Input Encoding Strategy

Replace the symbolic input with a multiple-valued variable as input

10	inp1	01
01	inp1	10
1-	inp2	01
01	inp2	10
11	inp3	01
01	inp3	10

10	100	01
01	100	10
1-	010	01
01	010	10
11	001	01
01	001	10

Minimize function:

10	110	01
11	011	01
01	111	10

Constructing a Cover

If an encoding satisfied ALL constraints a minimum cardinality encoded binary cover can be constructed directly

Just replace each multiple-valued implicant by its supercube

M-V Cover :

10	110	01
11	011	01
01	111	10

Encoding :

inp1	00
inp2	01
inp3	11

Minimized binary cover :

10	0-	01
11	-1	01
01	--	10

supercube (00, 01)
supercube (01, 11)
supercube (00, 01, 11)

Note: Used 10 as a don't-care!

Transpose Coding

Any set of constraints can be satisfied by an encoding constructed as the transpose of the set of constraints

constraints			
10110	transpose encoding :	inp1	100
01101		inp2	011
01110		inp3	111
		inp4	101
		inp5	010

<u>Why?</u> For constraint i supercube has a 1 in position i, whereas codes of symbolic values not in constraint i have a 0

Constraint Compaction

(1) Constraints like 10000 and 01000 are trivially satisfied by any encoding and can be discarded (since all codes have to be different)

(2) Constraints like 111111 can also be discarded

(3) A constraint that is the bitwise intersection of 2 or more constraints is always satisfied by an encoding that satisfies the “parent” constraints.

Compaction Example

constraints

01010
11011
01110
11100

01010 = 11011 & 01110

compacted constraints

11011
01110
11100

transpose

inp1	101
inp2	111
inp3	011
inp4	110
inp5	100

supercube(111, 110) = 11-

Consider 01010

$11- \cap 011 = \phi$

$11- \cap 101 = \phi$

$11- \cap 100 = \phi$

Input Encoding Example

Initial Specification

00	inp1	001
01	inp1	010
10	inp1	001
11	inp1	100
01	inp2	010
01	inp3	010
10	inp3	001
11	inp3	100
00	inp4	001
10	inp4	001
--	inp5	111

Initial Multiple-valued Cover

00	10000	001
01	10000	010
10	10000	001
11	10000	100
01	01000	010
01	00100	010
10	00100	001
11	00100	100
00	00010	001
10	00010	001
--	00001	111

Minimized M-V Cover

-0	10010	000
01	11100	010
10	10110	001
11	10100	100
--	00001	111

Example - 2

Constraints	10010 11100 10110 10100 00001

Non-trivial set of constraints	10010 11100 10110 10100	10100 = 11100 & 10110

Compacted set	10010 11100 10110

Example - 3

Compacted Set

10010
11100
10110

Encoding as Transpose

inp1	111
inp2	010
inp3	011
inp4	101
inp5	000

-0	10010	001
01	11100	010
10	10110	001
11	10100	100
--	00001	111

$\longrightarrow$

-0	1-1	001
01	-1-	010
10	--1	001
11	-11	100
--	000	111

encoded and optimized cover

Output Encoding

0101	out1
0110	out1
0100	out2
0111	out2
1000	out3
1001	out3

Harder problem than input encoding

Unlike in the input encoding case, no trivial encoding (one-hot) gives a minimum number of product terms

One-Hot Output Coding

1100	out1
1111	out1
0001	out2
0000	out2
1110	out2
1101	out2
0110	out3
0101	out3
1101	out3

disjoint minimization →

1100	out1
1111	out1
000 -	out2
1110	out2
1101	out2
0110	out3
- 101	out3

one-hot output encoding

1100	100
1111	100
0001	010
0000	010
1110	010
1101	010
0110	001
0101	001
1101	001

→

1100	100
1111	100
000 -	010
1110	010
1101	010
0110	001
- 101	001

Worst encoding

Dominance Relations

Definition: A binary code dominates another if for every bit position in the second code that has a 1, the first also has a 1

1110	does not dominate	1101
1011	dominates	0000 etc.

Exploiting dominance relations reduces product term count.

One-hot coding, i.e., 1000, 0100 has no dominance relations.

Dominance Don't Cares

If code given to outi dominates code given to outj then cubes that assert outi can be used as don't cares to minimize cubes of outj

1100	out1
1111	out1
1101	out2
1110	out2

out1	01
out2	11

out2 dominates out1

Minimize out1 using out2 cubes as don't cares

1100	out1
1111	out1
1101	-
1110	-

minimize $\rightarrow$

$\boxed{\text{11 - -} \quad \text{out1}}$	
1101	out2
1110	out2

Heuristic Output Encoding

(1) Disjointly minimize each symbolic value

(2) Pick a symbolic value outi
$DC(outi) = \phi$

(3) Pick a symbolic value $outj \neq outi$

(4) $g = P_m[\, outi, DC(outi)\,] - P_m[\, outi, DC(outi) \cup outj\,]$

(5) If $g > 0$ then $DC(outi) = DC(outi) \cup outj$
Add dominance edge $outj \supset outi$
to graph (if there is no cycle)

(6) Goto 4 if there are other symbolic values
$outj \neq outi$ else go to 2

(7) Find encoding satisfying dominance graph

Output Encoding Example

Disjointly Minimized

00011	out1
00000	out1
10100	out1
01100	out1
00010	out2
00001	out2
11001	out2
00100	out3
11100	out3
1110 -	out3
01001	out4
10001	out4

Pick out1
Minimize with out2 as DC

00011	out1
00000	out1
10100	out1
01100	out1
00010	-
00001	-
11001	-

000- -	out1
10100	out1
01100	out1

Add edge out2 $\supset$ out1 to dominance graph

Example - 2

Next, minimize out1
with out2 $\cup$ out3 as DC

00011	out1
00000	out1
10100	out1
01100	out1
00010	-
00001	-
11001	-
00100	-
1110 -	-

$\rightarrow$

000- -	out1
- -100	out1

Add edge out3 $\supset$ out1

Next, minimize out1
with out2 $\cup$ out3 $\cup$ out4 as DC
No reduction

Pick out2
Can't use out1 as a DC (cycle)
Minimize with out3 as DC

00010	out2
00001	out2
11001	out2
00100	-
1110 -	-

no reduction

Example - 3

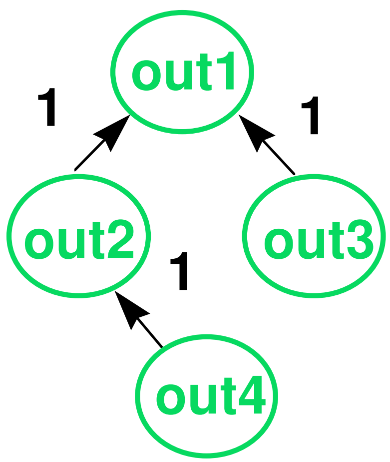
Minimize out2 with out4 as DC

00010	out2
00001	out2
11001	out2
01001	-
10001	-

→

00010	out2
- - 001	out2

Add edge out4 $\supset$ out2



out1	001
out2, out3	011, 101
out4	111

encoded and minimized cover

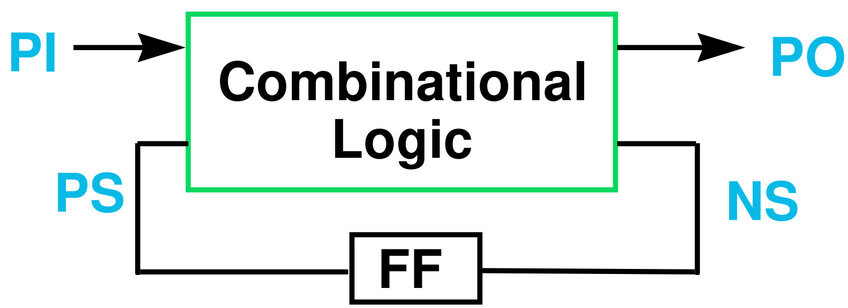
11 - 3 = 8

- - 100	001
000 - -	001
- - 001	011
00010	010
00100	100
10001	100
01001	100
1110 -	101

State Assignment

Encoding

S0	01	S1	10
S2	00	S3	11



State Transition Table

PI	PS	NS	PO
000	S0	S0	01
100	S0	S1	01
010	S0	S3	01
- -1	S0	S0	10
100	S1	S1	01
0-0	S1	S0	01
110	S1	S2	01
- -1	S1	S0	10
110	S2	S2	01
100	S2	S1	01
- - -	S3	S3	- -

State Assignment Strategy

Integrate input and output encoding techniques

Obtain input constraints via M-V minimization

Look at the output encoding problem for M-V input, symbolic output function

Symbolic output

0	10000	S4	01
1	10000	S2	10
0	01000	S3	10
1	01000	S3	01
-	00100	S1	10
0	00010	S5	10
1	00010	S5	01
-	00001	S1	01

Obtain dominance relations

A Complete Example

0	S1	S2	1
1	S1	S3	1
1	S2	S5	1
0	S3	S3	1
1	S3	S2	1
0	S4	S3	1
1	S4	S4	1
0	S5	S4	1
1	S5	S4	1

State Transition Table

Multiple-valued cover

0	10000	01000	1
1	10000	01000	1
0	01000	00100	1
1	01000	00001	1
0	00100	00100	1
1	00010	01000	1
0	00010	00100	1
1	00001	00010	1
0	00001	00010	1
1	00001	00010	1

→

Minimized M-V Cover

-	10000	01000	1
1	10100	01000	1
0	01110	00100	1
1	00011	00010	1
-	00001	00010	1
1	01000	00001	1

Example - 2

1	10100	S2	1
-	10000	S2	1
0	01110	S3	1
1	00011	S4	1
-	00001	S4	1
1	01000	S5	1

Output encode

Minimize S2 with S3 as DC

1	10100	S2	1
-	10000	S2	1
0	01110	-	1

$\rightarrow$

-	10100	S2	1

Add edge S3 $\supset$ S2

(No further reduction for S2, no reduction for S3)

Minimize S4 with S3 as DC

1	00011	S4	1
-	00001	S4	1
0	01110	-	1

$\rightarrow$

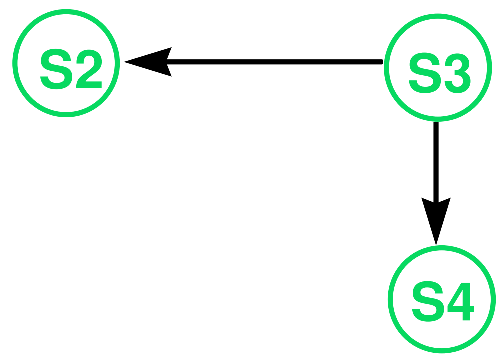
-	00011	S4	1

Add edge S3 $\supset$ S4

Example - 3

Input constraints

10100
01110
00011



dominance graph

Encoding as transpose

Violating bit

S1	100
S2	010
S3	110
S4	011
S5	001

S3 $\supset$ S2
S3 $\not\supset$ S4

Complement bit

S1	10 1
S2	01 1
S3	11 1
S4	01 0
S5	00 0

satisfies both input and dominance constraints

Example - 4

0	S1	S2	1
1	S1	S2	1
0	S2	S3	1
1	S2	S5	1
0	S3	S3	1
1	S3	S2	1
0	S4	S3	1
1	S4	S4	1
0	S5	S4	1
1	S5	S4	1

Original STT

Substitute Encoding →

0	101	011	1
1	101	011	1
0	011	111	1
1	011	001	1
0	111	111	1
1	111	011	1
0	010	111	1
1	010	010	1
0	000	010	1
1	000	010	1

Add don’t care codes and Minimize

-	--0	010	0
1	-1-	111	0
-	---	000	1
-	1--	011	0

4 product terms versus 6 in one-hot