Digital Logic Design

Hajar Falahati

Department of Computer Engineering
IRAN University of Science and Technology

hfalahati@iust.ac.ir

Outline

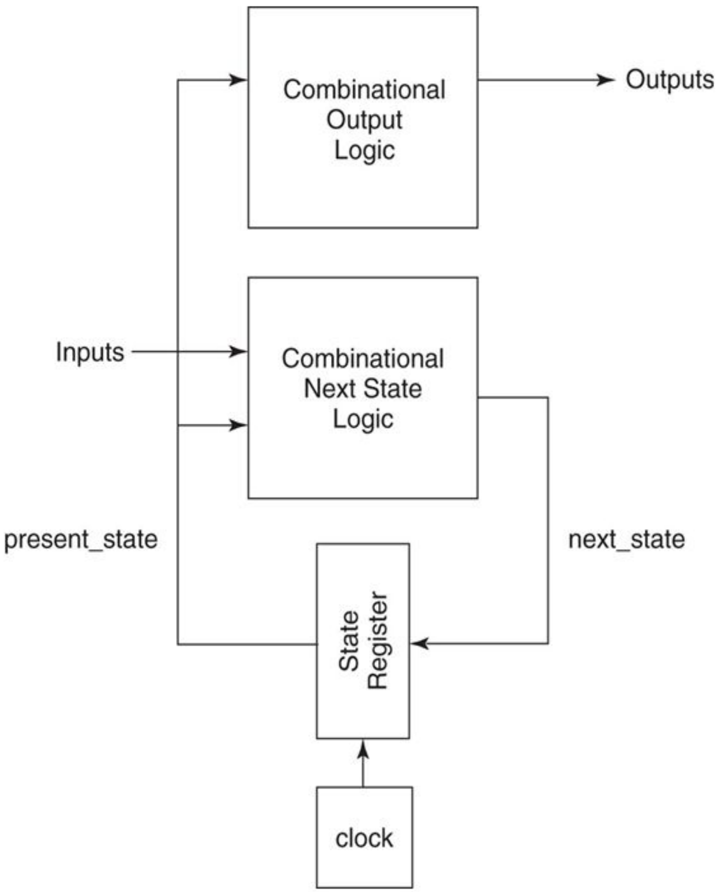
- Simplification of Synchronous Sequential Circuits

Simplification

Simplification

- Design a **simplified** sequential circuit
  - Reduce the number of states
  - **Without changing the functionality**

- Advantages of reducing the states
  - Less memory elements
  - => decreases cost
  - => decreases complexity
  - => Aids failure analysis



How to Simplify?

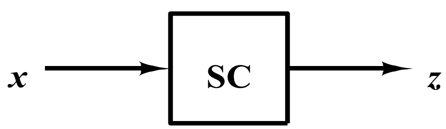
- Find equivalent states
  - E.g., S0, S1, S2
- Keep one of these equivalent states
  - E.g., S0
- Remove other equivalent states in each group ones
  - E.g., S1, S2

Equivalent States

- S_1, S_2, …, S_j are **equivalent** *If and ony if*
  - For **every possible input sequence**
  - **Same output sequence** is produced
    - Same **output**
    - Same **next state**

- Let S_i and S_j be states of a completely specified sequential circuit.
- Let S_k and S_l be the next states of S_i and S_j, respectively for input I_p.
- S_i and S_j are **equivalent** if and only if for every possible I_p the following conditions are satisfied.
  - The outputs produced by S_i and S_j are the same,
  - The next states S_k and S_l are equivalent.

Equivalent States: sample



(a)

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

(b)

Initial State	Input Sequences 00	01	11	10
A	11	10	01	00
B	11	10	00	01
C	11	10	00	01
D	00	01	11	10
E	00	01	11	10

(c)

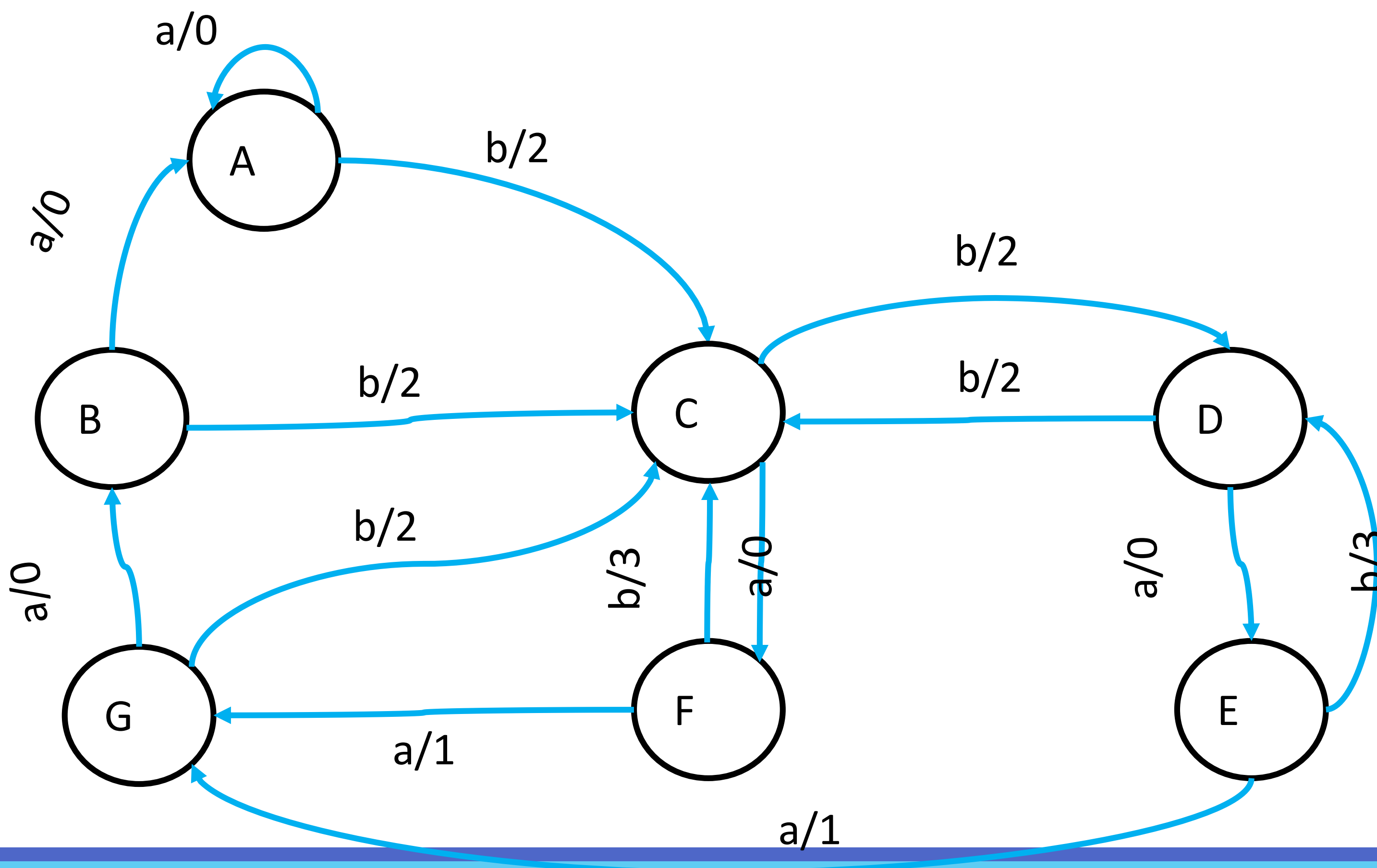
Initial State	Input Sequences 000	001	010	011	100	101	110	111
A	111	110	100	101	011	010	000	001
B	111	110	100	101	000	001	011	010
C	111	110	100	101	000	001	011	010
D	000	001	011	010	111	110	100	101
E	000	001	011	010	111	110	101	100

(d)

Conditional Equivalent States

- S_1, S_2, …, S_j are **conditional equivalent** *If and ony if*
  - For **every possible input sequence**
    - Same **output**
    - Different **next state**
  - **Next states** should be **equivalent**

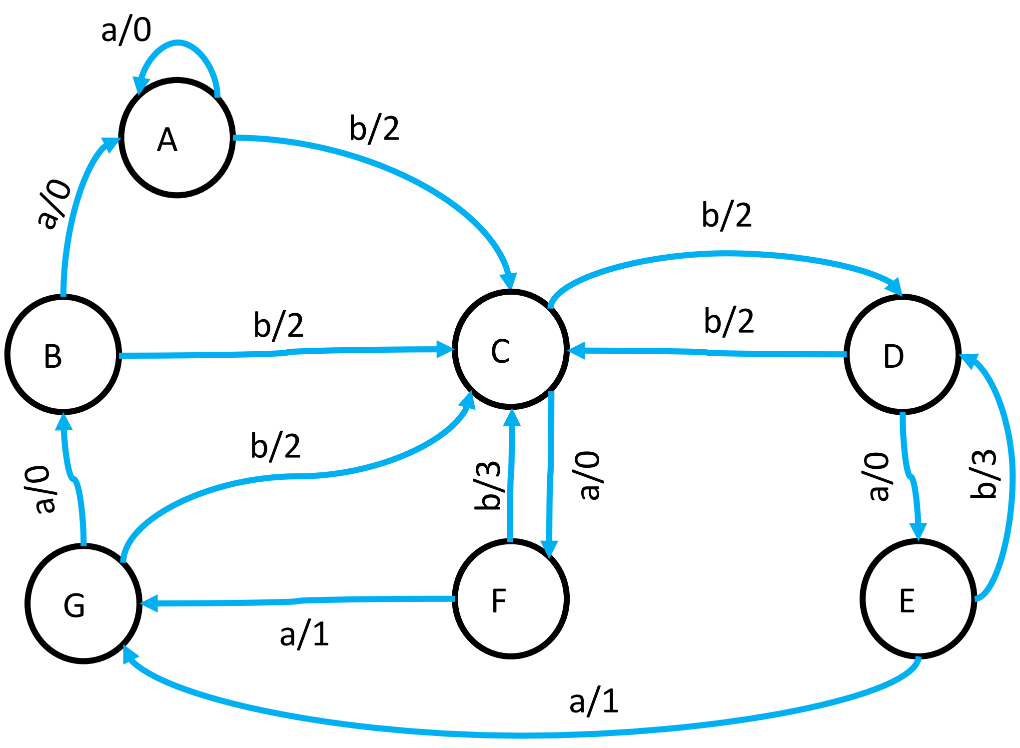
Simplification Sample 1



Sample 1: Equivalent States

- Find equivalent states
  - A, B

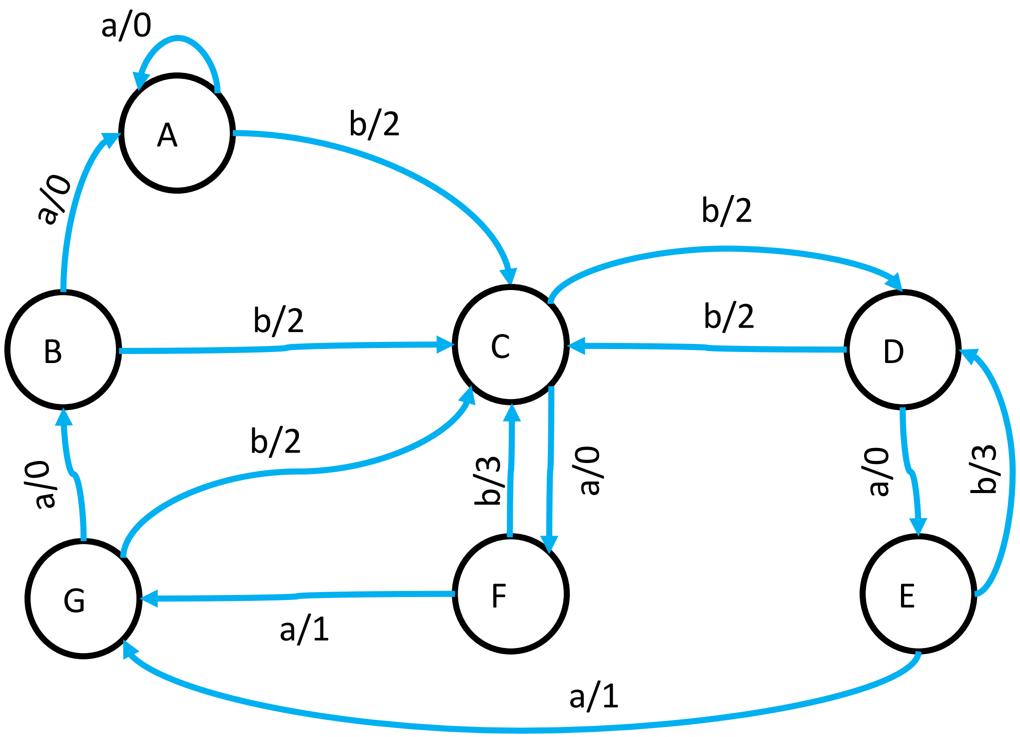
Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2



Sample 1: Equivalent States (cont'd)

- Find equivalent states
  - A, B

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

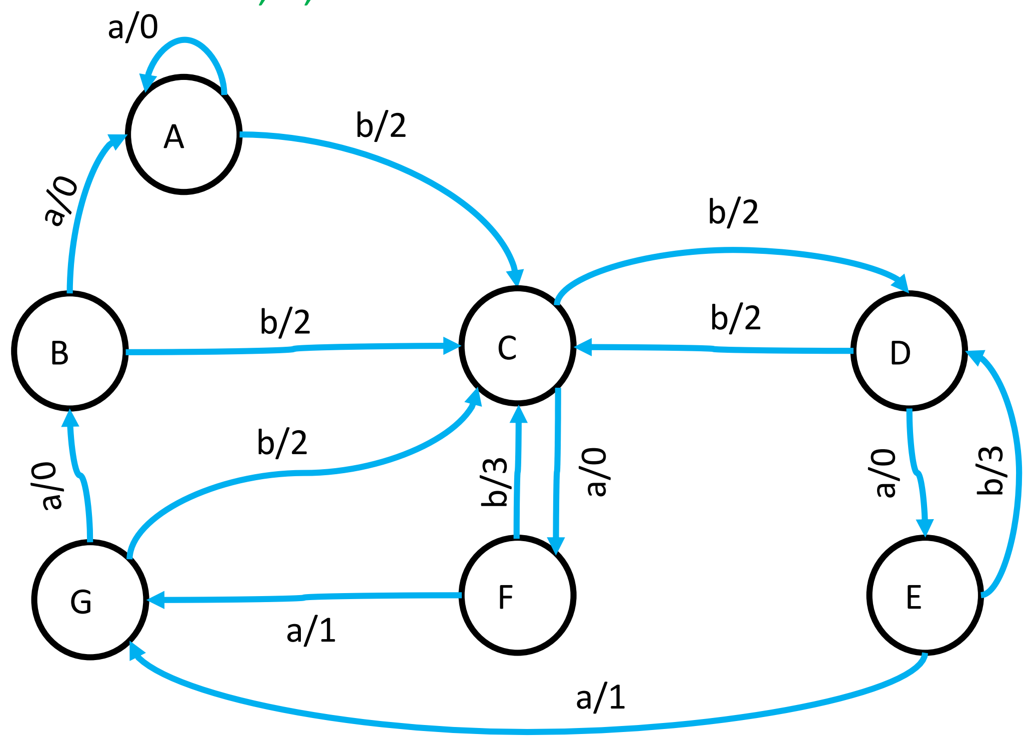


Sample 1: Equivalent States (cont'd)

- Find equivalent states
  - A, B

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

- Find conditional equivalent states
  - C, D
  - E, F
  - A, B, G

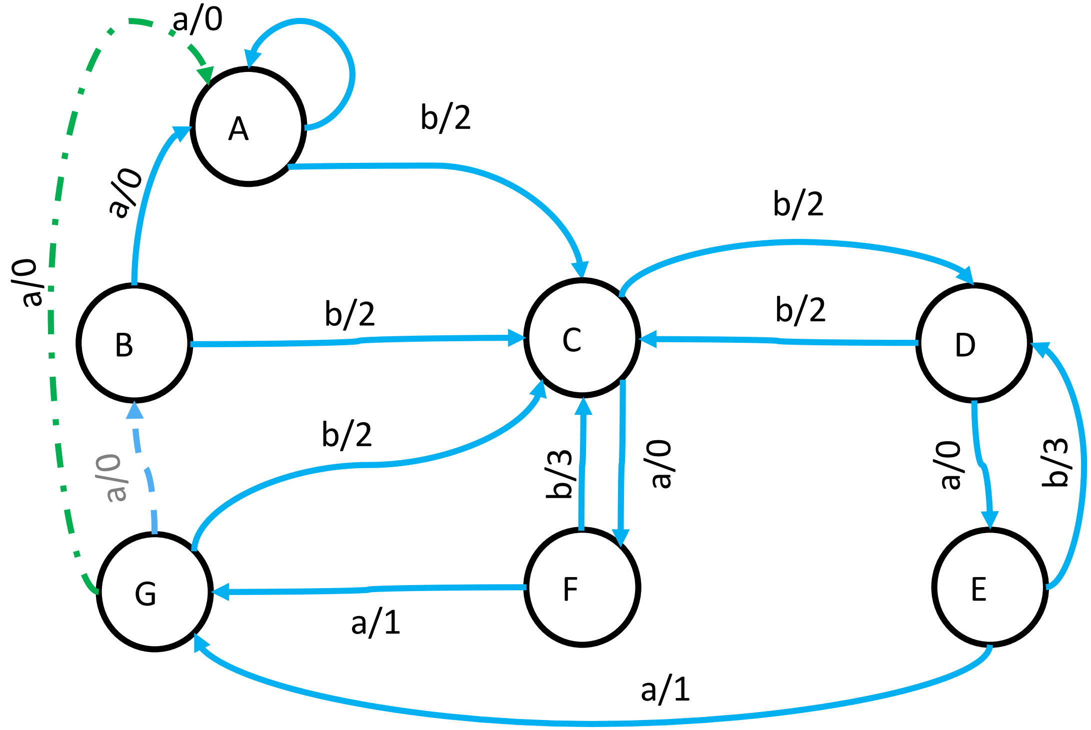


Sample 1: Partitioning

- Find equivalent states
  - A, B

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

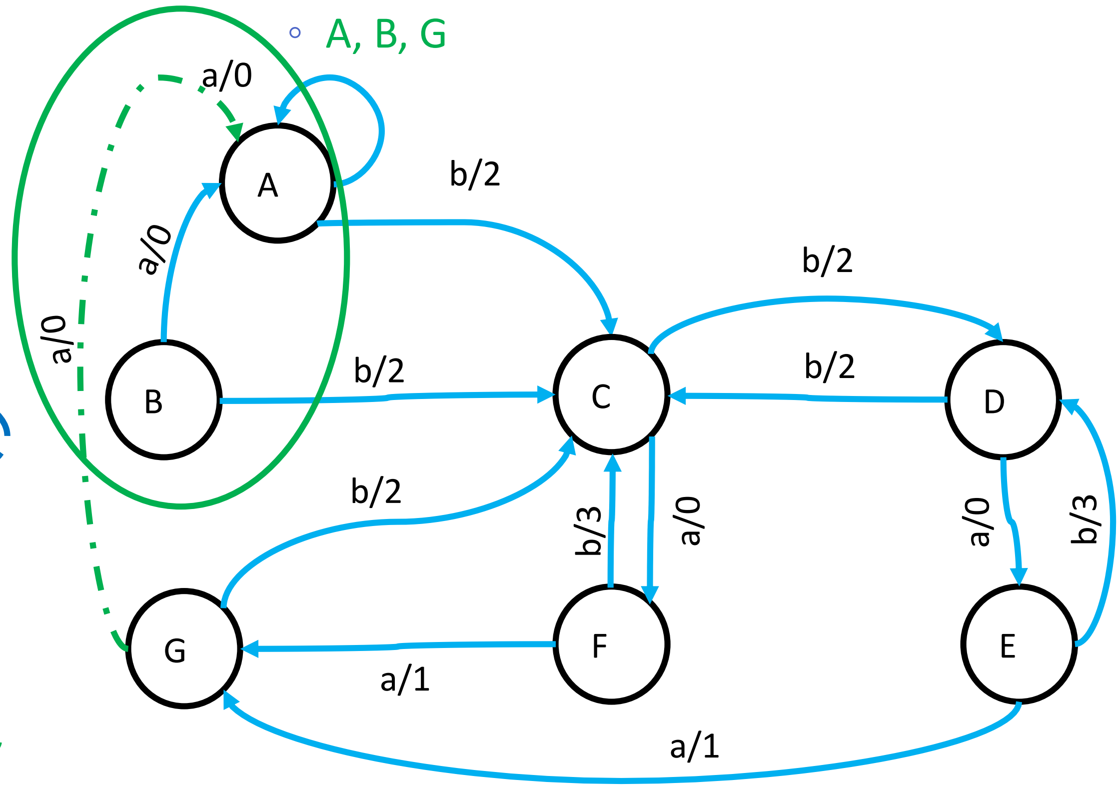
- Find conditional equivalent states
  - C, D
  - E, F
  - A, B, G



Sample 1: Partitioning

- Find equivalent states
  - A, B
- Find conditional equivalent states
  - C, D
  - E, F
  - A, B, G

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

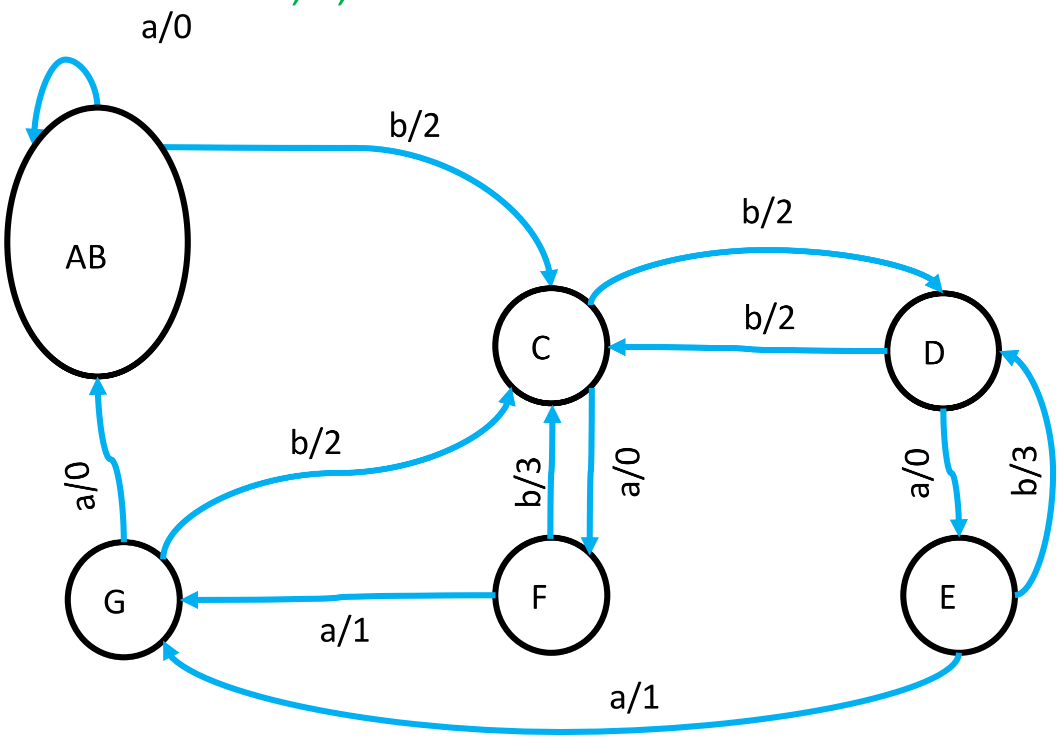


Sample 1: Partitioning

- Find equivalent states
  - A, B

- Find conditional equivalent states
  - C, D
  - E, F
  - A, B, G

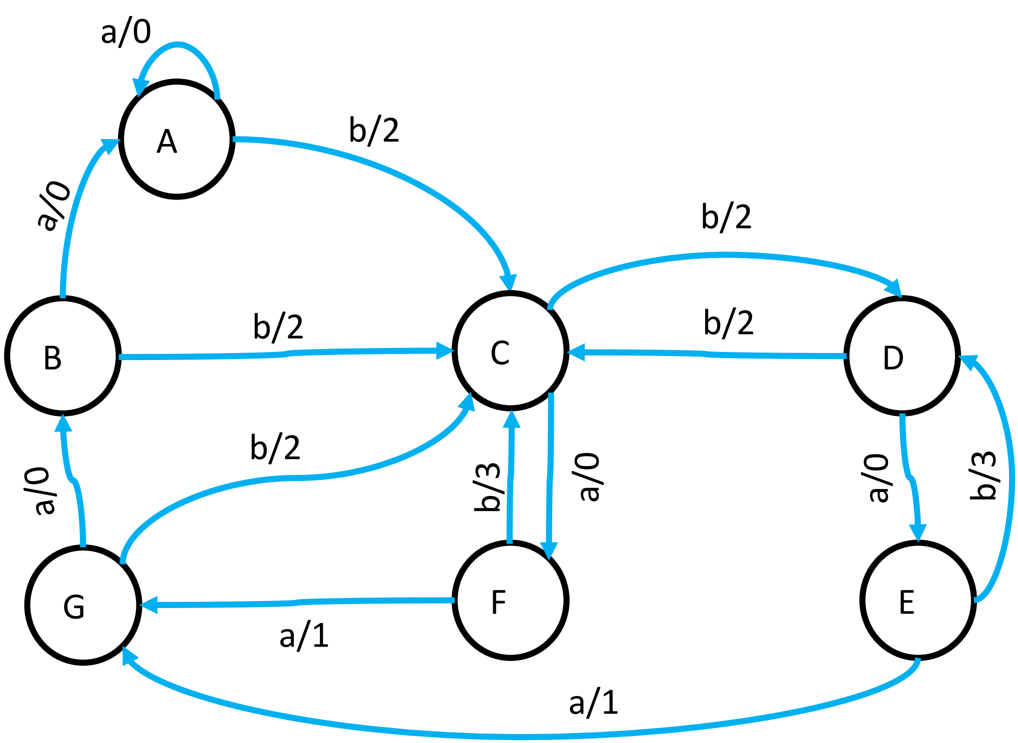
Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2



Implication Table

- A procedure for finding all the equivalent states

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2



How to Find Equivalent States?

- Inspection
- Partitioning
- Implication Tables

Inspection

Inspection Technique

- The simplest and most obvious technique
- A.k.a., row matching
- Steps
  1. Recognize multiple rows in the state tables that perform the same function
  2. Remove the redundant states

Inspection Sample

	x = 0	1
A	*B*/0	*C*/1
B	*C*/0	*A*/1
C	*D*/1	*B*/0
D	*C*/0	*A*/1

B and *D*: **Equivalent Sates** ⇒

	x = 0	1
A	*B*/0	*C*/1
B	*C*/0	*A*/1
C	*B*/1	*B*/0

Partitioning

Partitioning Technique

- The successive determination of partitions P_k
  - k=1,2,3,…
- Each P_k involves number of blocks
  - Each block consists of one or more states
  - The states contained in a block are k-equivalent
- Suppose you have states of s_1,s_2,s_3,s_4,s_5
  - $P_k = (s_1, s_3)(s_2, s_2)(s_5)$
  - P_k consists of 3 blocks such that
    - s_1 and s_3 are k-equivalent
    - s_2 and s_2 are k-equivalent
  - s_5 is not k-equivalent with any other state

Partitioning Procedure: Step 1

- Forms the **first partition P_1**
  - Placing two or more states in the same block if their **output is identical for each input**

	$x = 0$	$x = 1$
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

$P_1 = (A,B,C)\ (D,E)$

Partitioning Procedure: Step 2

- Derives Successive partitions P_k, k=1,2,3,…
  - Placing two or more states in the same block of P_k, if for **each input value** (e.g., x=0 and x=1), their next states **all lie** in a single block of p_{k-1}

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

P_1 = (A,B,C) (D,E)

- Let's check the next states for A,B, and C
  - X=0
    - A -> C , B -> C , C -> B
    - C ,C, and B **all lie** in first block P_1
  - X=1
    - A -> B , B -> E , C -> E
    - B , E, and E lie in different blocks of P_1

The **block** of (A,B,C) is spited into (A) (B,C)

Partitioning Procedure: Step 2 (cont'd)

- Derives Successive partitions P_k, k=1,2,3,…
  - Placing two or more states in the same block of P_k, if for **each input value** (e.g., x=0 and x=1), their next states **all lie** in a single block of p_{k-1}

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

P_1 = (A,B,C) (D,E)

- Let's check the next states for D and E
  - X=0
    - D -> D and E -> E
    - D and E **all lie** in first block P_1
  - X=1
    - D -> B and E -> A
    - B and A **all lie** in first block P_1

The **block** of (D,E) remains **unchanged**

Partitioning Procedure: Step 2 (cont'd)

- Derives Successive partitions P_k, k=1,2,3,…
  - Placing two or more states in the same block of P_k, if for **each input value** (e.g., x=0 and x=1), their next states **all lie** in a single block of p_{k-1}

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

P_1 = (A,B,C) (D,E) → P_2 = (A) (B,C) (D,E)

Partitioning Procedure: Step 3

- **Stop the procedure** when $P_k = P_{k+1}$
  - I.e., the states in each block of P_k are k-equivalent, also (k+1)-equivalent and also (k+2)-equivalent and so on

Partitioning Procedure: Step 3 (cont'd)

- **Stop the procedure** when $P_k = P_{k+1}$
  - I.e., the states in each block of P_k are k-equivalent, also (k+1)-equivalent and also (k+2)-equivalent and so on

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

P_1 = (A,B,C) (D,E) → P_2 = (A) (B,C) (D,E)

- Let's check the next states for B and C
  - X=0
    - B -> C , C -> B
    - C and B **all lie** in the same block of P_2
  - X=1
    - B -> E , C -> E
    - E and E **all lie** in the same block of P_2

The **block** of (B,C) remains **unchanged**

Partitioning Procedure: Step 3 (cont’d)

Iran University of Science & Technology IUST

- **Stop the procedure** when $P_k = P_{k+1}$
  - I.e., the states in each block of P_k are k-equivalent, also (k+1)-equivalent and also (k+2)-equivalent and so on

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

P_1 = (A,B,C) (D,E) → P_2 = (A) (B,C) (D,E)

- Let’s check the next states for D and E
  - X=0
    - D -> E , E -> E
    - E and E **all lie** in the same block of P_2
  - X=1
    - D -> B , E -> A
    - B and A lie in different blocks of P_1

The **block** of (D,E) is spited into (D) (E)

Partitioning Procedure: Step 3 (cont'd)

- **Stop the procedure** when $P_k = P_{k+1}$
  - I.e., the states in each block of P_k are k-equivalent, also (k+1)-equivalent and also (k+2)-equivalent and so on

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

P_1 = (A,B,C) (D,E)

P_2 = (A) (B,C) (D,E)

P_3 = (A) (B,C) (D) (E)

Partitioning Procedure: Step 3 (cont’d)

- **Stop the procedure** when $P_k = P_{k+1}$
  - I.e., the states in each block of P_k are k-equivalent, also (k+1)-equivalent and also (k+2)-equivalent and so on

P_1 = (A,B,C) (D,E) → P_2 = (A) (B,C) (D,E) → P_3 = (A) (B,C) (D) (E)

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

	Partition blocks	Action
Partition P_0	(ABCDE)	
Output for x = 0	11100	Separate (ABC) and (DE)
Output for x = 1	00011	Separate (ABC) and (DE)
Partition P_1	(ABC) | (DE)	
Next state for x = 0	CCB | DE	
Next state for x = 1	BEE | BA	Separate (A) and (BC)
Partition P_2	(A) | (BC) | (DE)	
Next state for x = 0	C | CB | DE	
Next state for x = 1	B | EE | BA	Separate (D) and (E)
Partition P_3	(A) | (BC) | (D) | (E)	
Next state for x = 0	C | CB | D | E	
Next state for x = 1	B | EE | B | A	
Partition $P_4 = P_3$	(A) (BC) (D) (E)	

States B and C are equivalent

Partitioning Procedure: Step 3 (cont'd)

- **Stop the procedure** when $P_k = P_{k+1}$
  - I.e., the states in each block of P_k are k-equivalent, also (k+1)-equivalent and also (k+2)-equivalent and so on

P_1 = (A,B,C) (D,E) → P_2 = (A) (B,C) (D,E) → P_3 = (A) (B,C) (D) (E)

→ P_4 = (A) (B,C) (D) (E)

	x = 0	x = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

P3 equals to P4

The procedure is terminated

States B and C are equivalent

You can remove either B or C

Sample 2

Using the partitioning, reduce the following state table

	x = 0	x = 1
A	E/0	D/0
B	A/1	F/0
C	C/0	A/1
D	B/0	A/0
E	D/1	C/0
F	C/0	D/1
G	H/1	G/1
H	C/1	B/1

P1 = (A, D) (B, E) (C, F) (G, H) — Checking the outputs

P2 = (A, D) (B, E) (C, F) (G) (H) — Checking the next **states** in the same **block** of P_1

P3 = (A, D) (B, E) (C, F) (G) (H) — Checking the next **states** in the same **block** of P_2

P2 = P3

States of the pairs of (A,D) and (B,E) and (C,F) are equivalent

Sample 2 (cont'd)

	x = 0	x = 1
A	E/0	D/0
B	A/1	F/0
C	C/0	A/1
D	B/0	A/0
E	D/1	C/0
F	C/0	D/1
G	H/1	G/1
H	C/1	B/1

(A = D)
(B = E)
(C = F)

A' = (A,D)
B' = (BE)
C' = (CF)
D' = (G)
E' = (H)

	x = 0	x = 1
A'	B'/0	A'/0
B'	A'/1	C'/0
C'	C'/0	A'/1
D'	E'/1	D'/1
E'	C'/1	B'/1

Sample 3

Using the partitioning, reduce the following state table

	x = 0	x = 1
A	A/0	B/0
B	H/1	C/0
C	E/0	B/0
D	C/1	D/0
E	C/1	E/0
F	F/1	G/1
G	B/0	F/0
H	H/1	C/0

P1 = (A, C, G) (B, D, E, H) (F)

Checking the outputs

P2 = (A) (C, G) (B,H) (D, E) (F)

From column x = 0

= (A) (C) (G) (B,H) (D, E) (F)

From column x = 1

P2 = (A) (C) (G) (B,H) (D, E) (F)

P3 = (A) (C) (G) (B,H) (D, E) (F)

P2 = P3

Sample 3 (cont'd)

	x = 0	x = 1
A	A/0	B/0
B	H/1	C/0
C	E/0	B/0
D	C/1	D/0
E	C/1	E/0
F	F/1	G/1
G	B/0	F/0
H	H/1	C/0

A' = (A)
C' = (C)
D' = (G)
E' = (B,H)
F' = (D, E)
B' = (F)

	x = 0	x = 1
A'	A'/0	E'/0
B'	B'/1	D'/1
C'	F'/0	E'/0
D'	E'/0	B'/0
E'	E'/1	C'/0
F'	C'/1	F'/0

Sample 4

Using the partitioning, reduce the following state table with two inputs

	x_1x_2 = 00	01	11	10
A	D/0	D/0	F/0	A/0
B	C/1	D/0	E/1	F/0
C	C/1	D/0	E/1	A/0
D	D/0	B/0	A/0	F/0
E	C/1	F/0	E/1	A/0
F	D/0	D/0	A/0	F/0
G	G/0	G/0	A/0	A/0
H	B/1	D/0	E/1	A/0

P1 = (A, D, F, G) (B, C, E, H)   Using checking the outputs

P2 = (A, F, G) (D) (B, C, E, H)

P3 = (A, F) (G) (D) (B, C, H) (E)

P3 = P4

Sample 4 (cont’d)

	x_1x_2 = 00	01	11	10
A	D/0	D/0	F/0	A/0
B	C/1	D/0	E/1	F/0
C	C/1	D/0	E/1	A/0
D	D/0	B/0	A/0	F/0
E	C/1	F/0	E/1	A/0
F	D/0	D/0	A/0	F/0
G	G/0	G/0	A/0	A/0
H	B/1	D/0	E/1	A/0

A’ = (A, F)
E’ = (G)
C’ = (D)
B’ = (B, C, H)
D’ = (E)

	x_1x_2 = 00	01	11	10
A’	C’/0	C’/0	A’/0	A’/0
B’	B’/1	C’/0	D’/1	A’/0
C’	C’/0	B’/0	A’/0	A’/0
D’	B’/1	A’/0	D’/1	A’/0
E’	E’/0	E’/0	A’/0	A’/0

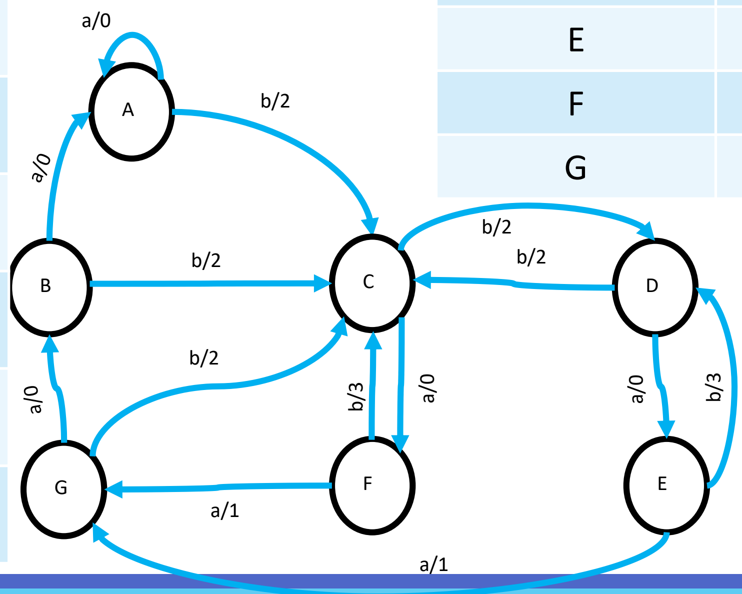
Implication Table

Implication Table: Big Picture

Iran University of Science & Technology
IUST

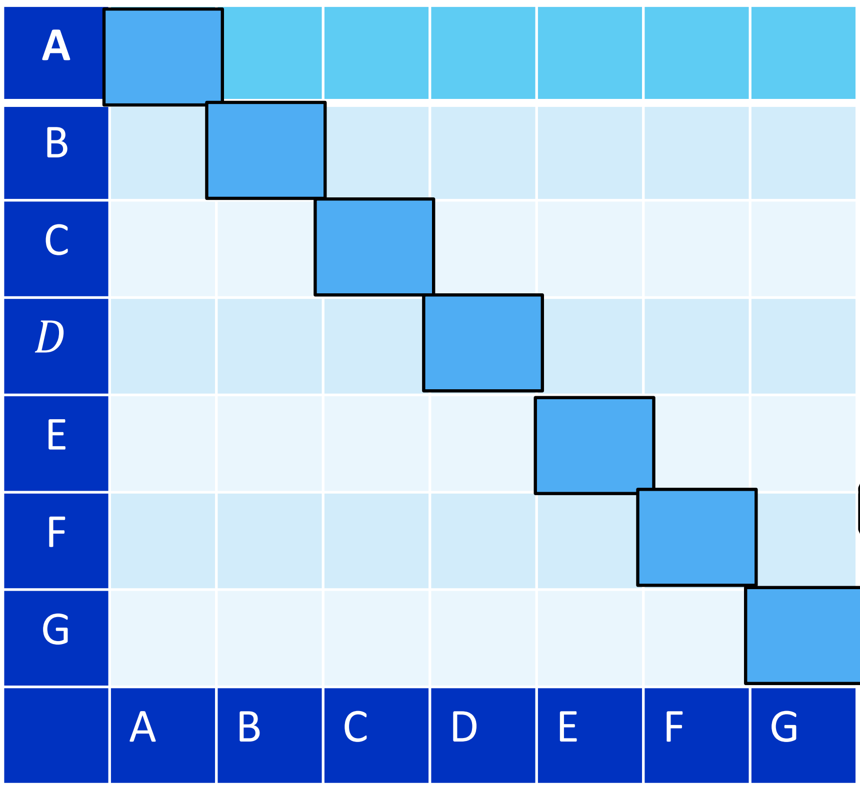
A							
B							
C							
D							
E							
F							
G							
	A	B	C	D	E	F	G

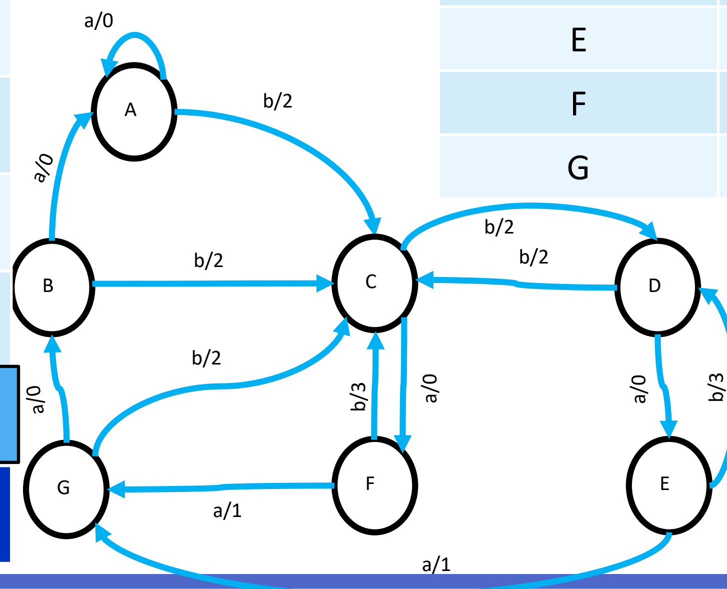
Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2



Do We Need All the Cubes?

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2



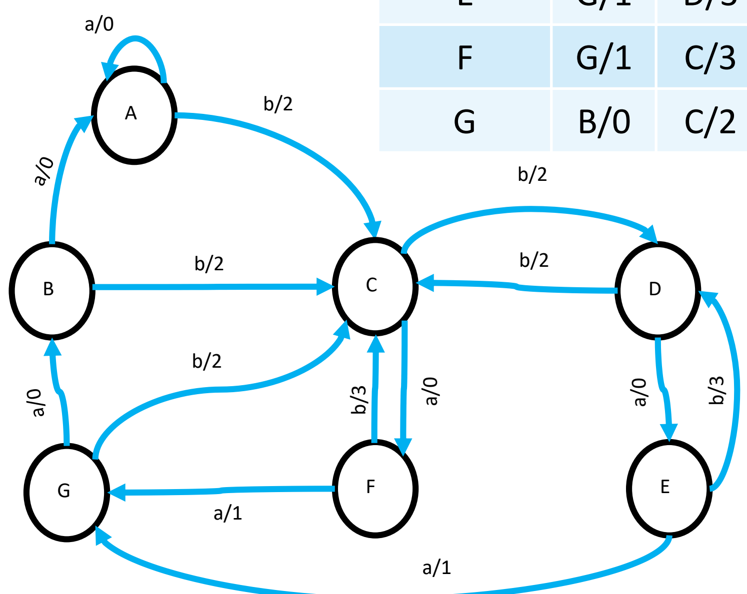


Implication Table: Step 1

- Draw a chart that has a square for each pair of states

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

B						
C						
D						
E						
F						
G						
	A	B	C	D	E	F

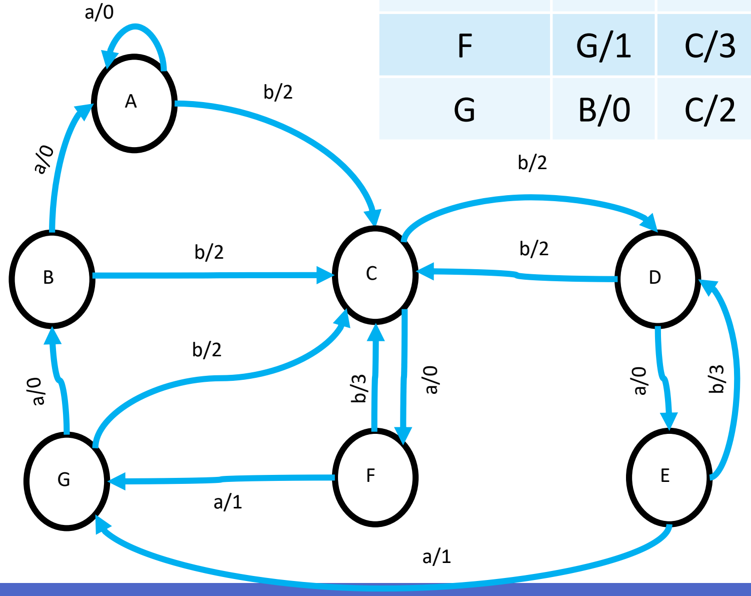


Implication Table: Step 2

- Find incompatible States
- Put X in the corresponding square

Present State	Next State a	Next State b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

B						
C						
D						
E						
F						
G						
	A	B	C	D	E	F

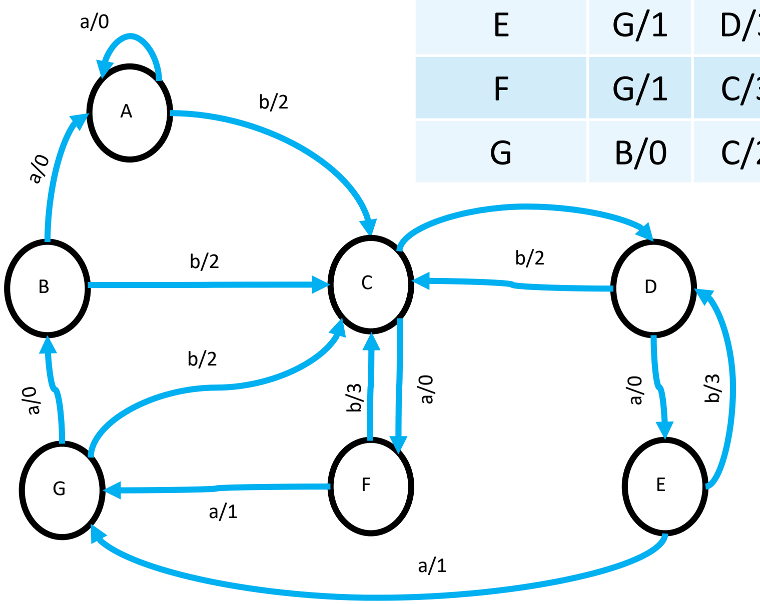


Implication Table: Step 2 (cont'd)

- Find incompatible States
- Put X in the corresponding square

B						
C						
D						
E	X					
F	X					
G						
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

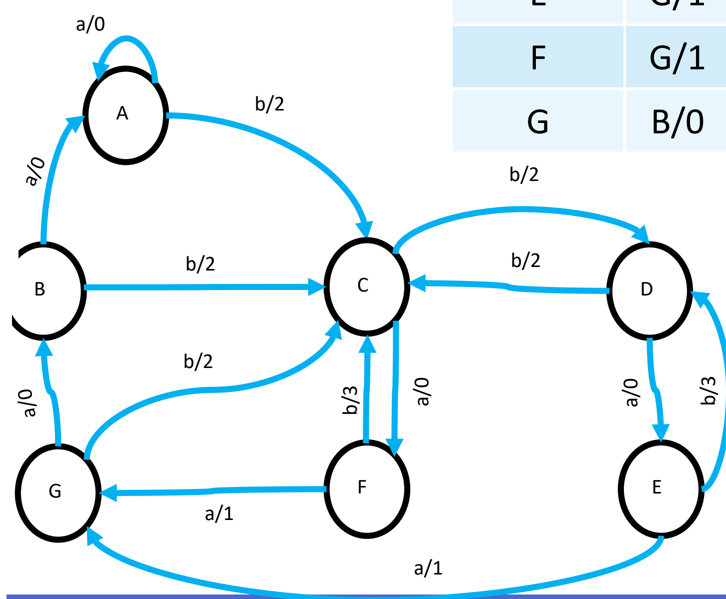


Implication Table: Step 2 (cont’d)

- Find incompatible States
- Put X in the corresponding square

Present State	Next State a	Next State b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

B						
C						
D						
E	X	X				
F	X	X				
G						
	A	B	C	D	E	F

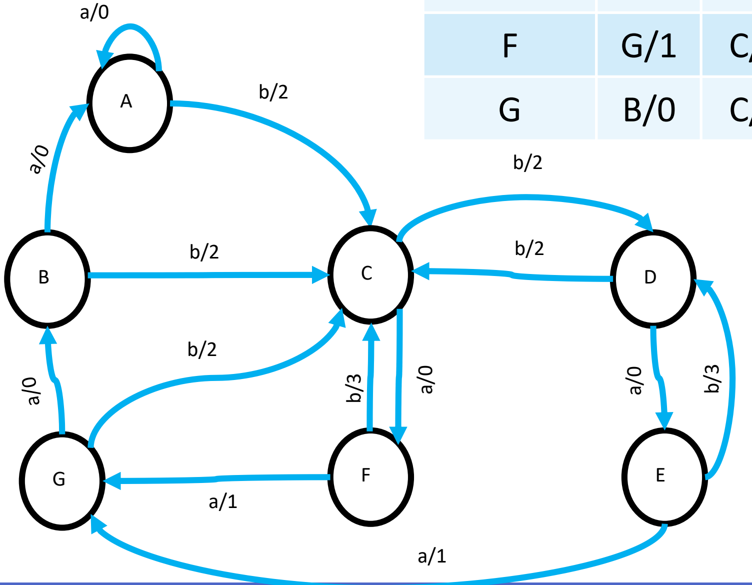


Implication Table: Step 2 (cont'd)

- Find incompatible States
- Put X in the corresponding square

Present State	Next State a	Next State b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

B						
C						
D						
E	X	X	X			
F	X	X	X			
G						
	A	B	C	D	E	F

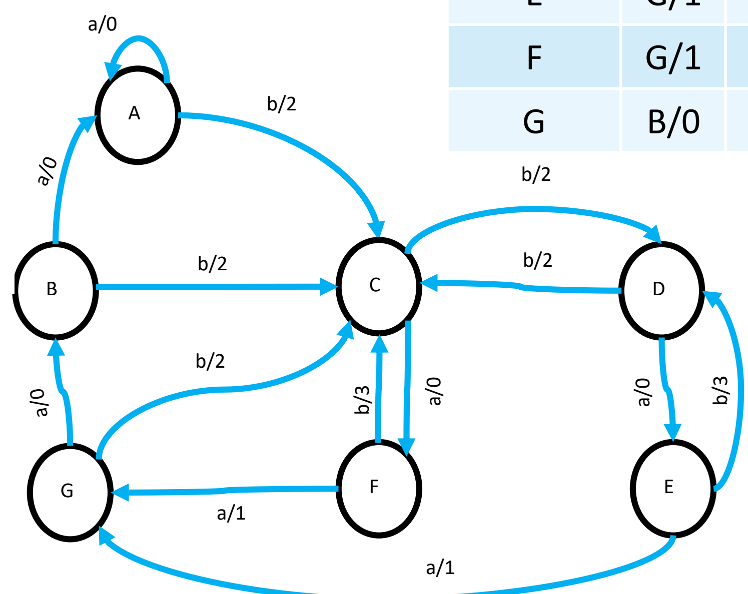


Implication Table: Step 2 (cont'd)

- Find incompatible States
- Put X in the corresponding square

B						
C						
D						
E	X	X	X	X		
F	X	X	X	X		
G						
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

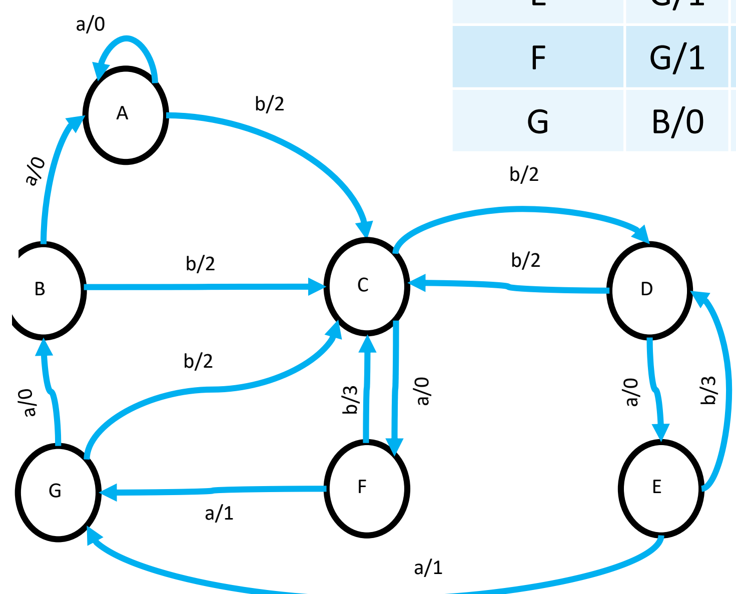


Implication Table: Step 2 (cont'd)

- Find incompatible States
- Put X in the corresponding square

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

B						
C						
D						
E	X	X	X	X		
F	X	X	X	X		
G					X	X
	A	B	C	D	E	F

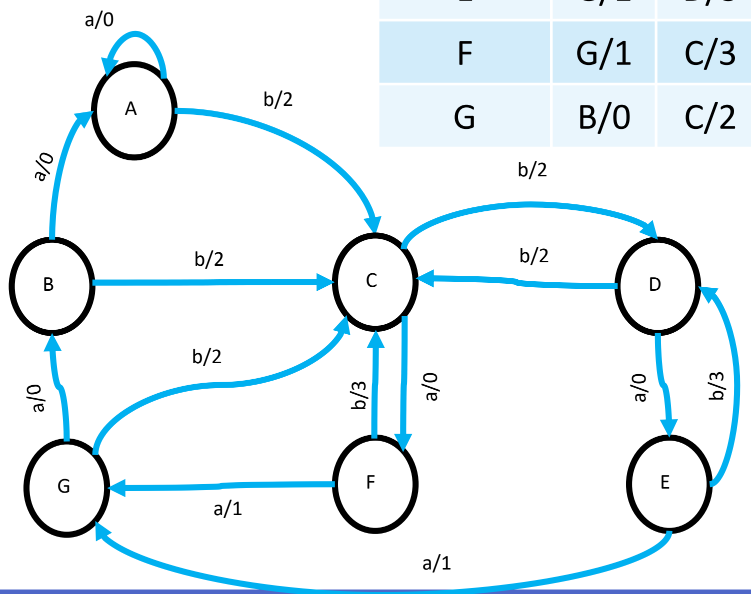


Implication Table: Step 3

- Enter implied pair in non square
  - Put *V* for equivalent states
  - Write conditional states for conditional equivalent states

B						
C						
D						
E	X	X	X	X		
F	X	X	X	X		
G					X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

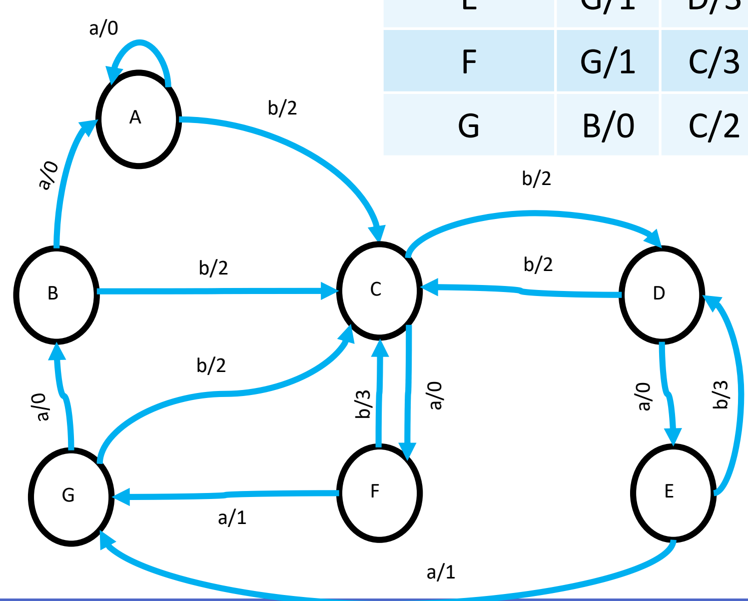


Implication Table: Step 3 (cont'd)

- Enter implied pair in non square
  - Put V for equivalent states
  - Write conditional states for conditional equivalent states

	A	B	C	D	E	F
B	V					
C						
D						
E	X	X	X	X		
F	X	X	X	X		
G					X	X

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

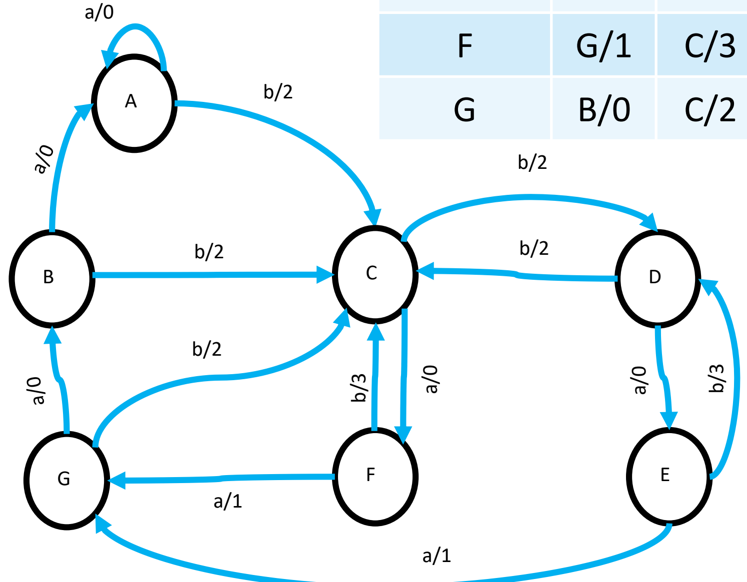


Implication Table: Step 3 (cont’d)

- Enter implied pair in non square
  - Put *V* for equivalent states
  - Write conditional states for conditional equivalent states

B	*V*					
C	A=F C=D					
D	A=E					
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B				X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

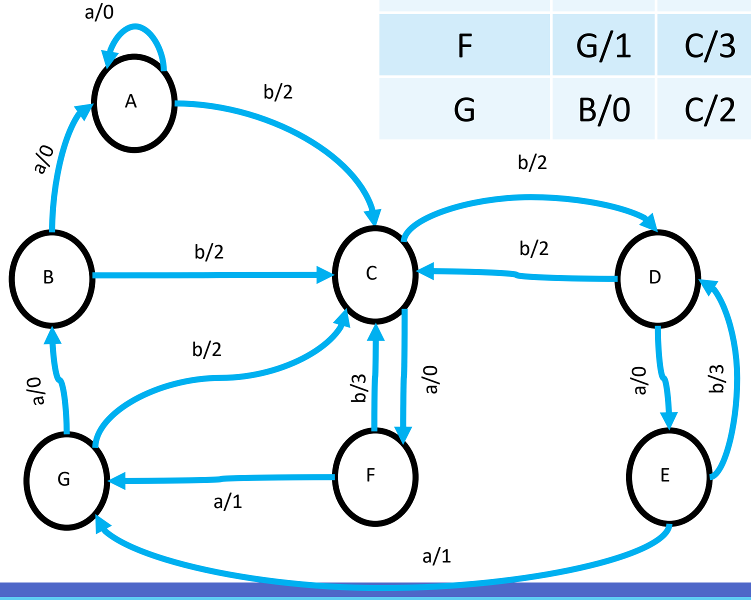


Implication Table: Step 3 (cont'd)

- Enter implied pair in non square
  - Put *V* for equivalent states
  - Write conditional states for conditional equivalent states

B	*V*					
C	A=F C=D	A=F C=D				
D	A=E	A=E				
E	X	X	X	X		
F	X	X	X	X		
G	A=B	A=B			X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

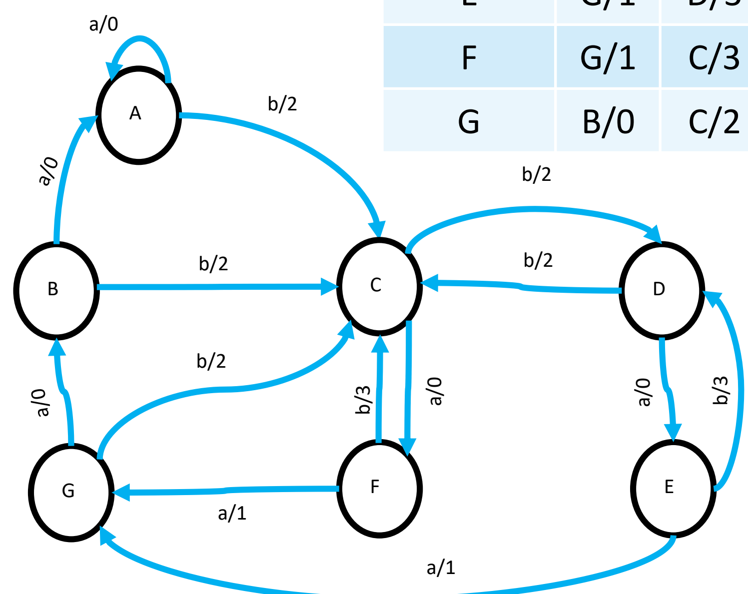


Implication Table: Step 3 (cont’d)

- Enter implied pair in non square
  - Put *V* for equivalent states
  - Write conditional states for conditional equivalent states

B	*V*					
C	A=F C=D	A=F C=D				
D	A=E	A=E	C=D E=F			
E	X	X	X	X		
F	X	X	X	X		
G	A=B	A=B	C=D B=F		X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

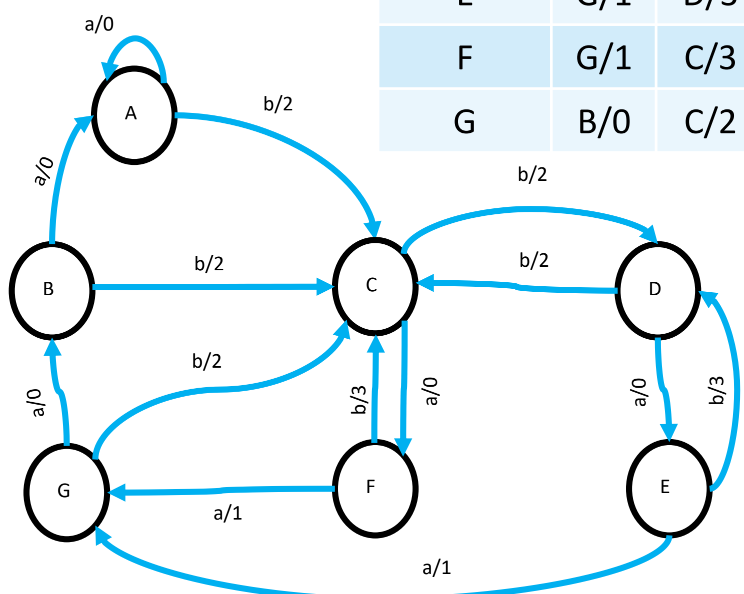


Implication Table: Step 3 (cont'd)

- Enter implied pair in non square
  - Put V for equivalent states
  - Write conditional states for conditional equivalent states

B	V					
C	A=F C=D	A=F C=D				
D	A=E	A=E	C=D E=F			
E	X	X	X	X		
F	X	X	X	X		
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

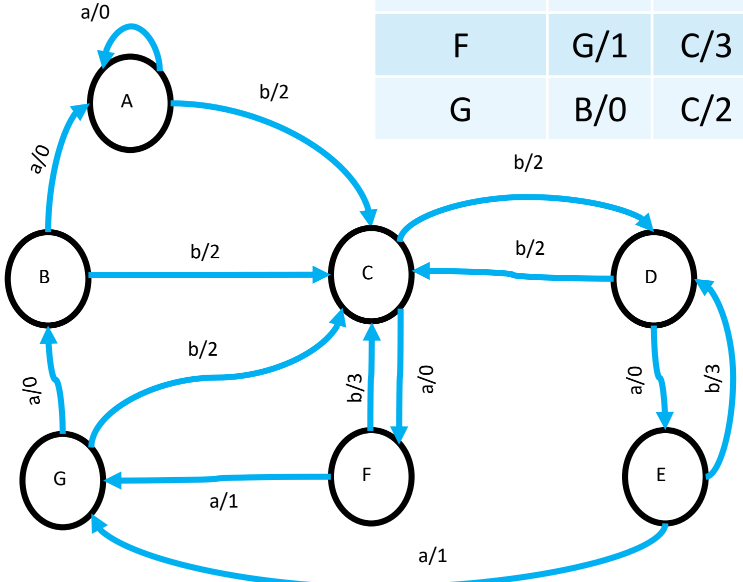


Implication Table: Step 3 (cont'd)

- Enter implied pair in non square
  - Put V for equivalent states
  - Write conditional states for conditional equivalent states

B	V					
C	A=F C=D	A=F C=D				
D	A=E	A=E	C=D E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

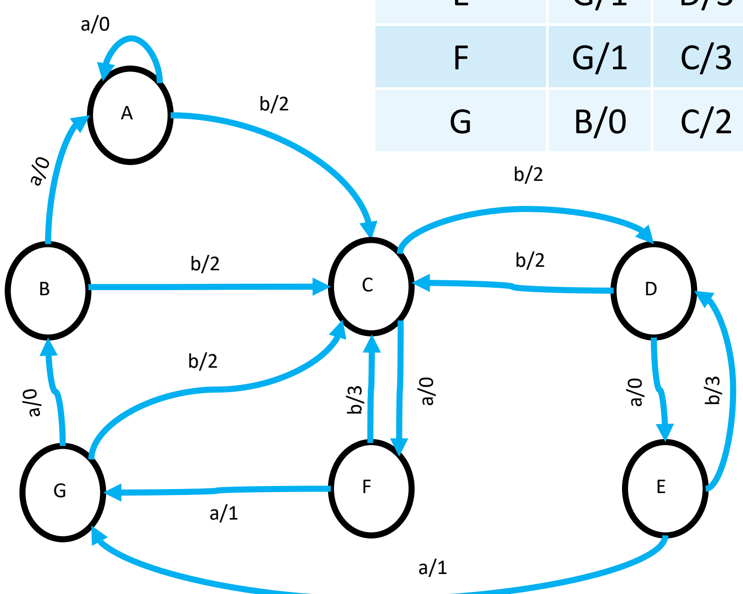


Implication Table: Step 4

- Remove self redundant pairs
  - C-D

B	*V*					
C	A=F C=D	A=F C=D				
D	A=E	A=E	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	**B**	**C**	**D**	**E**	**F**

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

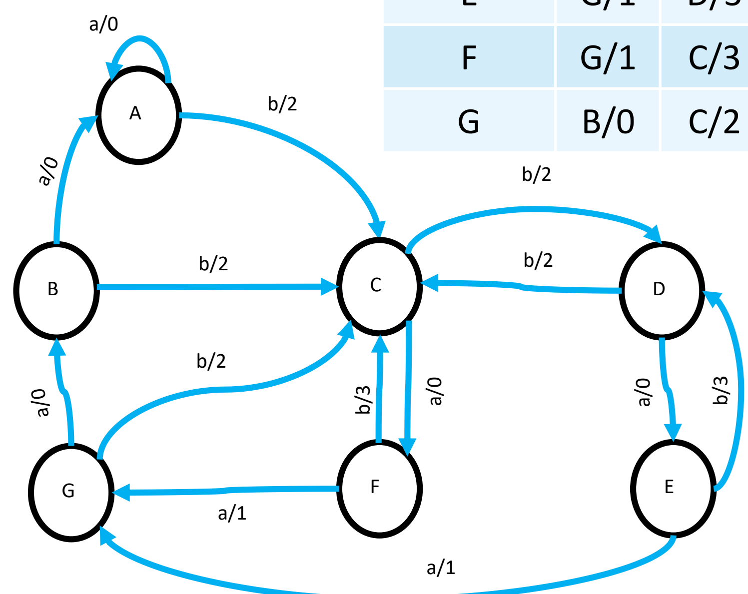


Implication Table: Step 5

- Find squares with implied pairs that are not equivalent
- Put x
  - A-E are not compatible
  - => Each square that has A-E is incompatible too

Present State	Next State a	Next State b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

B	*V*					
C	A=F C=D	A=F C=D				
D	A=E	A=E	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

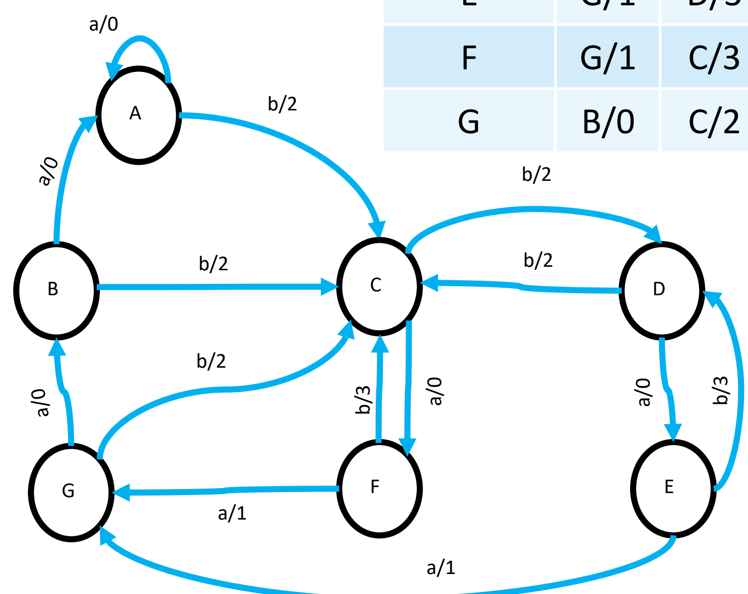


Implication Table: Step 5 (cont'd)

- Find squares with implied pairs that are not equivalent
- Put x
  - A-E are not compatible
  - => Each square that has A-E is incompatible too

Present State	Next State a	Next State b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

B	V					
C	A=F C=D	A=F C=D				
D	~~A=E~~ (X)	A=E	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

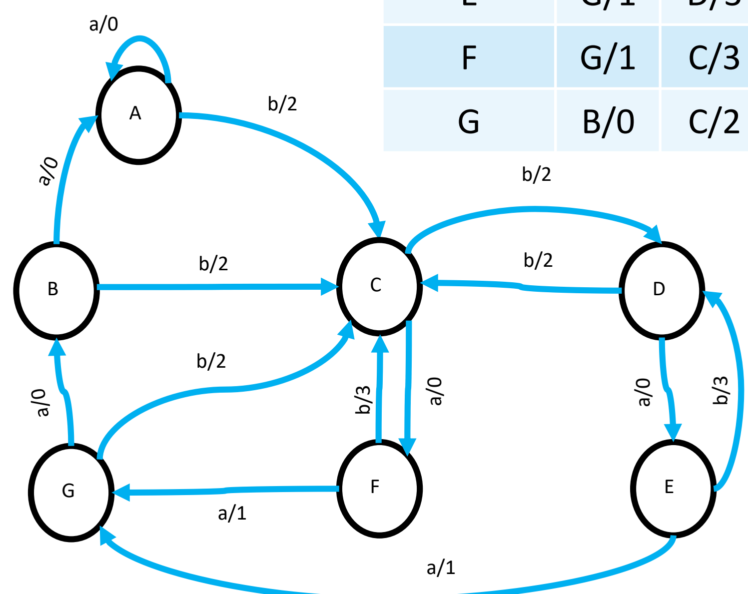


Implication Table: Step 5 (cont'd)

- Find squares with implied pairs that are not equivalent
- Put x
  - A-E are not compatible
  - => Each square that has A-E is incompatible too

B	V					
C	A=F C=D	A=F C=D				
D	~~A=E~~	~~A=E~~	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

Present State	Next State a	Next State b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

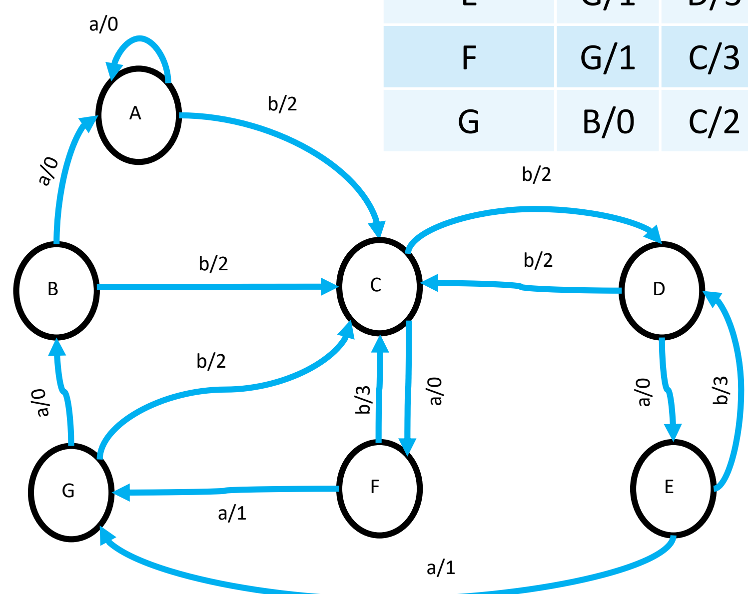


Implication Table: Step 5 (cont'd)

- Find squares with implied pairs that are not equivalent
- Put x
  - A-F are not compatible
  - => Each square that has A-F is incompatible too

B	V					
C	A=F C=D	A=F C=D				
D	~~A=F~~	~~A=F~~	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

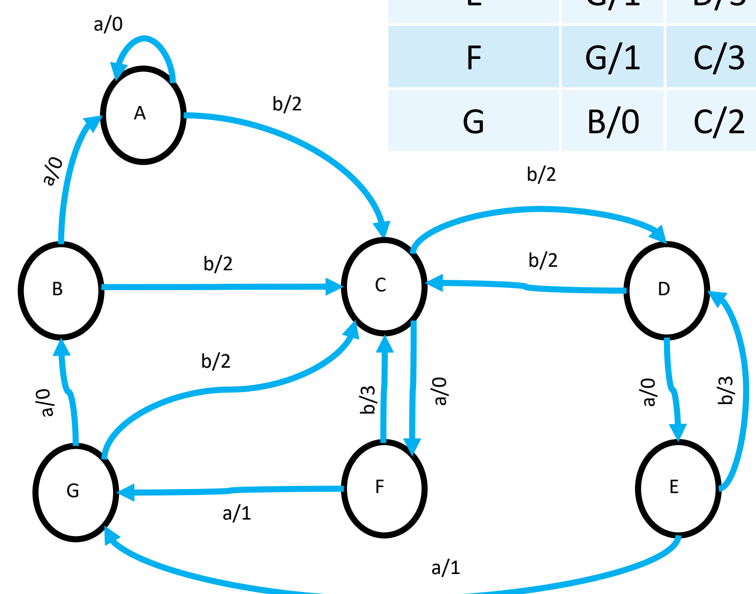


Implication Table: Step 5 (cont’d)

- Find squares with implied pairs that are not equivalent
- Put x
  - A-F are not compatible
  - => Each square that has A-F is incompatible too

B	V					
C	A=F C=D	A=F C=D				
D	A=E	A=E	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

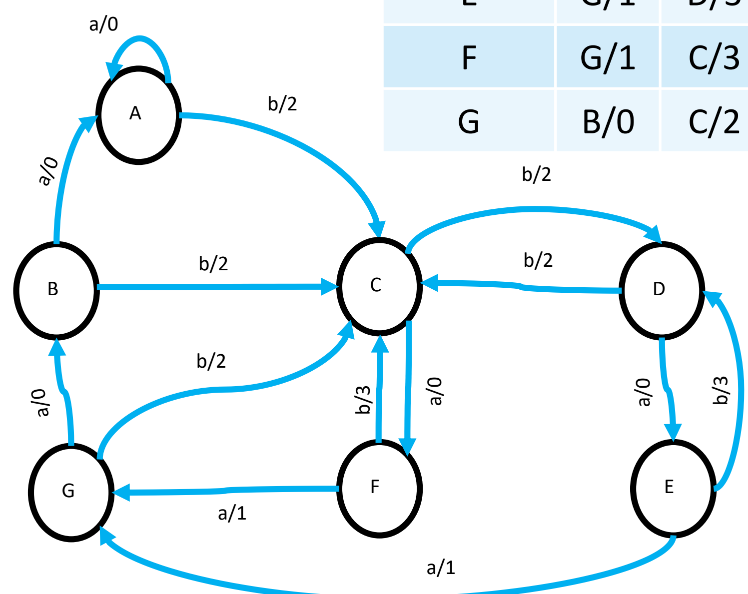


Implication Table: Step 5 (cont’d)

- Find squares with implied pairs that are not equivalent
- Put x
  - B-E are not compatible
  - => Each square that has B-E is incompatible too

B	V					
C	A=F C=D	A=F C=D				
D	A=E	A=E	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

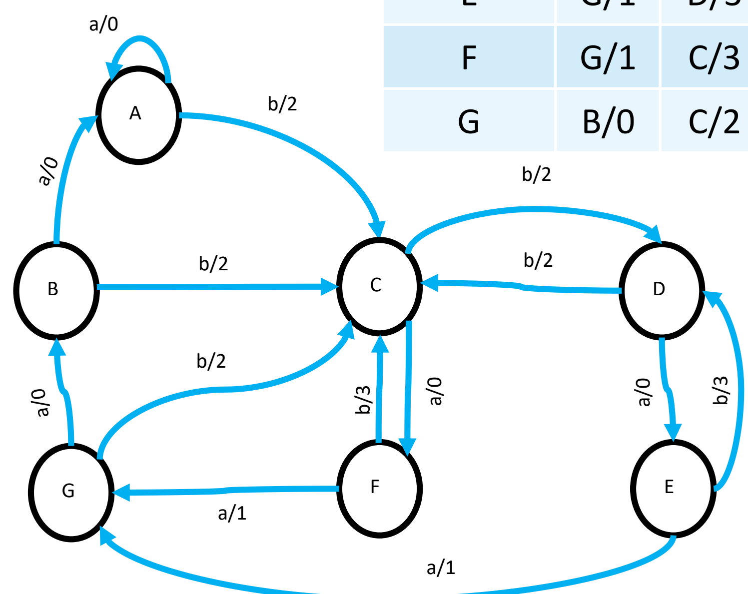


Implication Table: Step 5 (cont’d)

- Find squares with implied pairs that are not equivalent
- Put x
  - B-E are not compatible
  - => Each square that has B-E is incompatible too

B	V					
C	A=F C=D	A=F C=D				
D	A=E	A=E	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	B=E	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

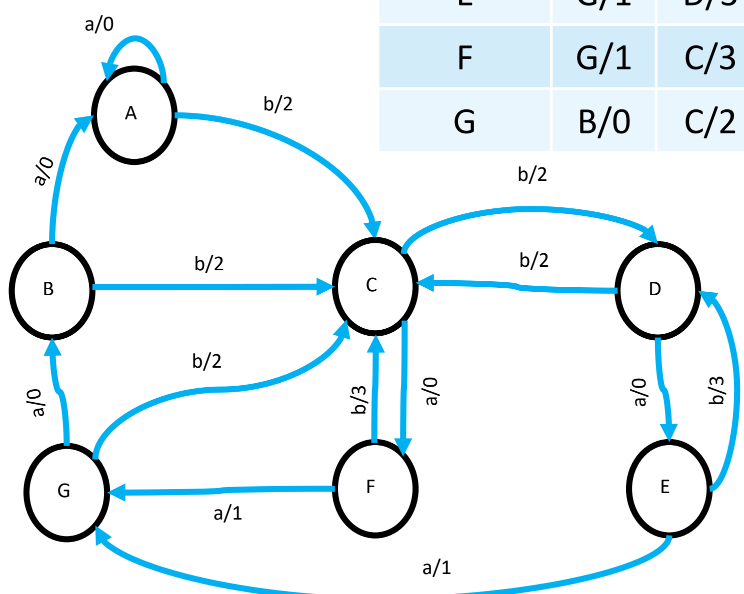


Implication Table: Step 5 (cont'd)

- Find squares with implied pairs that are not equivalent
- Put x
  - B-F are not compatible
  - => Each square that has B-F is incompatible too

B	V					
C	~~A=F C=D~~	~~A=F C=D~~				
D	~~A=E~~	~~A=E~~	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	C=D B=F	~~B=F~~	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

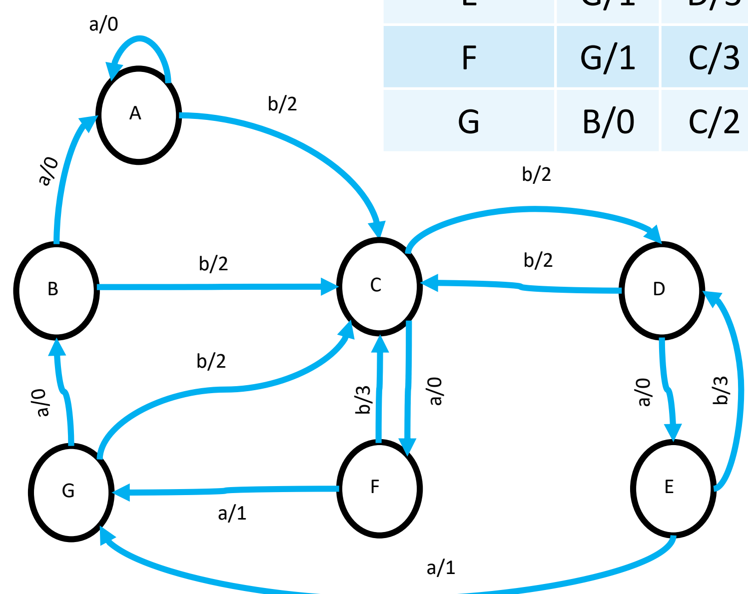


Implication Table: Step 6

- Find squares with implied pairs that are equivalent
- Put V
  - A-B are not compatible
  - => Each square that has A-B is incompatible too

B	V					
C	~~A=F C=D~~	~~A=F C=D~~				
D	~~A=E~~	~~A=E~~	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B	A=B	~~C=D B=F~~	~~B=E~~	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

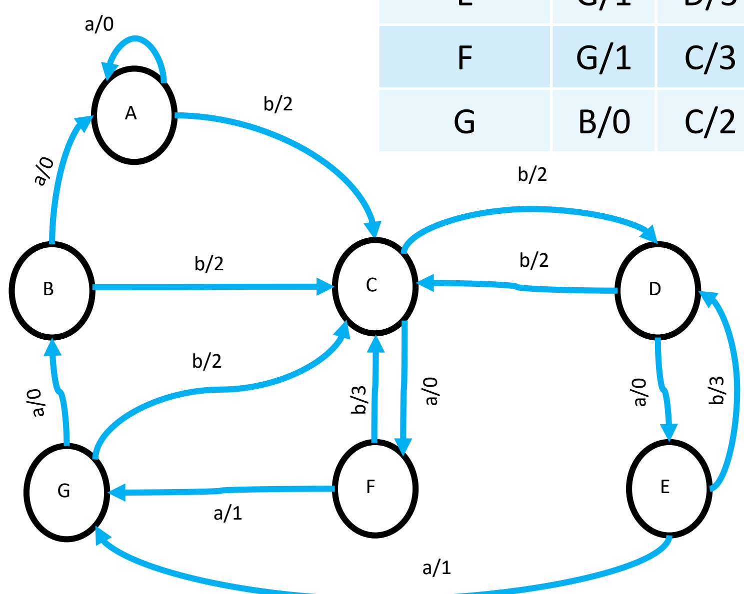


Implication Table: Step 6 (cont'd)

- Find squares with implied pairs that are equivalent
- Put V
  - A-B are not compatible
  - => Each square that has A-B is incompatible too

B	V					
C	~~A=F C=D~~	~~A=F C=D~~				
D	~~A=E~~	~~A=E~~	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B V	A=B V	~~C=D B=F~~	~~B=E~~	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

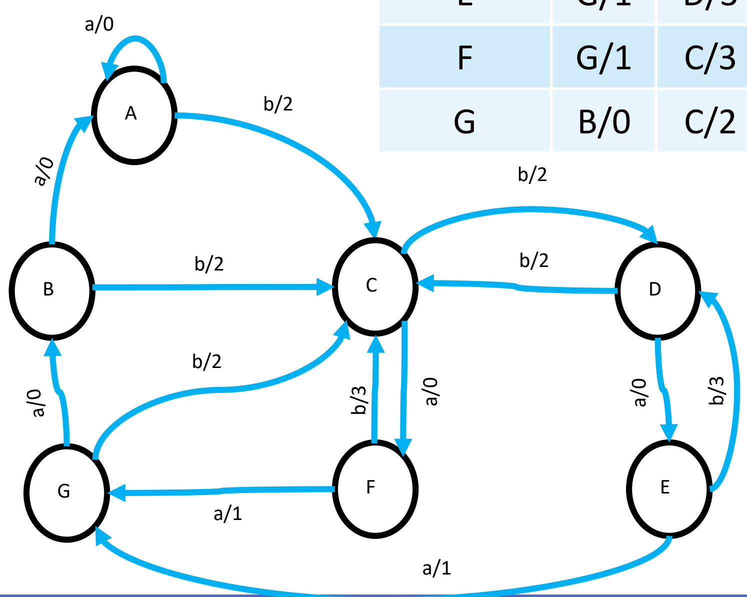


Implication Table: Step 6 (cont'd)

- Find squares with implied pairs that are equivalent
- Put *V*
  - C-D is equivalent if E=F be equivalent
  - E-F is equivalent if C-D be equivalent
  - => They are equal

B	*V*					
C	A=F C=D	A=F C=D				
D	A=E	A=E	~~C=D~~ E=F			
E	X	X	X	X		
F	X	X	X	X	C=D	
G	A=B *V*	A=B *V*	C=D B=F	B=E	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2



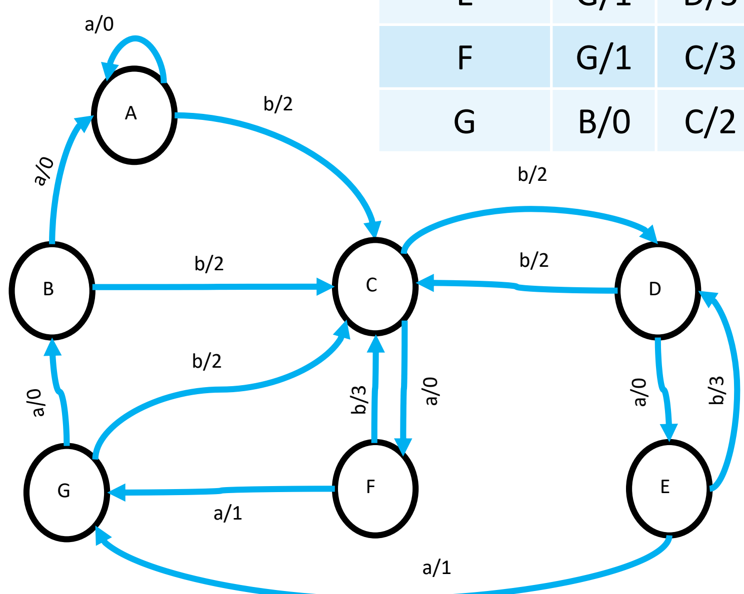
Iran University of Science & Technology

Implication Table: Step 6 (cont'd)

- Find squares with implied pairs that are equivalent
- Put *V*
  - C-D is equivalent if E=F be equivalent
  - E-F is equivalent if C-D be equivalent
  - => They are equal

B	*V*					
C	~~A=F C=D~~	~~A=F C=D~~				
D	~~A=E~~	~~A=E~~	~~C=D~~ E=F *V*			
E	X	X	X	X		
F	X	X	X	X	C=D *V*	
G	A=B *V*	A=B *V*	~~C=D B=F~~	~~B=E~~	X	X
	A	B	C	D	E	F

Present State	Next State a	Next State b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

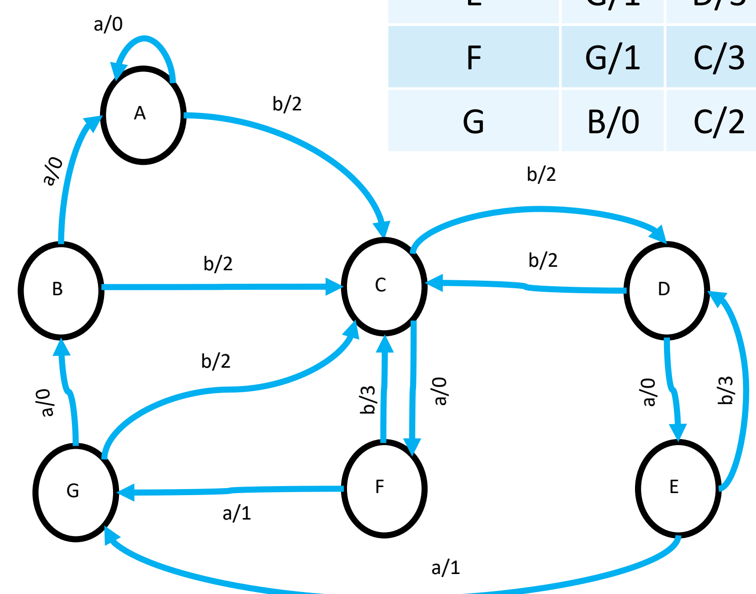


Implication Table: Step 7

- Reduce transition table
  - Remove equivalent states

Present State	Next State a	Next State b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

	A	B	C	D	E	F
B	V					
C	A=F C=D	A=F C=D				
D	A=E	A=E	C=D E=F V			
E	X	X	X	X		
F	X	X	X	X	C=D V	
G	A=B V	A=B V	C=D B=F	B=E	X	X

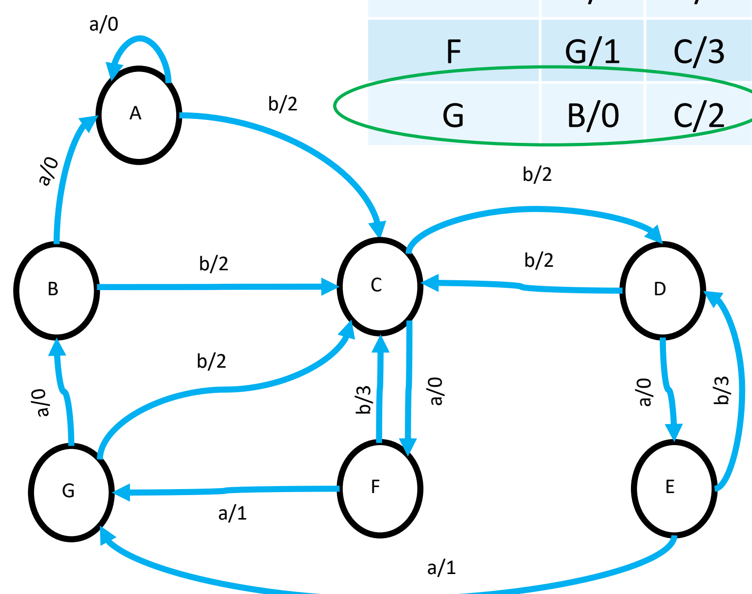


Implication Table: Step 7 (cont'd)

- Reduce transition table
  - Remove equivalent states

Present State	**Next State**	
	a	b
A	A/0	C/2
B	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3
G	B/0	C/2

B	V					
C	A=F C=D	A=F C=D				
D	A=E	A=E	C=D E=F V			
E	X	X	X	X		
F	X	X	X	X	C=D V	
G	A=B V	A=B V	C=D B=F	B=F	X	X
	A	B	C	D	E	F

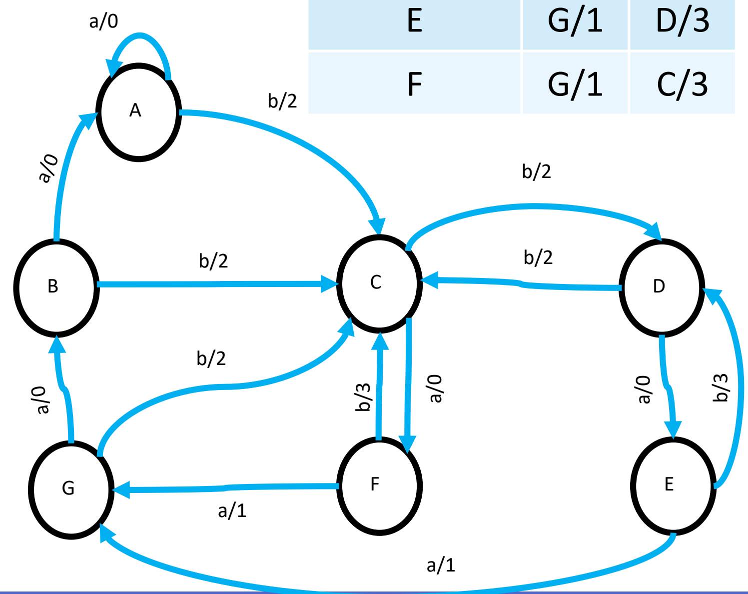


Implication Table: Step 7 (cont'd)

- Reduce transition table
  - Remove equivalent states

Present State	Next State a	Next State b
A,B,G	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3

	A	B	C	D	E	F
B	V					
C	~~A=F C=D~~	~~A=F C=D~~				
D	~~A=E~~	~~A=E~~	~~C=D~~ E=F V			
E	X	X	X	X		
F	X	X	X	X	C=D V	
G	A=B V	A=B V	~~C=D B=F~~	~~B=E~~	X	X

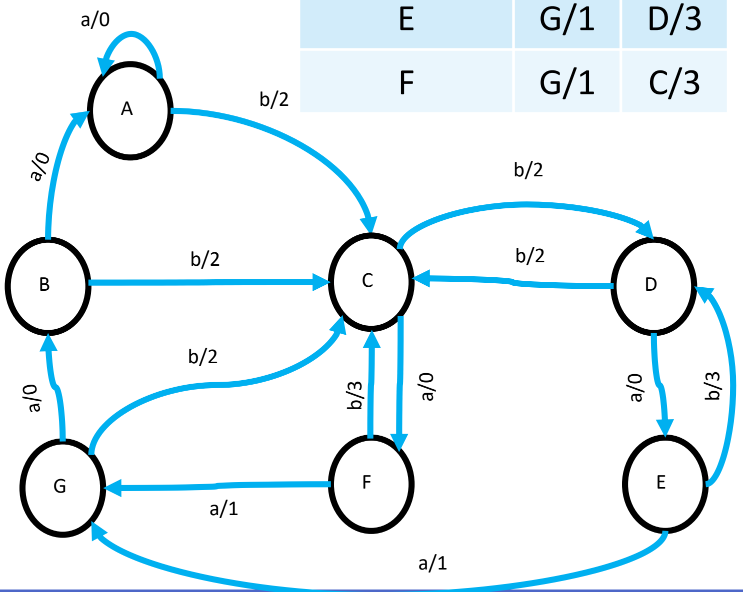


Implication Table: Step 7 (cont'd)

- Reduce transition table
  - Remove equivalent states

Present State	Next State a	Next State b
A,B,G	A/0	C/2
C	F/0	D/2
D	E/0	C/2
E	G/1	D/3
F	G/1	C/3

B	V					
C	~~A=F C=D~~	~~A=F C=D~~				
D	~~A=E~~	~~A=E~~	~~C=D~~ E=F V			
E	X	X	X	X		
F	X	X	X	X	C=D V	
G	A=B V	A=B V	~~C=D B=F~~	~~B=E~~	X	X
	A	B	C	D	E	F

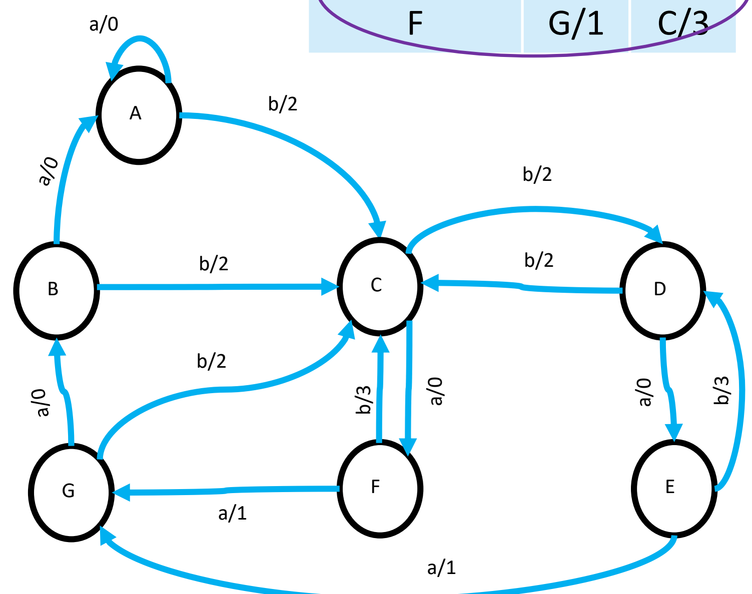


Implication Table: Step 7 (cont'd)

- Reduce transition table
  - Remove equivalent states

Present State	Next State a	Next State b
A,B,G	A/0	C/2
C, D	F/0	D/2
E	G/1	D/3
F	G/1	C/3

	A	B	C	D	E	F
B	V					
C	A=F C=D	A=F C=D				
D	A=F	A=F	C=D E=F V			
E	X	X	X	X		
F	X	X	X	X	C=D V	
G	A=B V	A=B V	C=D B=F	B=F	X	X



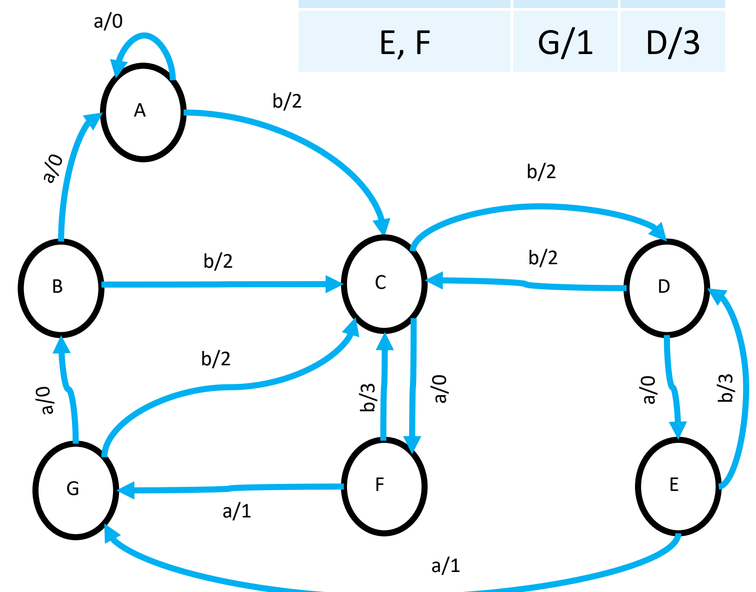
Implication Table: Step 7 (cont’d)

Iran University of Science & Technology
IUST

- Reduce transition table
  - Remove equivalent states

B	√					
C	~~A=F C=D~~	~~A=F C=D~~				
D	~~A=E~~	~~A=E~~	~~C=D~~ E=F √			
E	X	X	X	X		
F	X	X	X	X	C=D √	
G	A=B √	A=B √	~~C=D B=F~~	~~B=E~~	X	X
	A	B	C	D	E	F

Present State	Next State	
	a	b
A, B, G	A/0	C/2
C, D	F/0	D/2
E, F	G/1	D/3



Thank You