
    (EF) as other two");
6 disp("So the periodof the autonomous clock is p=3")

```

Scilab code Exa 12.11 partition pair

```

1 clc;
2 clear
3 disp("From the definition of partition pair(T,T^
    which is an ordered pair of partitions such that
    if Si and Sj are in the same block of t, then for
    every input Ik in I, next states are in the same
    block of T^ ");
4 disp("By applying the above definition we can obtain
    the following partition pairs");
5 disp("(pi1 , pi1 ^)={({(ABC) ,(DEF) }},{(ABC) ,(DEF) })");
6 disp("(t1 , t1 ^)={({(ABCD) ,(EF) }},{(AE) ,(BCDF) })");
7 disp("(t2 , t2 ^)={({(AE) ,(BCDF) }},{(ACDE) ,(BF) })");

```

Scilab code Exa 12.12 partition pair 2

```

1 clc;
2 clear;
3 //1 is equivalent to A ad 2 to B and so on.
4 p1=['A' 'D']; //pairs in partition t(1 and 4
    represents that 1st and 4th state are in the same
    block);
5 p2=['C' 'E'];
6
7 q1=['A' 'E']; //pairs in partition t^

```

```

8 q2=[ 'B' 'D'];
9 q3=[ 'C' 'F'];
10
11 disp("the following are the partitions of the
    machine M8");
12 disp("T");
13 disp('F',p2(2),p2(1), 'B',p1);
14 disp("T^");
15 disp(q3,q2,q1);
16 disp("-----");
17 disp("T");
18 disp('F',p2, 'B',p1);
19 disp("T^");
20 disp(q3,[q1 q2]);
21 disp("-----");

```

Scilab code Exa 12.14 State Consistent Partitions

```

1 clc;
2 clear;
3 //assume the first cloumn values are of machine M1
    and 2nd column are of M2
4 p=[1,1;1 3;2 2;2 4;3 3;3 1;4 4;4 2];
5 z=1;
6 for i=1:length(p(:,1))
7     for j=i:length(p(:,1))
8         if(p(i,1)==p(j,1) & i~=j)
9             q(z,:)=p(i,:) p(j,:);
10            z=z+1;
11        end
12    end
13 end
14 disp("pi(R)");
15 disp(q);
16 z=1;

```

```
17 for i=1:length(p(:,1))
18     for j=i:length(p(:,1))
19         if(p(i,2)==p(j,2) & i~=j)
20             q(z,:)= [p(i,:) p(j,:)];
21             z=z+1;
22         end
23     end
24 end
25 disp(" pi(S)");
26 disp(q);
```

Chapter 13

State Identificaiton And Fault Detection Experiments

Scilab code Exa 13.1 Fault DetectionExeriment

```
1 clc;
2 clear;
3 disp("By following the general procedure for fault
  detection in the text book Page n.o 478");
4 disp("According to 3 & 4 steps of general procedure
  all the inputs are taken as 0 and the output is
  checked]");
5 disp("X:      0  0  0");
6 disp("      A  B  A  B");
7 disp("Z:      1  0  1");
8 disp("By following the step 5");
9 disp("X:      1  0");
10 disp("      B  B  A");
11 disp("Z:      0  0");
12 disp("Accoring to step 6");
13 disp("X:      1  0  0  1  0  0");
14 disp("      A  D  D  D  C  D  D");
15 disp("Z:      0  3  3  1  2  3");
16 disp("Step 7 & 8")
```

```

17 disp("X:      1  1  0");
18 disp("      D  C  A  B");
19 disp("Z:      1  1  1");
20 disp("Whole checking experiment")
21 disp("X:      0  0  0  1  0  1  0  0  1  0  0  1  1  0
");
22 disp("      A  B  A  B  B  A  D  D  D  C  D  D  C  A
      B");
23 disp("Z:      1  0  1  0  0  0  3  3  1  2  3  1  1
      1");

```

Chapter 14

Memory Definiteness Information Losslessness of Finite Automata

Scilab code Exa 14.2 Synchronizing tree

```
1  clc;
2  clear;
3  N=4;
4  //state table of the machine
5  s=[ 'PS' 'x=0' 'x=1';
6      'A' 'A' 'B';
7      'B' 'C' 'B';
8      'C' 'A' 'D';
9      'D' 'C' 'B'];
10
11 k=1;l=2;m=1;n=1;
12 f(k,4)='ABCD';
13 k=k+1;
14 for i=2:max(size(s(:,1)))
15     for j=i:max(size(s(:,1)))
16         if(s(i,2)==s(j,2) & i~=j)
17             f(k,1)=strcat([s(i,1) s(j,1)]);
```

```

18         l=l+4;
19         if(s(i,2)==s(j,2))
20             g(n,m)=s(i,2);
21             g(n,m+1)='-';
22             m=m+2;
23         else
24             g(n,m)=s(i,2);
25             g(n,m+1)=s(j,2);
26             g(n,m+2)='-';
27             m=m+3;
28         end
29         if(s(i,3)==s(j,3))
30             g(n,m)=s(i,3);
31             g(n,m+1)='-';
32             m=m+2;
33         else
34             g(n,m)=strcat([s(i,3) s(j,3)]);
35             g(n,m+1)='-';
36             m=m+2;
37             for z=2:max(size(s(:,1)))
38                 if(s(z,3)==s(z,1))
39                     h(1,3)=s(z,2);
40                     h(1,5)=s(z,3);
41                 end
42             end
43         end
44     end
45 end
46 end
47 disp("Synchronizing tree for Machine M3 is ")
48 disp(f);
49 disp(g);
50 disp(h);

```

Scilab code Exa 14.3 Contracted State Machine

```

1  clc;
2  clear;
3  s=['PS' 'x=0' 'x=1'];
4      'A' 'A' 'B';
5      'B' 'E' 'B';
6      'C' 'E' 'F';
7      'D' 'E' 'F';
8      'E' 'A' 'D';
9      'F' 'E' 'B'];
10 z=0;
11 disp("Original Machine M4");
12 disp(s);
13 p=s;
14 n=max(size(s(:,1)));
15 for i=2:n
16     for j=i:n
17         if(s(i,2)==s(j,2) & s(i,3)==s(j,3) & i~=j)
18             z=z+1;
19             for k=j:n-1
20                 p(k,:)=p(k+1,:);
21             end
22             for k=2:n-1
23                 if(p(k,2)==s(j,1))
24                     p(k,2)=s(i,1);
25                 end
26                 if(p(k,3)==s(j,1))
27                     p(k,3)=s(i,1);
28                 end
29             end
30         end
31     end
32 end
33 s=p(1:n-z,:);
34 disp("The contracted Table M4");
35 disp(s);
36 p=s;
37 n=n-z;
38 z=0;

```

```

39 for i=2:n
40     for j=i:n
41         if(s(i,2)==s(j,2) & s(i,3)==s(j,3) & i~=j)
42             z=z+1;
43             for k=j:n-1
44                 p(k,:)=p(k+1,:);
45             end
46             for k=2:n-1
47                 if(p(k,2)==s(j,1))
48                     p(k,2)=s(i,1);
49                 end
50                 if(p(k,3)==s(j,1))
51                     p(k,3)=s(i,1);
52                 end
53             end
54         end
55     end
56 end
57 s=p(1:n-z,:);
58 disp("Repeated Contractions of M4");
59 disp(s);
60 p=s;
61 n=n-z;
62 z=0;
63 for i=2:n
64     for j=i:n
65         if(s(i,2)==s(j,2) & s(i,3)==s(j,3) & i~=j)
66             z=z+1;
67             for k=j:n-1
68                 p(k,:)=p(k+1,:);
69             end
70             for k=2:n-1
71                 if(p(k,2)==s(j,1))
72                     p(k,2)=s(i,1);
73                 end
74                 if(p(k,3)==s(j,1))
75                     p(k,3)=s(i,1);
76                 end

```

```

77         end
78     end
79 end
80 end
81 s=p(1:n-z,:);
82 disp(s);
83 p=s;
84 n=n-z;
85 z=0;
86 for i=2:n
87     for j=i:n
88         if(s(i,2)==s(j,2) & s(i,3)==s(j,3) & i~=j)
89             z=z+1;
90             for k=j:n-1
91                 p(k,:)=p(k+1,:);
92             end
93             for k=2:n-1
94                 if(p(k,2)==s(j,1))
95                     p(k,2)=s(i,1);
96                 end
97                 if(p(k,3)==s(j,1))
98                     p(k,3)=s(i,1);
99                 end
100            end
101        end
102    end
103 end
104 s=p(1:n-z,:);
105 disp(s);

```

Scilab code Exa 14.4 Testing Table

```

1  clc;
2  clear;
3  N=4;

```

```

4 s=[ 'PS' 'x=0' 'x=1';
5     'A' 'A' 'B';
6     'B' 'C' 'B';
7     'C' 'A' 'D';
8     'D' 'C' 'B'];
9 n=max(size(s(:,1)));
10 s(n+1,:)=['-' '-' '-'];
11 l=2;
12 for i=2:n
13     for j=i:n
14         if(i~=j)
15             s(n+1,1)=strcat([s(i,1) s(j,1)]);
16             s(n+1,2)=strcat([s(i,2) s(j,2)]);
17             s(n+1,3)=strcat([s(i,3) s(j,3)]);
18             l=l+1;
19         end
20     end
21 end
22 disp(s);

```

Scilab code Exa 14.6 Testing graph

```

1 clc;
2 clear;
3 s=['PS' 'x=0' 'x=1';
4   'A' 'A' 'B';
5   'B' 'C' 'D';
6   'C' 'D' 'C';
7   'D' 'B' 'A'];
8 q=[0 0;0 0;1 1;1 1];
9 n=max(size(s(:,1)));
10 l=1;
11 for i=2:n
12     z=0;x=0;
13     for j=i:n

```

```

14         if(q(i-1,1)==0 & q(i-1,2)==0 & i~=j & z==0)
15             p(1,1)=s(i,1);
16             p(1,2)=strcat([s(i,2) s(i,3)]);
17             p(1,3)='-';
18             l=l+1;z=z+1;
19         else
20             if(q(i-1,1)==1 & q(i-1,2)==1 & x==0)
21                 p(1,1)=s(i,1);
22                 p(1,2)='-';
23                 p(1,3)=strcat([s(i,2) s(i,3)]);
24                 l=l+1;x=x+1;
25             end
26         end
27     end
28 end
29 disp(['PS' 'z=0' 'z=1']);
30 disp(p);
31 disp(['AB' '(AC) (AD) (BC) (BD)' '-'];
32       'CD' '- '(AC) (AD) (BC) (BD)']);

```

Scilab code Exa 16.7 Testing Table 2

```

1  clc;
2  clear;
3  s=['S' 'C1' 'SB1';
4     'B1' 'S' '-';
5     'C1' 'C2' '-';
6     'C2' '- 'S'];
7  n=max(size(s(:,1)));
8  l=1;
9  for i=1:n
10     for j=i:n
11         if(i~=j)
12             p(1,1)=strcat([s(i,1) s(j,1)]);
13             if(s(i,2)=='-' | s(j,2)=='-')

```

```
14         p(1,2)='-';
15     else
16         p(1,2)=strcat([s(i,2) s(j,2)]);
17     end
18     if(s(i,3)=='-' | s(j,3)=='-')
19         p(1,3)='-';
20     else
21         p(1,3)=strcat([s(i,3) s(j,3)]);
22     end
23     l=l+1;
24     end
25 end
26 end
```

Chapter 15

Linear Sequential Machines

Scilab code Exa 15.3 Transfer function

```
1  clc;
2  clear;
3  T1=[1 2 1];
4  T2=[0 1 1];
5  n=max(size(T1));
6  a=n-1;
7  b=a;
8  T3=zeros(1,a+b+1);
9  for i=1:n
10     b=n-1;
11     for j=1:n
12         T3(a+b+1)=T3(a+b+1)+(T1(i)*T2(j));
13         b=b-1;
14     end
15     a=a-1;
16 end
17 m=max(size(T3));
18 i=1;
19 j=m;
20 while(i<=floor(m/2) & j>=(ceil(m/2)+1))
21     t=T3(i);
```

```

22         T3(i)=modulo(T3(j),3);
23         T3(j)=modulo(t,3);
24         i=i+1;
25         j=j-1;
26     end
27     T3(ceil(m/2))=modulo(T3(ceil(m/2)),3);
28     disp("Top row of the below matrix is powers of the
          transfer function and bottom row is of
          coefficients of respective powers");
29     disp([4 3 2 1 0]);
30     disp(T3);

```

Scilab code Exa 15.4 Impulse response

```

1  clc;
2  clear;
3  T1=[1 0 1 1];
4  h=[1 1 0 1 0 0 0 0];
5  in=[1 0 1 1];
6  l=1;
7  for i=2:max(size(in(1,:)))
8      h1=h;
9      if(in(i)==1)
10         for k=1:i-1
11             t=h1(max(size(h1(1,:)))));
12             for j=max(size(h1(1,:)))-1:-1:1
13                 h1(j+1)=h1(j);
14             end
15             h1(1)=t;
16         end
17         D(1,:)=h1(1,:);
18         l=l+1;
19     end
20 end
21 output=h+D(1,:)+D(2,:);

```

```
22 output=modulo(output,2);
23 disp(output);
```

Scilab code Exa 15.5 Null Sequence

```
1  clc;
2  clear;
3  //T=1+D+D^3 over GF(2)
4  //let X0 is the null sequence
5  //0=X0+DX0+D^3X0 since output of the null sequence
   is zero
6  //Adding X0 on both sides we get X0=DX0+D^3X0
7  nul=[0 0 1];
8  T=[0 1 0 1];
9  j=1;
10 for i=1:max(size(T));
11     if(T(i)==1)
12         s(1,j)=i-1;
13         j=j+1;
14     end
15 end
16 j=4;
17 for i=1:7
18     nul(1,j)=nul(1,j-s(1))+nul(1,j-s(2));
19     nul(1,j)=modulo(nul(1,j),2);
20     j=j+1;
21 end
22 disp("Null Sequence for input 001 is ");
23 disp(nul);
```

Scilab code Exa 15.6 Null Sequence 2

```
1  clc;
```

```

2 clear;
3 //T=1+2D^2+D^3 over GF(3)
4 //let X0 is the null sequence
5 //0=X0+2D^2*X0+D^3*X0 since output of the null
   sequence is zero
6 //Adding X0 on both sides we get 2X0=2D^2*X0+D^3*X0
7 //X0=D^2*X0+2D^3*X0
8 nul=[1 1 1];
9 T=[0 0 1 2];
10 j=1;
11 for i=1:max(size(T));
12     if(T(i)>=1)
13         s(1,j)=i-1;
14         j=j+1;
15     end
16 end
17 j=4;
18 for i=1:15
19     nul(1,j)=nul(1,j-s(1))+2*nul(1,j-s(2));
20     nul(1,j)=modulo(nul(1,j),3);
21     j=j+1;
22 end
23 disp("Null Sequence for input 001 is ");
24 disp(nul);

```

Scilab code Exa 15.10 Reducible

```

1 clc;
2 clear;
3 A=[0 1 1;
4     1 0 0;
5     1 0 0];
6 B=[1;1;0];
7 C=[1 1 0;
8     1 1 1];

```

```

9 D=[0;1];
10 K=[C;C*A;C*(A*A)];
11 K=modulo(K,2);
12 disp("K matrix")
13 disp(K);
14 disp(rank(K),'since the rank of K is ');
15 disp("K is not further reducible");
16 //z=K * y;

```

Scilab code Exa 15.11 Linear Machine

```

1 clc;
2 clear;
3 A=[0 1 0;
4     1 0 0;
5     0 1 1];
6 B=[1;1;1];
7 C=[1 0 0];
8 D=[1];
9 K=[C;C*A;C*(A*A)];
10 disp(K);
11 disp(rank(K),'rank of K matrix is ');
12 disp("Since the rank is 2 K matrix can be reduced");
13 disp("As only the first 2 rows are linearly
14     independent therefore");
14 T=[1 0 0;0 1 0];
15 disp(T,'T =');
16 disp("In this case Q is");
17 Q=[1 0;0 1];
18 disp(Q);
19 Q_inv=inv(Q);
20 R=T';
21 //y_red=T*y;
22 A_red=T*A*R;
23 B_red=T*B;

```

```
24 C_red=C*R;
25 D_red=D;
```

Scilab code Exa 15.12 Linear Machine 2

```
1  clc;
2  clear;
3  A=[1 0 0 0;
4     0 0 1 1;
5     1 1 0 0;
6     1 0 1 0];
7  B=[1 0;
8     0 0;
9     1 1;
10    1 1];
11 C=[0 1 0 1;
12    1 1 1 0];
13 D=[1 0;
14    0 1];
15 K=[C;C*A;C*(A*A)];
16 K=modulo(K,2);
17 T=K(1:3,:);
18 Q=K(1:3,1:3);
19 Q_inv=inv(Q);
20 Q_inv=abs(modulo(Q_inv,2));
21 R=[Q_inv;[0 0 0]];
22 disp("Reduced matrix A is");
23 disp(A_red=T*A*R);
24 disp("Reduced matrix B is");
25 disp(B_red=T*B);
26 disp("Reduced matrix C is");
27 disp(C_red=C*R);
28 disp("Reduced matrix D is");
29 disp(D_red=D);
```

Scilab code Exa 15.15 Commutative ring

```
1  clc;
2  clear;
3  n=4;
4  for i=1:4
5      for j=1:4
6          p(i,j)=modulo(i+j-2,4);
7      end
8  end
9  disp("modulo 4 Addition");
10 disp(p);
11 for i=1:4
12     for j=1:4
13         p(i,j)=modulo((i-1)*(j-1),4);
14     end
15 end
16 disp("modulo 4 Multiplication");
17 disp(p);
```

Scilab code Exa 15.16 Identifying Whether the Ring is a Field Or not

```
1  clc;
2  clear;
3  n=4;
4  a=zeros(1,n);
5  for i=1:n
6      for j=1:n
7          pad(i,j)=modulo(i+j-2,4);
8      end
9  end
10 for i=1:n
```

```

11     for j=1:n
12         pmu(i,j)=modulo((i-1)*(j-1),4);
13     end
14 end
15 t=0;
16 for i=1:n
17     for j=1:n
18         if(pad(i,j)==1)
19             t=1;
20             break;
21         end
22     end
23 end
24 if(t==0)
25     disp("Modulo 4 ring is not a field");
26 end
27 for i=0:3
28     for j=0:3
29         if(modulo(i*j,4)==1)
30             a(i)=1;
31         end
32     end
33 end
34 res=0;
35 for i=1:max(size(a))
36     if(a(i)==1)
37         res=res+1;
38     end
39 end
40 if(res==4)
41     disp("Modulo 4 ring is a field");
42 else
43     disp("Modulo 4 ring is not a field");
44 end

```

Scilab code Exa 15.17 Finite Field

```
1  clc;
2  clear;
3  n=3;
4  for i=1:n
5      for j=1:n
6          p(i,j)=modulo(i+j-2,3);
7      end
8  end
9  disp("modulo 3 Addition");
10 disp(p);
11 for i=1:n
12     for j=1:n
13         p(i,j)=modulo((i-1)*(j-1),3);
14     end
15 end
16 disp("modulo 3 Multiplication");
17 disp(p);
```

Chapter 16

Finite State Recognizers

Scilab code Exa 16.2 Prove the identity

```
1 clc;
2 clear;
3 disp("R1=  +1*(011)*(1*(011))*");
4 //from the identity  +RR*=R*  where R=1*(011)*
5 disp("R2=(1+011)*");
6 //from the identity (P+Q)*=(P*Q)*
7 disp("R1=  +1*(011)*(1*(011))*");
8 disp("(1*(011))*");
9 disp("(1+011)*=R2");
```

Scilab code Exa 16.3 Prove the identity 2

```
1 clc;
2 clear;
3 disp("(1+00*1)+(1+00*1)(0+10*1)*(0+10*1)=0*1(0+10*1)
      *");
4 disp("(1+00*1)+(1+00*1)(0+10*1)*(0+10*1)");
5 //by taking (1+00*1) as common
```

```
6 disp("=(1+00*1)[ + (0+10*1)*(0+10*1)]");
7 disp("[( +00*) 1][ + (0+10*1)*(0+10*1)]");
8 //from the identity +RR*=R*
9 disp("[0*1(0+10*1)*]");
```

Appendix

Scilab code AP 1 Kmap Solver with out Dontcares

```
1 function []=karmap(k)
2     n=4;
3     k(:, :, 2)=zeros(n,n);
4     var=['y' 'z' 'w' 'x'];
5     //var=['w' 'x' 'y' 'z'];
6     p1=['y' 'z' 'y' 'z' 'yz' 'yz'];
7     p2=['w' 'x' 'w' 'x' 'wx' 'wx'];
8     cmn4=4;
9     cmn2=2;
10    temp=1;
11    //16 cells
12    for i=1:n
13        for j=1:n
14            if(k(i,j)~=1)
15                temp=0;
16                break;
17            end
18        end
19    end
20    if(temp==1)
21        printf("1");
22        abort;
23    end
24    //8 cells
25    z1=ones(2,4);
26    z2=ones(4,2);
```

```

27     temp1=['00' '01' '11' '10'];
28     temp2=temp1';
29     for i=1:n
30         if(i==4)
31             t=1;
32         else
33             t=i+1;
34         end
35         z=[k(i,:,1);k(t,:,1)];
36         if(z==z1)
37             k(i,:,2)=[1 1 1 1];
38             k(t,:,2)=[1 1 1 1];
39             a=strsplit(temp2(i,1));
40             b=strsplit(temp2(t,1));
41             c=strcmp(a,b);
42             for in=1:max(size(c))
43                 if(c(in)==0 & a(in)=='0')
44                     printf('%s''',var(in));
45                     disp("");
46                     break;
47                 else
48                     if(c(in)==0 & a(in)=='1')
49                         printf(var(in));
50                         disp("");
51                         break;
52                     end
53                 end
54             end
55         end
56     end
57     for j=1:n
58         if(j==4)
59             t=1;
60         else
61             t=j+1;
62         end
63         z=[k(:,j,1) k(:,t,1)];
64         if(z==z2)

```

```

65         k(:,j,2)=[1;1;1;1];
66         k(:,t,2)=[1;1;1;1];
67         a=strsplit(temp1(1,j));
68         b=strsplit(temp1(1,t));
69         c=strcmp(a,b);
70         for in=1:max(size(c))
71             if(c(in)==0 & a(in)=='0')
72                 printf('%s''',var(2+in));
73                 disp("");
74                 break;
75             else
76                 if(c(in)==0 & a(in)=='1')
77                     printf(var(2+in));
78                     disp("");
79                     break;
80                 end
81             end
82         end
83     end
84 end
85 //4 cells
86 z1=ones(1,4);
87 z2=ones(4,1);
88 z3=ones(2,2);
89 temp1=['00' '01' '11' '10'];
90 temp2=temp1';
91 for t=1:n
92     z=k(t,:,1);
93     no=number_of(k(t,:,2),1);
94     if(z==z1 & no<cmn4)
95         k(t,:,2)=z1;
96         a=strsplit(temp1(1,t));
97         for in=1:max(size(a))
98             if(a(in)=='0')
99                 printf('%s''',var(in));
100            end
101            if(a(in)=='1')
102                printf(var(in));

```

```

103             end
104             end
105             disp("");
106         end
107     end
108     for t=1:n
109         z=k(:,t,1);
110         no=number_of(k(:,t,2),1);
111         if(z==z2 & no<cmn4)
112             k(:,t,2)=z2;
113             a=strsplit(temp2(t,1));
114             for in=1:max(size(a))
115                 if(a(in)=='0')
116                     printf('%s''',var(2+in));
117                 end
118                 if(a(in)=='1')
119                     printf(var(2+in));
120                 end
121             end
122             disp("");
123         end
124     end
125     for i=1:n
126         for j=1:n
127             if(i==n)
128                 t1=1;
129             else
130                 t1=i+1;
131             end
132             if(j==n)
133                 t2=1;
134             else
135                 t2=j+1;
136             end
137             z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
138                 ,1)];
139             z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
140                 ,2)];

```

```

139         no=number_of(z5,1);
140         if(z4==z3 & no<cmn4)
141             k(i,j,2)=1;
142             k(i,t2,2)=1;
143             k(t1,j,2)=1;
144             k(t1,t2,2)=1;
145             a=strsplit(temp2(i,1));
146             b=strsplit(temp2(t1,1));
147             c=strcmp(a,b);
148             for in=1:max(size(c))
149                 if(c(in)==0 & a(in)=='0')
150                     printf('%s''',var(in));
151                 end
152                 if(c(in)==0 & a(in)=='1')
153                     printf(var(in));
154                 end
155             end
156             a=strsplit(temp1(1,j));
157             b=strsplit(temp1(1,t2));
158             c=strcmp(a,b);
159             for in=1:max(size(c))
160                 if(c(in)==0 & a(in)=='0')
161                     printf('%s''',var(2+in));
162                 end
163                 if(c(in)==0 & a(in)=='1')
164                     printf(var(2+in));
165                 end
166             end
167             disp('');
168         end
169     end
170 end
171 //2 cells
172 z6=[1 1];
173 z7=z6';
174 for i=1:n
175     for j=1:n
176         if(i==n)

```



```

177         t1=1;
178     else
179         t1=i+1;
180     end
181     if(j==n)
182         t2=1;
183     else
184         t2=j+1;
185     end
186     z8=[k(i,j,1) k(i,t2,1)];
187     z9=[k(i,j,2) k(i,t2,2)];
188     no1=number_of(z9,1);
189     if(z8==z6 & no1<cmn2 & i+j~=2)
190         k(i,j,2)=1;
191         k(i,t2,2)=1;
192         a=strsplit(temp1(1,j));
193         b=strsplit(temp1(1,t2));
194         c=strcmp(a,b);
195         for in=1:max(size(c))
196             if(c(in)==0 & a(in)=='0')
197                 printf(p1(1,i));
198                 printf(' %s ',var(2+in));
199                 disp("");
200             end
201             if(c(in)==0 & a(in)=='1')
202                 printf(p1(1,i));
203                 printf(var(2+in));
204                 disp("");
205             end
206         end
207     end
208 end
209 end
210 for i=1:n
211     for j=1:n
212         if(i==n)
213             t1=1;
214         else

```

```

215         t1=i+1;
216     end
217     if(j==n)
218         t2=1;
219     else
220         t2=j+1;
221     end
222     z10=[k(i,j,1);k(t1,j,1)];
223     z11=[k(i,j,2);k(t1,j,2)];
224     no2=number_of(z11,1);
225     if(z10==z7 & no2<cmn2)
226         k(i,j,2)=1;
227         k(t1,j,2)=1;
228         a=strsplit(temp2(i,1));
229         b=strsplit(temp2(t1,1));
230         c=strcmp(a,b);
231         for in=1:max(size(c))
232             if(c(in)==0 & a(in)=='0')
233                 printf(p2(j,1));
234                 printf('%s''',var(in));
235                 disp("");
236             end
237             if(c(in)==0 & a(in)=='1')
238                 printf(p2(j,1));
239                 printf(var(in));
240                 disp("");
241             end
242         end
243     end
244 end
245 end
246 //single cell
247 for i=1:n
248     for j=1:n
249         if(k(i,j,2)==0 & k(i,j,1)==1)
250             a=strsplit(temp1(1,j));
251             b=strsplit(temp2(i,1));
252             for in=1:max(size(a(:,1)))

```

```

253         if(a(in,1)=='1')
254             printf(var(in+2));
255         else
256             if(a(in,1)=='0')
257                 printf('%s',var(2+in))
                ;
258             end
259         end
260     end
261     for in=1:max(size(b(:,1)))
262         if(b(in,1)=='1')
263             printf(var(in));
264         else
265             if(b(in,1)=='0')
266                 printf('%s',var(in));
267             end
268         end
269     end
270     if(i~=4 & j~=4)
271         disp("");
272     end
273 end
274 end
275 end
276 endfunction

```

Scilab code AP 2 No.of

```

1 // finds the number of z's in the matrix A
2 function res=number_of(a,z)
3     res=0;
4     for i=1:max(size(a(:,1)))
5         for j=1:max(size(a(1,:)))
6             if(a(i,j)==z)
7                 res=res+1;
8             end
9         end
10    end

```

11 `endfunction`

Scilab code AP 3 Knap Solver with out Dontcares

```
1 function []=karmap1(k)
2     n=4;
3     k(:, :, 2)=zeros(n,n);
4     var=['y' 'z' 'w' 'x'];
5     p1=['y' 'z' 'y' 'z' 'yz' 'yz'];
6     p2=['w' 'x' 'w' 'x' 'wx' 'wx'];
7     cmn4=1;
8     cmn2=1;
9     temp=1;
10    //16 cells
11    for i=1:n
12        for j=1:n
13            if(k(i,j)~=1)
14                temp=0;
15                break;
16            end
17        end
18    end
19    if(temp==1)
20        printf("1");
21        abort;
22    end
23    //8 cells
24    z1=ones(2,4);
25    z2=ones(4,2);
26    temp1=['00' '01' '11' '10'];
27    temp2=temp1';
28    for i=1:n
29        if(i==4)
30            t=1;
31        else
32            t=i+1;
33        end
34    z=[k(i, :, 1);k(t, :, 1)];
```

```

35         if(z==z1)
36             k(i,:,2)=[1 1 1 1];
37             k(t,:,2)=[1 1 1 1];
38             a=strsplit(temp2(i,1));
39             b=strsplit(temp2(t,1));
40             c=strcmp(a,b);
41             for in=1:max(size(c))
42                 if(c(in)==0 & a(in)=='0')
43                     printf('%s''',var(in));
44                     disp("");
45                     break;
46                 else
47                     if(c(in)==0 & a(in)=='1')
48                         printf(var(in));
49                         disp("");
50                         break;
51                     end
52                 end
53             end
54         end
55     end
56     for j=1:n
57         if(j==4)
58             t=1;
59         else
60             t=j+1;
61         end
62         z=[k(:,j,1) k(:,t,1)];
63         if(z==z2)
64             k(:,j,2)=[1;1;1;1];
65             k(:,t,2)=[1;1;1;1];
66             a=strsplit(temp1(1,j));
67             b=strsplit(temp1(1,t));
68             c=strcmp(a,b);
69             for in=1:max(size(c))
70                 if(c(in)==0 & a(in)=='0')
71                     printf('%s''',var(2+in));
72                     disp("");

```

```

73         break;
74     else
75         if(c(in)==0 & a(in)=='1')
76             printf(var(2+in));
77             disp("");
78             break;
79         end
80     end
81 end
82     end
83 end
84 //4 cells
85 z1=ones(1,4);
86 z2=ones(4,1);
87 z3=ones(2,2);
88 temp1=['00' '01' '11' '10'];
89 temp2=temp1';
90 for t=1:n
91     z=k(t, :, 1);
92     no=number_of(k(t, :, 2), 1);
93     if(z==z1 & no<cmn4)
94         k(t, :, 2)=z1;
95         a=strsplit(temp1(1,t));
96         for in=1:max(size(a))
97             if(a(in)=='0')
98                 printf('%s' ', ', var(in));
99             end
100             if(a(in)=='1')
101                 printf(var(in));
102             end
103         end
104         disp("");
105     end
106 end
107 for t=1:n
108     z=k(:, t, 1);
109     no=number_of(k(:, t, 2), 1);
110     if(z==z2 & no<cmn4)

```

```

111         k(:,t,2)=z2;
112         a=strsplit(temp2(t,1));
113         for in=1:max(size(a))
114             if(a(in)=='0')
115                 printf('%s''',var(2+in));
116             end
117             if(a(in)=='1')
118                 printf(var(2+in));
119             end
120         end
121         disp("");
122     end
123 end
124 for i=1:n
125     for j=1:n
126         if(i==n)
127             t1=1;
128         else
129             t1=i+1;
130         end
131         if(j==n)
132             t2=1;
133         else
134             t2=j+1;
135         end
136         z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
137             ,1)];
138         z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
139             ,2)];
140         no=number_of(z5,1);
141         if(z4==z3 & no<cmn4)
142             k(i,j,2)=1;
143             k(i,t2,2)=1;
144             k(t1,j,2)=1;
145             k(t1,t2,2)=1;
146             a=strsplit(temp2(i,1));
147             b=strsplit(temp2(t1,1));
148             c=strcmp(a,b);

```

```

147         for in=1:max(size(c))
148             if(c(in)==0 & a(in)=='0')
149                 printf('%s''',var(in));
150             end
151             if(c(in)==0 & a(in)=='1')
152                 printf(var(in));
153             end
154         end
155         a=strsplit(temp1(1,j));
156         b=strsplit(temp1(1,t2));
157         c=strcmp(a,b);
158         for in=1:max(size(c))
159             if(c(in)==0 & a(in)=='0')
160                 printf('%s''',var(2+in));
161             end
162             if(c(in)==0 & a(in)=='1')
163                 printf(var(2+in));
164             end
165         end
166         disp('');
167     end
168 end
169 end
170 //2 cells
171 z6=[1 1];
172 z7=z6';
173 for i=1:n
174     for j=1:n
175         if(i==n)
176             t1=1;
177         else
178             t1=i+1;
179         end
180     if(j==n)
181         t2=1;
182     else
183         t2=j+1;
184     end

```



```

185         z8=[k(i,j,1) k(i,t2,1)];
186         z9=[k(i,j,2) k(i,t2,2)];
187         no1=number_of(z9,1);
188         if(z8==z6 & no1<cmn2)
189             k(i,j,2)=1;
190             k(i,t2,2)=1;
191             a=strsplit(temp1(1,j));
192             b=strsplit(temp1(1,t2));
193             c=strcmp(a,b);
194             for in=1:max(size(c))
195                 if(c(in)==0 & a(in)=='0')
196                     printf(p1(1,i));
197                     printf('%s' , var(2+in));
198                     disp("");
199                 end
200                 if(c(in)==0 & a(in)=='1')
201                     printf(p1(1,i));
202                     printf(var(2+in));
203                     disp("");
204                 end
205             end
206         end
207     end
208 end
209 for i=1:n
210     for j=1:n
211         if(i==n)
212             t1=1;
213         else
214             t1=i+1;
215         end
216         if(j==n)
217             t2=1;
218         else
219             t2=j+1;
220         end
221         z10=[k(i,j,1);k(t1,j,1)];
222         z11=[k(i,j,2);k(t1,j,2)];

```

```

223         no2=number_of(z11);
224         if(z10==z7 & no2<cmn2)
225             k(i,j,2)=1;
226             k(t1,j,2)=1;
227             a=strsplit(temp2(i,1));
228             b=strsplit(temp2(t1,1));
229             c=strcmp(a,b);
230             for in=1:max(size(c))
231                 if(c(in)==0 & a(in)=='0')
232                     printf(p2(j,1));
233                     printf('%s''',var(in));
234                     disp("");
235                 end
236                 if(c(in)==0 & a(in)=='1')
237                     printf(p2(j,1));
238                     printf(var(in));
239                     disp("");
240                 end
241             end
242         end
243     end
244 end
245 //single cell
246 for i=1:n
247     for j=1:n
248         if(k(i,j,2)==0 & k(i,j,1)==1)
249             a=strsplit(temp1(1,j));
250             b=strsplit(temp2(i,1));
251             for in=1:max(size(a(:,1)))
252                 if(a(in,1)=='1')
253                     printf(var(in+2));
254                 else
255                     if(a(in,1)=='0')
256                         printf('%s''',var(2+in))
257                         ;
258                     end
259                 end
260             end
261         end
262     end
263 end

```

```

260         for in=1:max(size(b(:,1)))
261             if(b(in,1)=='1')
262                 printf(var(in));
263             else
264                 if(b(in,1)=='0')
265                     printf('%s',var(in));
266                 end
267             end
268         end
269         if(i~=4 & j~=4)
270             disp("");
271         end
272     end
273 end
274 end
275 endfunction

```

Scilab code AP 4 No.of 3d matrix

```

1 //finds the number of z's in the 3 dimensional
  matrix A
2 function res=noof3(a,z)
3     res=0;
4     for i=1:max(size(a(:,1,1)))
5         for j=1:max(size(a(1,:,1)))
6             for l=1:2
7                 if(a(i,j,l)==z)
8                     res=res+1;
9                 end
10            end
11        end
12    end
13 endfunction

```

Scilab code AP 5 Check

```

1 function out= check(a,b,c,d)
2     for i=1:max(size(a));

```

```

3         if(a(i,1)==b(i,1) & b(i,1)==c(i,1) & c(i,1)
           ==d(i,1))
4             out(i)=0;
5         else
6             out(i)=1;
7         end
8     end
9 endfunction

```

Scilab code AP 6 Decimal to Base 2 Converter

```

1 //dec21bin is a function whcih converts any decimal
   number given to it will output its equivalent
   binary number
2 //pass the decimal number as an argument to the
   function
3 // For eg:dec21bin(10)
4 //Will give an output of 1010
5
6 function [temp]=dec21bin(dec)
7     temp2=floor(dec);
8
9                                     //separating
   integer part from the given number
10    temp4=modulo(dec,1);
11
12                                     //separating
   decimal part from the given number
13
14    format('v',18);
15
16                                     //changing
   the default precision to 18
17
18    i=1;p=0;x=1;
19
20                                     //flag
   bits
21
22    while(temp2>0)
23        //storing each integer digit in vector for
   convenience

```

```

15         p(i)=(modulo(floor(temp2),2))
16         temp2=floor(temp2)/2;
17         i=i+1;
18     end
19
20     temp2=0;
21     //clearing the temporary variable 'temp2'
22     for j=1:length(p)
23         //multiplying the bits of integer part with
24         //their position values and adding
25         temp2=temp2+(p(j)*10^(j-1));
26     end
27     while (temp4~=0)
28         //storing
29         //each integer digit in vector for convenience
30         temp4=temp4*2;
31         d(x)=floor(temp4);
32         x=x+1;
33         temp4=modulo(temp4,1);
34     end
35     temp5=0;
36     //clearing the temporary variable 'temp2'
37     for j=1:x-1
38         //
39         //multiplying the bits of decimal part with
40         //their position values and adding
41         temp5=temp5+(10^(-1*j)*d(j))
42     end
43     temp=temp2+temp5;
44     //
45     //finally adding both the integer and decimal
46     //parts to get total output.
47 endfunction

```

Scilab code AP 7 Kmap with dont cares

```
1 function []=donkmap(k,l)
2     n=4;           //four variable kmap
3     k(:,:,2)=zeros(n,n); //temporary matrix to
4     //know whether a element is paired or not
5     //declaring notations to display output
6     var=['y' 'z' 'w' 'x'];
7     p1=['y' 'z' ' ' 'y' 'z' 'yz' 'yz' ' '];
8     p2=['w' 'x' ' ' 'w' 'x' 'wx' 'wx' ' '];
9     //minimum redundant elements accepted while
10    pairing
11    cmn4=4;
12    cmn2=2;
13    temp=1;
14    printf('f');
15    printf('%1d',l);
16    printf("=");
17    //16 cells
18    for i=1:n
19        for j=1:n
20            if(k(i,j)~=1 | k(i,j)~=2)
21                temp=0;
22                break;
23            end
24        end
25    end
26    if(temp==1)
27        printf("1");
28        abort;
29    end
30    //8 cells
31    z1=ones(2,4);
32    z2=ones(4,2);
33    temp1=['00' '01' '11' '10'];
34    temp2=temp1';
35    for i=1:n
36        if(i==4)
```

```

35         t=1;
36     else
37         t=i+1;
38     end
39     z=[k(i,:,1);k(t,:,1)];
40     if(number_of(z,0)==0 & number_of(z,1)>1)
41         k(i,:,2)=[1 1 1 1];
42         k(t,:,2)=[1 1 1 1];
43         a=strsplit(temp2(i,1));
44         b=strsplit(temp2(t,1));
45         c=strcmp(a,b);
46         for in=1:max(size(c))
47             if(c(in)==0 & a(in)=='0')
48                 printf('%s''',var(in));
49                 printf('+');
50                 break;
51             else
52                 if(c(in)==0 & a(in)=='1')
53                     printf(var(in));
54                     printf('+');
55                     break;
56                 end
57             end
58         end
59     end
60 end
61 for j=1:n
62     if(j==4)
63         t=1;
64     else
65         t=j+1;
66     end
67     z=[k(:,j,1) k(:,t,1)];
68     if(number_of(z,0)==0 & number_of(z,1)>0)
69         k(:,j,2)=[1;1;1;1];
70         k(:,t,2)=[1;1;1;1];
71         a=strsplit(temp1(1,j));
72         b=strsplit(temp1(1,t));

```

```

73         c=strcmp(a,b);
74         for in=1:max(size(c))
75             if(c(in)==0 & a(in)=='0')
76                 printf('%s',' ',var(2+in));
77                 printf('+');
78                 break;
79             else
80                 if(c(in)==0 & a(in)=='1')
81                     printf(var(2+in));
82                     printf('+');
83                     break;
84                 end
85             end
86         end
87     end
88 end
89 //4 cells
90 z1=ones(1,4);
91 z2=ones(4,1);
92 z3=ones(2,2);
93 temp1=['00' '01' '11' '10'];
94 temp2=temp1';
95 for t=1:n
96     z=k(t,:,1);
97     no=number_of(k(t,:,2),1);
98     if(number_of(z,0)==0 & no<cmn4 &
99         number_of(z,1)>0)
100         k(t,:,2)=z1;
101         a=strsplit(temp1(1,t));
102         for in=1:max(size(a))
103             if(a(in)=='0')
104                 printf('%s',' ',var(in));
105             end
106             if(a(in)=='1')
107                 printf(var(in));
108             end
109         end
110     end
111     printf("+" );

```



```

110             end
111         end
112         for t=1:n
113             z=k(:,t,1);
114             no=number_of(k(:,t,2),1);
115             if(number_of(z,0)==0 & no<cmn4 & number_of(z
116                 ,1)>0)
117                 k(:,t,2)=z2;
118                 a=strsplit(temp2(t,1));
119                 for in=1:max(size(a))
120                     if(a(in)=='0')
121                         printf('%s''',var(2+in));
122                     end
123                     if(a(in)=='1')
124                         printf(var(2+in));
125                     end
126                 end
127                 printf("+"");
128             end
129         for i=1:n
130             for j=1:n
131                 if(i==n)
132                     t1=1;
133                 else
134                     t1=i+1;
135                 end
136                 if(j==n)
137                     t2=1;
138                 else
139                     t2=j+1;
140                 end
141                 z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
142                     ,1)];
143                 z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
144                     ,2)];
145                 no=number_of(z5,1);
146                 if(number_of(z4,0)==0 & no<cmn4 &

```

```

145         number_of(z4,1)>0)
146         k(i,j,2)=1;
147         k(i,t2,2)=1;
148         k(t1,j,2)=1;
149         k(t1,t2,2)=1;
150         a=strsplit(temp2(i,1));
151         b=strsplit(temp2(t1,1));
152         c=strcmp(a,b);
153         for in=1:max(size(c))
154             if(c(in)==0 & a(in)=='0')
155                 printf('%s ',' ',var(in));
156             end
157             if(c(in)==0 & a(in)=='1')
158                 printf(var(in));
159             end
160         a=strsplit(temp1(1,j));
161         b=strsplit(temp1(1,t2));
162         c=strcmp(a,b);
163         for in=1:max(size(c))
164             if(c(in)==0 & a(in)=='0')
165                 printf('%s ',' ',var(2+in));
166             end
167             if(c(in)==0 & a(in)=='1')
168                 printf(var(2+in));
169             end
170         end
171         printf("+" );
172     end
173 end
174 //2 cells
175 z6=[1 1];
176 z7=z6';
177 for i=1:n
178     for j=1:n
179         if(i==n)
180             t1=1;

```

```

182         else
183             t1=i+1;
184         end
185     if(j==n)
186         t2=1;
187     else
188         t2=j+1;
189     end
190     z8=[k(i,j,1) k(i,t2,1)];
191     z9=[k(i,j,2) k(i,t2,2)];
192     no1=number_of(z9,1);
193     if(number_of(z8,0)==0 & no1<cmn2 &
        number_of(z8,1)>0)
194         k(i,j,2)=1;
195         k(i,t2,2)=1;
196         a=strsplit(temp1(1,j));
197         b=strsplit(temp1(1,t2));
198         c=strcmp(a,b);
199         for in=1:max(size(c))
200             if(c(in)==0 & a(in)=='0')
201                 printf(p1(1,i));
202                 printf(' %s ',var(2+in));
203                 printf(" +");
204             end
205             if(c(in)==0 & a(in)=='1')
206                 printf(p1(1,i));
207                 printf(var(2+in));
208                 printf(" +");
209             end
210         end
211     end
212 end
213
214 for i=1:n
215     for j=1:n
216         if(i==n)
217             t1=1;
218         else

```

```

219         t1=i+1;
220     end
221     if(j==n)
222         t2=1;
223     else
224         t2=j+1;
225     end
226     z10=[k(i,j,1);k(t1,j,1)];
227     z11=[k(i,j,2);k(t1,j,2)];
228     no2=number_of(z11,1);
229     if(number_of(z10,0)==0 & no2<cmn2 &
        number_of(z10,1)>0)
230         k(i,j,2)=1;
231         k(t1,j,2)=1;
232         a=strsplit(temp2(i,1));
233         b=strsplit(temp2(t1,1));
234         c=strcmp(a,b);
235         for in=1:max(size(c))
236             if(c(in)==0 & a(in)=='0')
237                 printf(p2(j,1));
238                 printf('%s' ',var(in));
239                 printf("+");
240             end
241             if(c(in)==0 & a(in)=='1')
242                 printf(p2(j,1));
243                 printf(var(in));
244                 printf("+");
245             end
246         end
247     end
248 end
249 end
250 //single cell
251 for i=1:n
252     for j=1:n
253         if(k(i,j,2)==0 & k(i,j,1)==1)
254             a=strsplit(temp1(1,j));
255             b=strsplit(temp2(i,1));

```

```

256         for in=1:max(size(a(:,1)))
257             if(a(in,1)=='1')
258                 printf(var(in+2));
259             else
260                 if(a(in,1)=='0')
261                     printf('%s',var(2+in))
262                         ;
263                 end
264             end
265         end
266         for in=1:max(size(b(:,1)))
267             if(b(in,1)=='1')
268                 printf(var(in));
269             else
270                 if(b(in,1)=='0')
271                     printf('%s',var(in));
272                 end
273             end
274             if(i~=4 & j~=4)
275                 printf("+");
276             end
277         end
278     end
279 end
280 printf("0");
281 disp(" ")
282 endfunction

```

Scilab code AP 8 Kmap for 3 variables with out dontcares

```

1 function []=karmap3(k)
2     n=4;           //three variable kmap
3     m=2
4     k(:, :, 2)=zeros(m,n);
5     var=['z' 'x' 'y'];
6     p1=['z' 'z'];
7     p2=['x' 'y' 'x' 'y'; 'xy'; 'xy'];

```

```

8     cmn4=4;
9     cmn2=3;
10    temp=1;
11    printf('The minimal ecpression of the given Kmap
        ');
12    disp(k(:, :, 1));
13    disp(" is :");
14    printf('f');
15    printf("=");
16    //8 cells
17    for i=1:m
18        for j=1:n
19            if(k(i, j)~=1 & k(i, j)~=2)
20                temp=0;
21                break;
22            end
23        end
24    end
25    if(temp==1)
26        printf("1");
27        abort;
28    end
29    //4 cells
30    z1=ones(1,4);
31    z2=ones(4,1);
32    z3=ones(2,2);
33    temp1=['0' '1'];
34    temp2=['00'; '01'; '11'; '10'];
35    for t=1:m
36        z=k(t, :, 1);
37        no=number_of(k(t, :, 2), 1);
38        if(number_of(z, 0)==0 & no<cmn4 &
           number_of(z, 1)>0)
39            k(t, :, 2)=z1;
40            a=strsplit(temp1(1, t));
41            for in=1:max(size(a))
42                if(a(in)=='0')
43                    printf('%s', var(in));

```

```

44         end
45         if(a(in)=='1')
46             printf(var(in));
47         end
48     end
49     printf("+");
50 end
51 end
52 for i=1:m-1
53     for j=1:n
54         t1=i+1;
55         if(j==n)
56             t2=1;
57         else
58             t2=j+1;
59         end
60         z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
61             ,1)];
62         z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
63             ,2)];
64         no=number_of(z5);
65         if(number_of(z4,0)==0 & no<cmn4 &
66             number_of(z4,1)>0)
67             k(i,j,2)=1;
68             k(i,t2,2)=1;
69             k(t1,j,2)=1;
70             k(t1,t2,2)=1;
71             a=strsplit(temp2(j,1));
72             b=strsplit(temp2(t2,1));
73             c=strcmp(a,b);
74             for in=1:max(size(c))
75                 if(c(in)==0 & a(in)=='0')
76                     printf('%s''',var(1+in));
77                 end
78                 if(c(in)==0 & a(in)=='1')
79                     printf(var(1+in));
80                 end
81             end
82         end
83     end
84 end

```

```

79             printf("+");
80         end
81     end
82 end
83 //2 cells
84 z6=[1 1];
85 z7=z6';
86 for i=1:m
87     for j=1:n
88         t1=i+1;
89         if(j==n)
90             t2=1;
91         else
92             t2=j+1;
93         end
94         z8=[k(i,j,1) k(i,t2,1)];
95         z9=[k(i,j,2) k(i,t2,2)];
96         no1=number_of(z9,1);
97         if(number_of(z8,0)==0 & no1<cmn2 &
98             number_of(z8,1)>0)
99             k(i,j,2)=1;
100            k(i,t2,2)=1;
101            printf(p1(1,i));
102            a=strsplit(temp2(j,1));
103            b=strsplit(temp2(t2,1));
104            c=strcmp(a,b);
105            for in=1:max(size(c))
106                if(c(in)==0 & a(in)=='0')
107                    printf('%s' ',var(1+in));
108                    printf("+");
109                end
110                if(c(in)==0 & a(in)=='1')
111                    printf(var(1+in));
112                    printf("+");
113                end
114            end
115        end
116    end
117 end

```



```

116     end
117     for i=1:m-1
118         for j=1:n
119             t1=i+1;
120             if(j==n)
121                 t2=1;
122             else
123                 t2=j+1;
124             end
125             z10=[k(i,j,1);k(t1,j,1)];
126             z11=[k(i,j,2);k(t1,j,2)];
127             no2=number_of(z11,1);
128             if(number_of(z10,0)==0 & no2<cmn2 &
                number_of(z10,1)>0)
129                 k(i,j,2)=1;
130                 k(t1,j,2)=1;
131                 printf(p2(j,1));
132                 printf("+");
133             end
134         end
135     end
136     //single cell
137     for i=1:m
138         for j=1:n
139             if(k(i,j,2)==0 & k(i,j,1)==1)
140                 printf(p1(1,i));
141                 printf(p2(j,1));
142                 printf("+");
143             end
144         end
145     end
146     printf("0");
147     disp(" ")
148 endfunction

```

Scilab code AP 9 Binary to Decimal convertor

```

1 //bin21dec is a function whcih converts any binary
   number given to it will output its equivalent
   decimal number
2 //pass the binary number as an argument to the
   function
3 // For eg:bin21decimal(1010)
4 //Will give an output of 10
5
6 function [temp]=bin21dec(bin)
7     i=1;w=1;
8
9     temp1=floor(bin);
                                   //separating
   integer part from the given number
10    temp2=modulo(bin,1);
                                   //separating
   decimal part from the given number
11    temp2=temp2*10^3;
                                   //converting
   decimal value to interger for convenience
12
13    while(temp1>0)
   //storing each integer digit in vector for
   convenience
14        p(i)=modulo(temp1,10);
15        temp1=floor(temp1/10);
16        i=i+1;
17    end
18
19    while(temp2>0)
   //storing each integer digit in vector for
   convenience
20        q(w)=modulo(temp2,2);
21        temp2=(temp2/10);
22        temp2=floor(temp2);
23        w=w+1;
24    end
25

```

```

26     temp1=0;
        //clearing the temporary variable 'temp2'
27
28     for i=1:length(p)
        //checking whether it is binary or not.
29         if(p(i)>1) then
30             disp('not a binary number');
31             abort;
32         end
33     end
34
35     for i=1:length(p)
        //multiplying the bits of integer part with
        their position values and adding
36         temp1=temp1+(p(i)*2^(i-1));
37     end
38
39     temp2=0;
        //clearing the temporary variable 'temp2'
40
41     for z=1:w-1
        //multiplying the bits of decimal part with
        their position values and adding
42         temp2=temp2+(q(z)*2^(-1*(4-z)));
43     end
44
45     temp=temp1+temp2;
        //finally adding both the integer and decimal
        parts to get total output.
46 endfunction

```
