Finite State Machine Minimization

Advanced

Methods based on triangular table and binate covering

Example 1. Minimize the following Mealy Finite State Machine

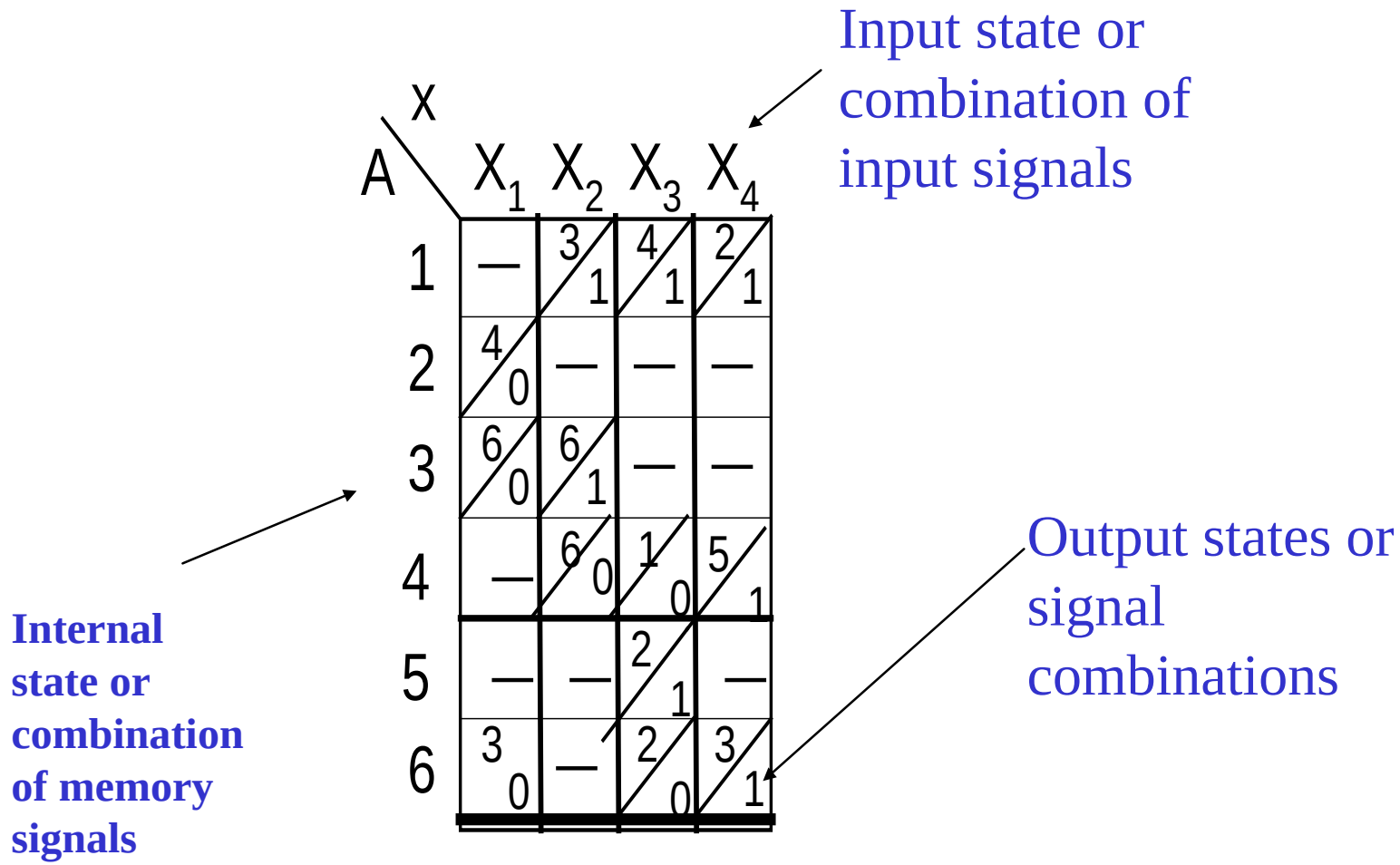
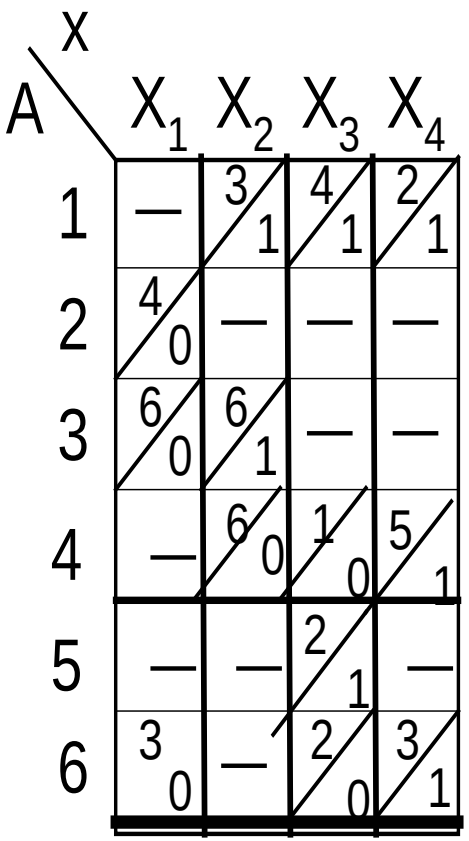
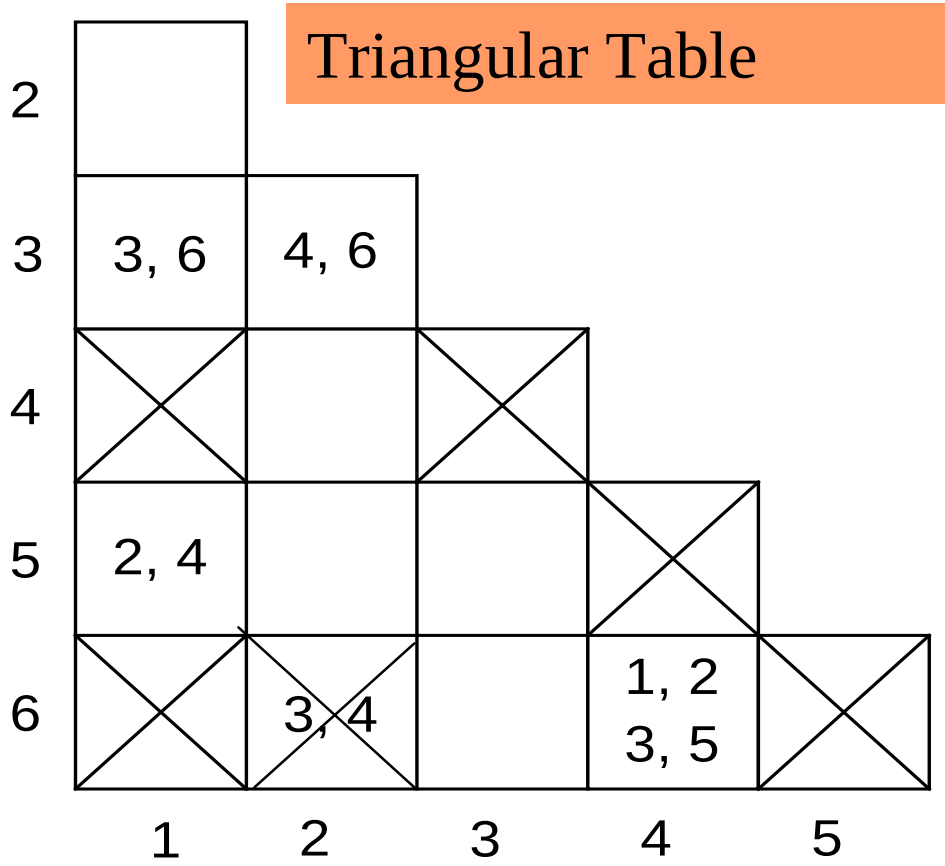


Fig 5.17bcd

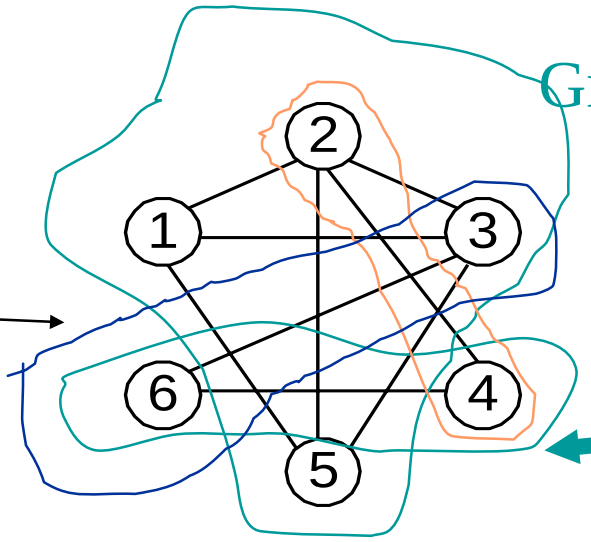


b)

Triangular Table



c)

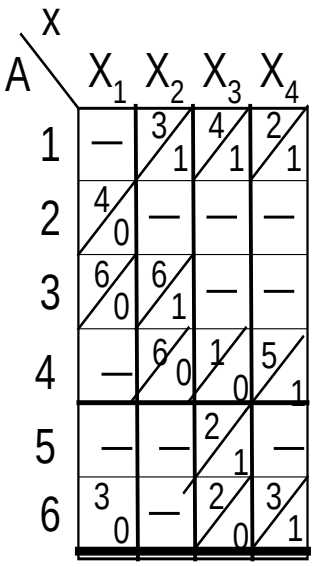


Group {1,2,3,5}

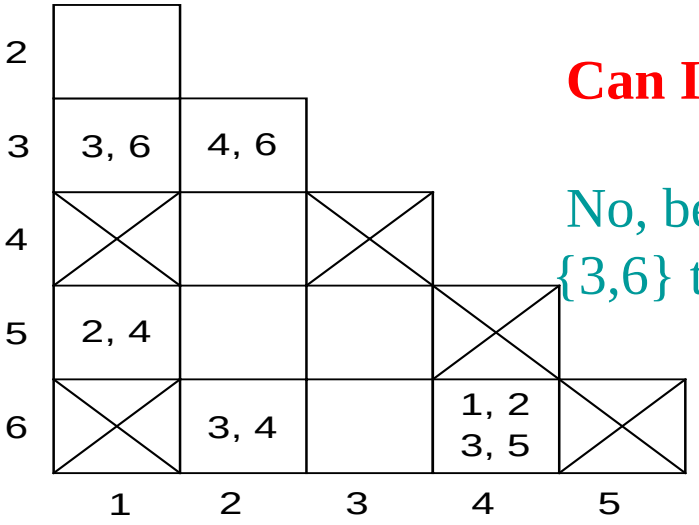
4 Maximum cliques

Compatibility Graph obtained from the Triangular Table

Fig 3.16



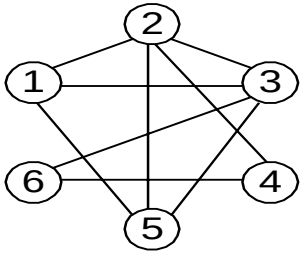
b)



Can I take {1,2,3,5} and {4,6}??

No, because {1,2,3,5} implies states {3,6} to be in one group.

c)



Solution is to split {1,2,3,5} to {1,2} and {3,5}

I can take all max cliques but solution will be not minimal

{1,2} implies nothing, {3,5} implies nothing, {4,6} implies {1,2} and {3,5}

Solution : {1,2}, {3,5}, {4,6}

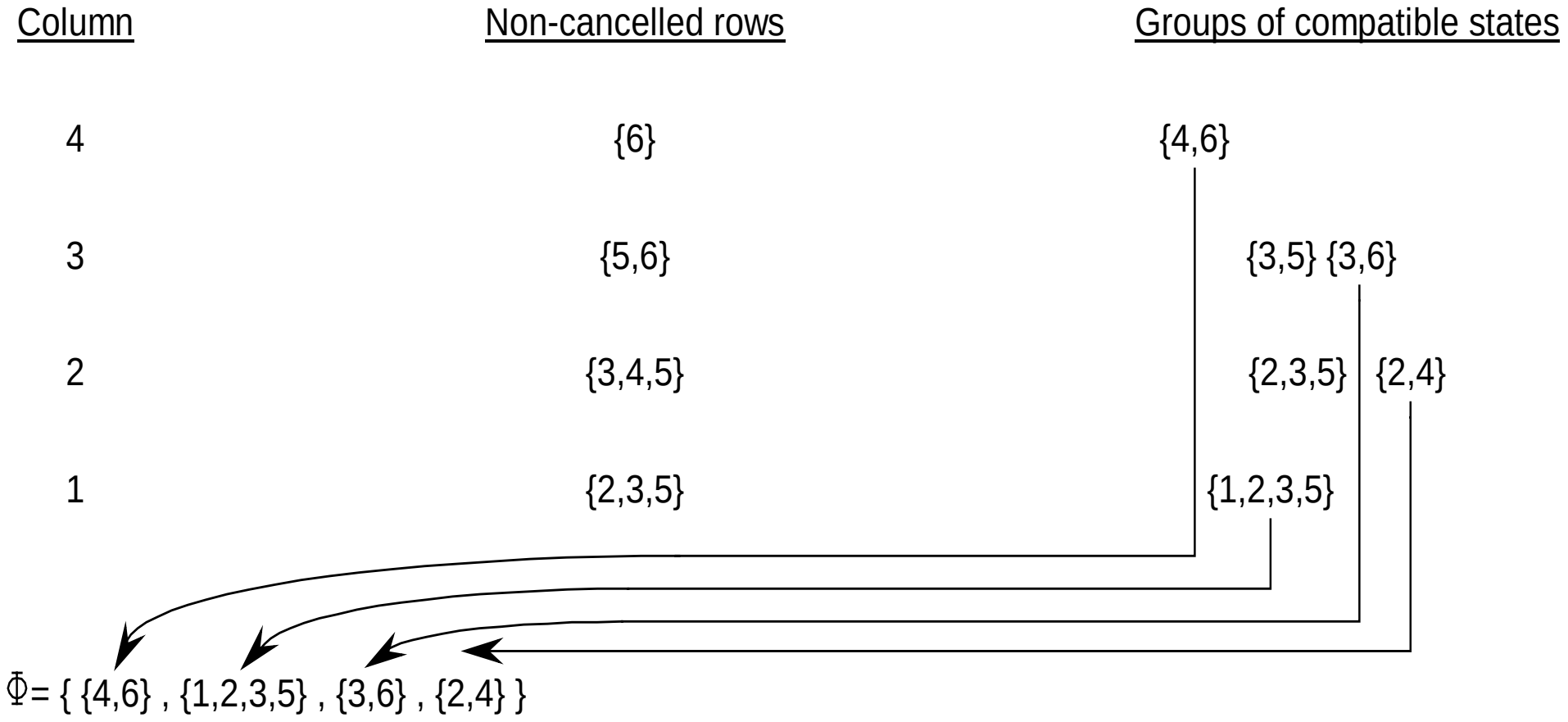
There are other solutions

But how I know to split this way? Heuristics!

In any case creating Maximum Compatible Groups is useful!

Systematic Method of Creating Maximum Compatible Groups

a)



This method is systematic and creates all maximum compatible groups (cliques)

For small FSMs you can find them by visual inspection

Fig5.17a

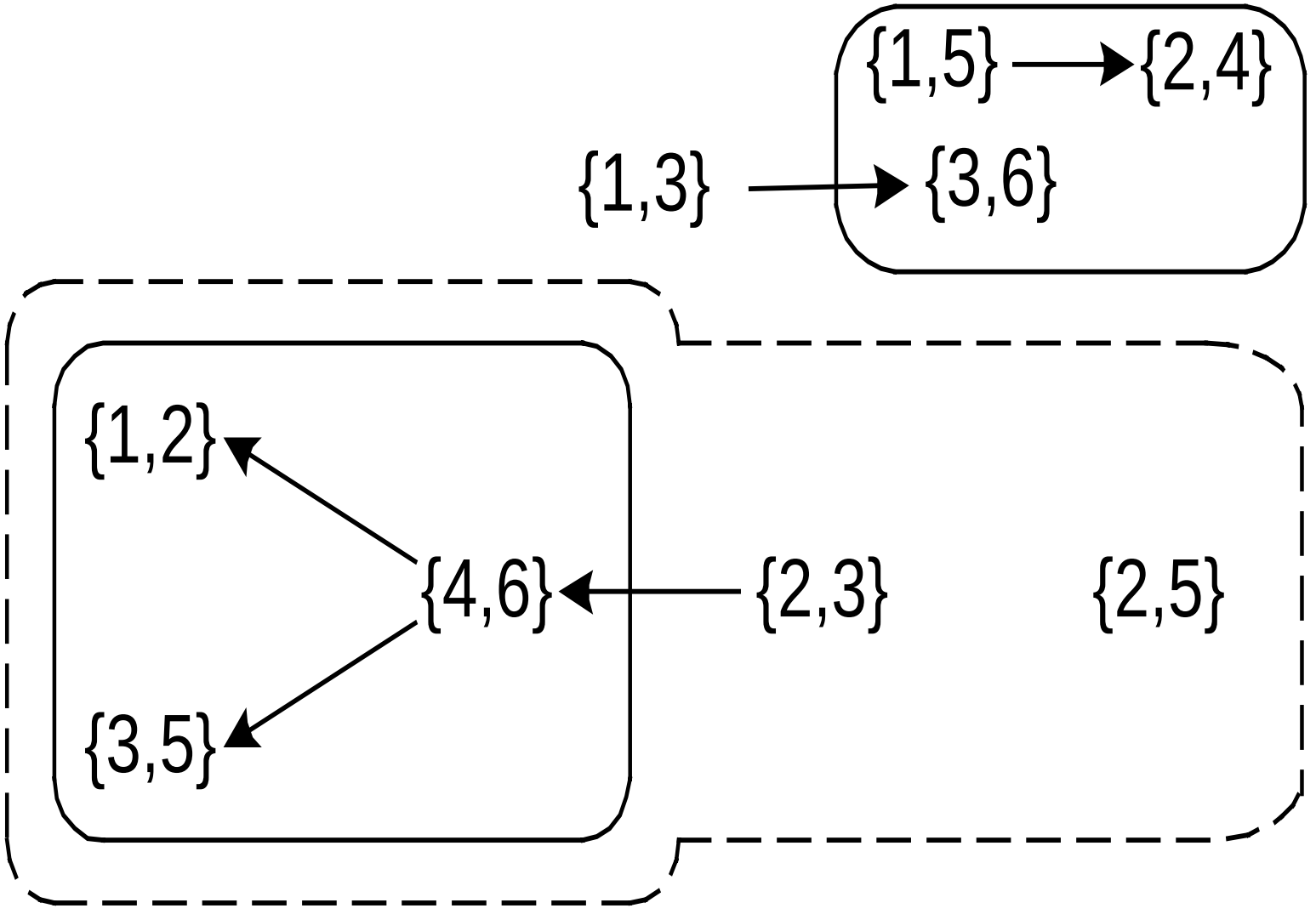
Complete and Closed Subgraph

- **Complete** = all state numbers have been used at least one inside it
- **Closed** = there is no arrow going out of this graph

Closure graph for compatibility pairs

This method selects subsets of maximum cliques in order to satisfy the completeness and closure conditions for state numbers

This way we found other solutions.
Please draw machines



Combining groups of compatibles from the cover to single state

This is a final stage of state table minimization

It can be done with:

1) ALL groups of compatible states or

2) with the set of closed and closed groups of compatible states

Now let us go back to fast method, remember that it is not optimal

Combining *ALL* groups of compatibles from the cover to single state

b)

A \ x	X_1	X_2	X_3	X_4
1	—	3 / 1	4 / 1	2 / 1
2	4 / 0	—	—	—
3	6 / 0	6 / 1	—	—
5	—	—	2 / 1	—
3	6 / 0	6 / 1	—	—
6	3 / 0	—	2 / 0	3 / 1
2	4 / 0	—	—	—
4	—	6 / 0	1 / 0	5 / 1
4	—	6 / 0	1 / 0	5 / 1
6	3 / 0	—	2 / 0	3 / 1

A = {1,2,3,5}

B = {3,6}

C = {2,4}

D = {4,6}

c)

A \ x	X_1	X_2	X_3	X_4
A	{4,6} / 0	{3,6} / 1	{2,4} / 1	2 / 1
B	{3,6} / 0	6 / 1	2 / 0	3 / 1
C	4 / 0	6 / 0	1 / 0	5 / 1
D	3 / 0	6 / 0	{1,2} / 0	{3,5} / 1

d)

A \ x	X_1	X_2	X_3	X_4
A	D / 0	B / 1	C / 1	A,C / 1
B	B / 0	B,D / 1	A,C / 0	A,B / 1
C	C,D / 0	B,D / 0	A / 0	A / 1
D	A,B / 0	B,D / 0	A / 0	A / 1

This is non-deterministic FSM, you can make a choice to simplify state assignment

Combining *ALL* groups of compatibles from the cover to single state

Select states from groups to create large groups of the same state

X	X_1	X_2	X_3	X_4
A	D / 0	B / 1	C / 1	A,C / 1
B	B / 0	B,D / 1	A,C / 0	A,B / 1
C	C,D / 0	B,D / 0	A / 0	A / 1
D	A,B / 0	B,D / 0	A / 0	A / 1

Select B in whole column

Select A in whole column

Select C

X	X_1	X_2	X_3	X_4
A	D / 0	B / 1	C / 1	A,C / 1
B	B / 0	B,D / 1	A,C / 0	A,B / 1
C	C,D / 0	B,D / 0	A / 0	A / 1
D	A,B / 0	B,D / 0	A / 0	A / 1

As you see, it is a good idea to combine FSM minimization and state assignment. Many methods are based on this idea.

Creating new table by combining states from groups of compatible states

- The same method of combining states can be applied to any set of compatible and closed

Problem for possible homework: Find an FSM table for which the following triangular table exists:

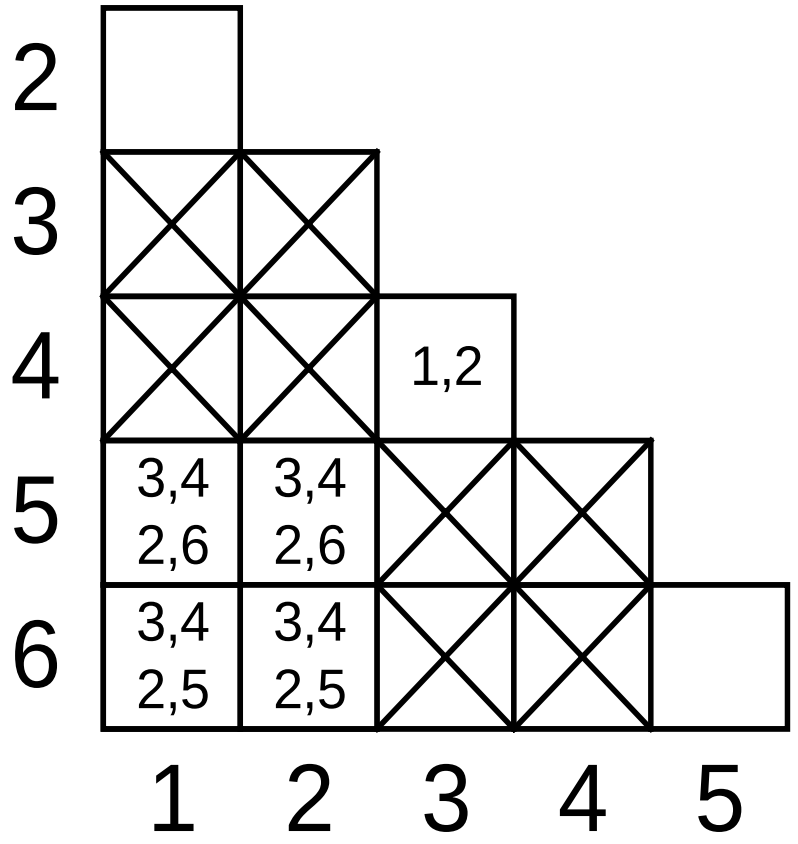


Fig.5.18

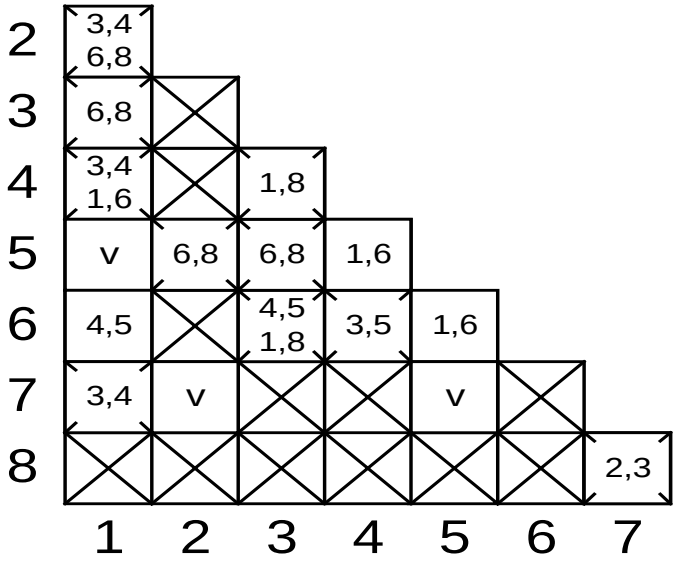
Example 3 of FSM Minimization

1	4/-	6/0
2	3/1	8/0
3	4/0	8/0
4	3/0	1/0
5	-/-	6/0
6	5/0	1/0
7	3/1	-/-
8	2/1	1/1

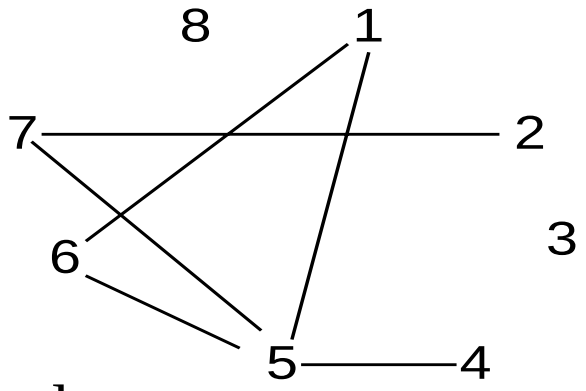
{1,6}
{2,7}
{3}
{4,5}
{8}

4/0	1/0
3/1	8/0
4/0	8/0
3/0	1/0
2/1	1/1

b)



c)



d)

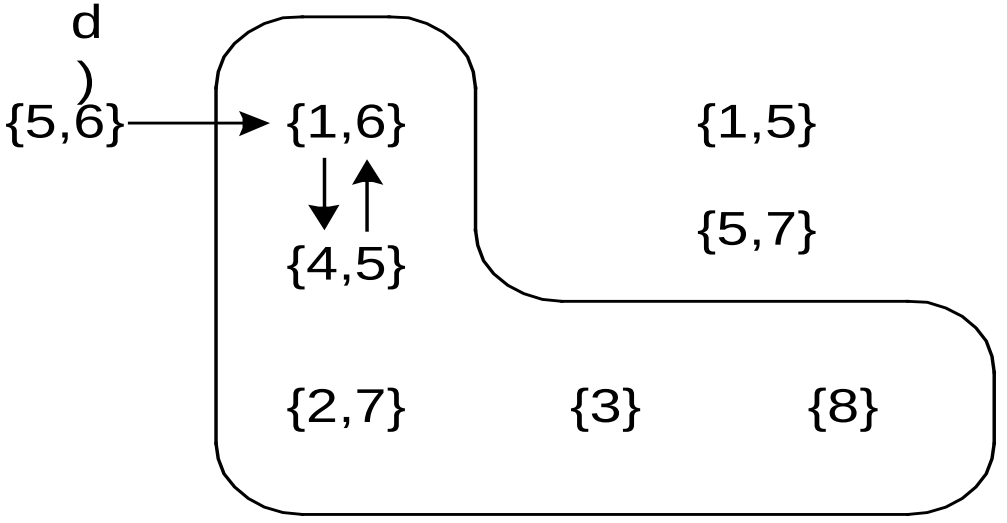


Fig.5.21.bcd