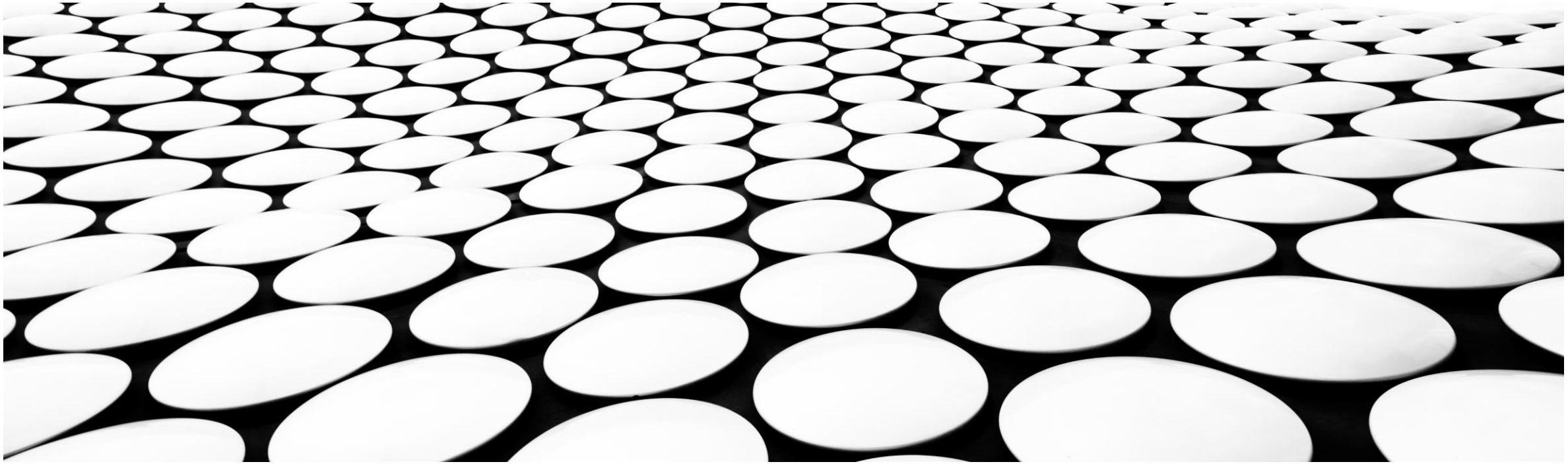
EXAMPLES: **DEADLOCKS**

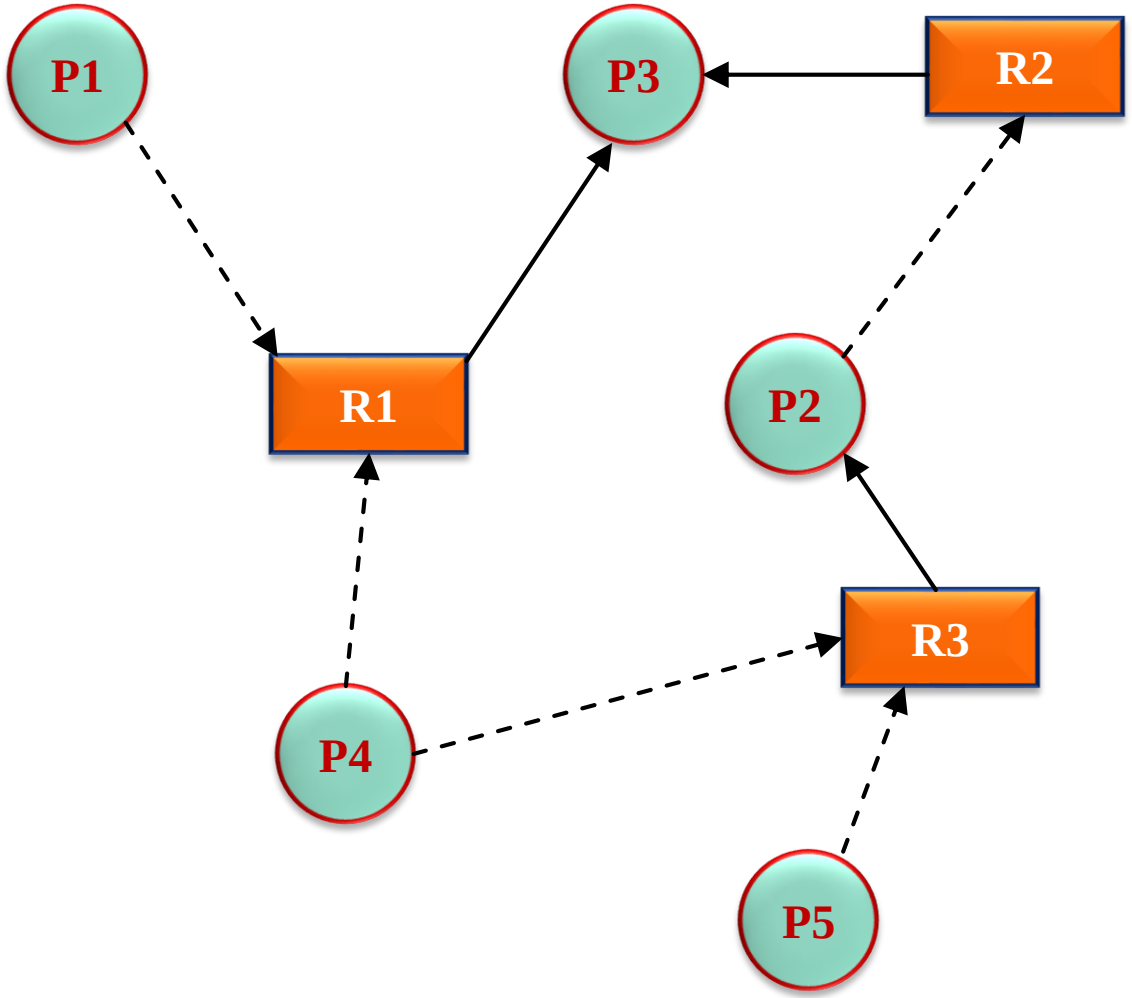
DR. ARIJIT ROY

COMPUTER SCIENCE AND ENGINEERING GROUP

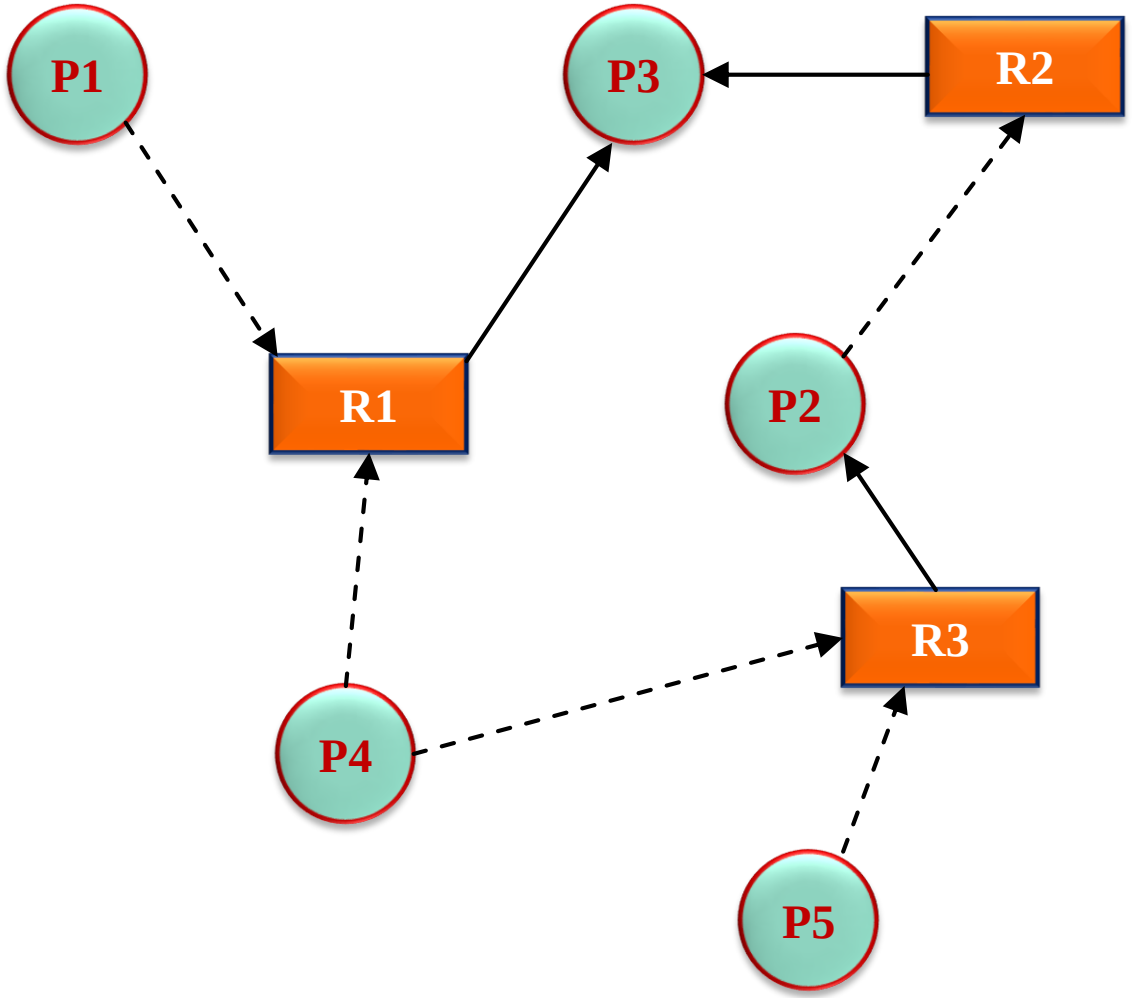
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY



EXAMPLE 1: IDENTIFY IF THERE WILL BE A DEADLOCK

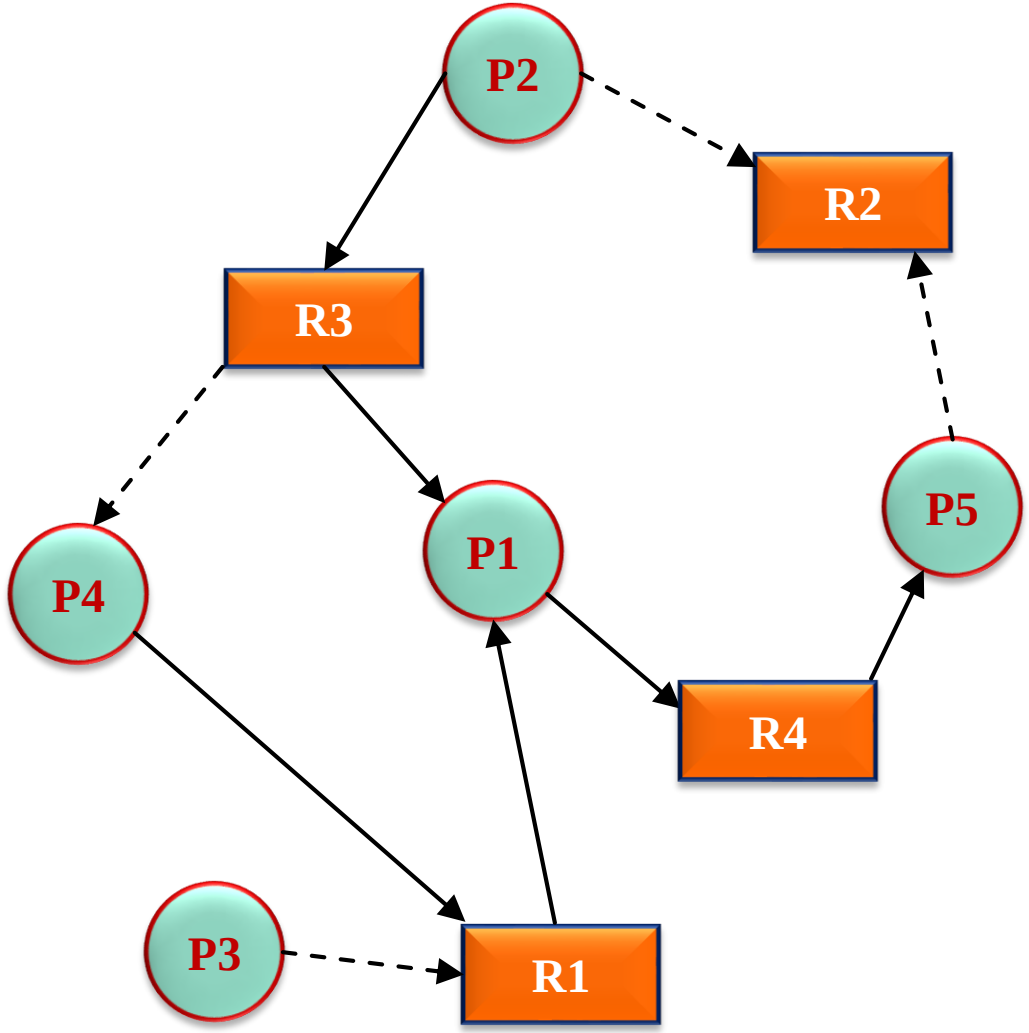


EXAMPLE 1: IDENTIFY IF THERE WILL BE A DEADLOCK

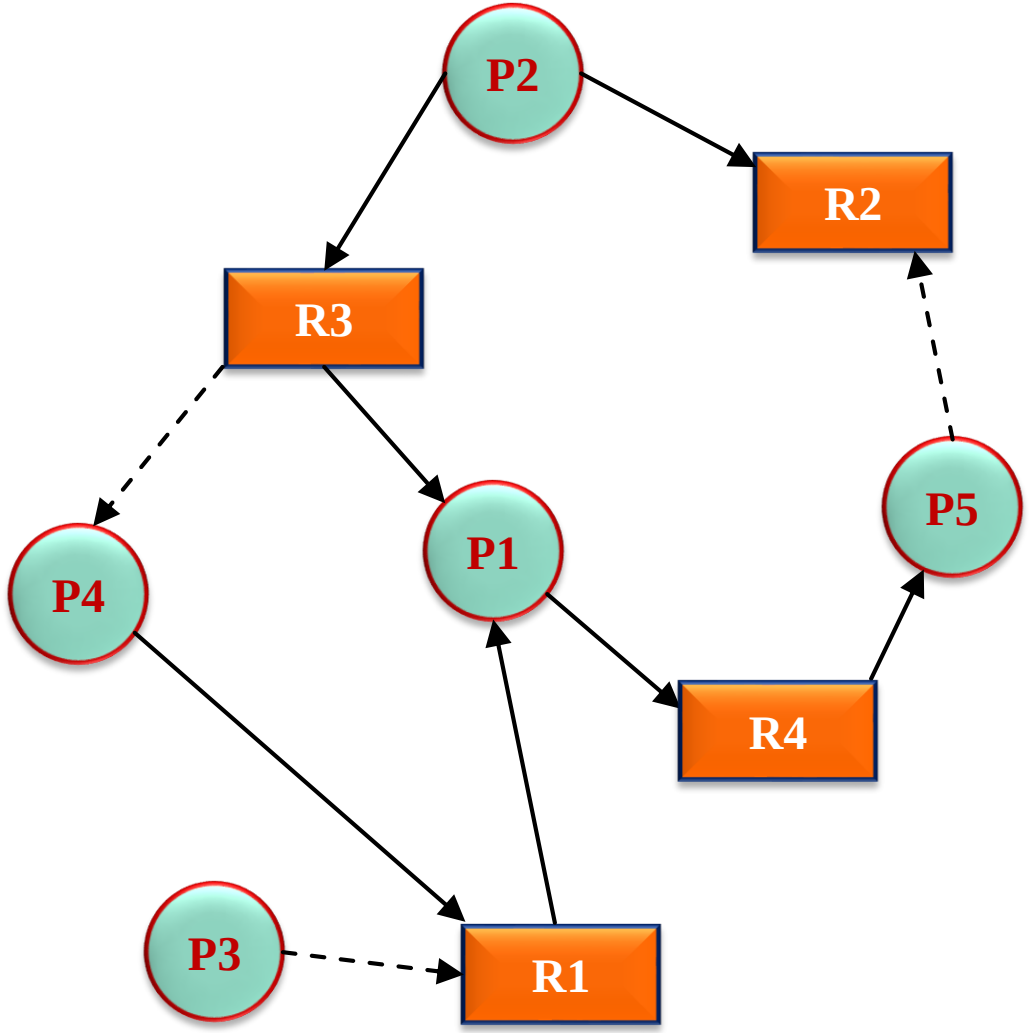


- No deadlock
- Most of the processes are independent
- Only possibility is P2, as it is holding R3 and requesting for R2 (which is holding by P3)
- P3 has already R1 and R3, so it can complete its execution and release the resources
- On releasing R2 by P3, P2 may acquire it.

EXAMPLE 2: IDENTIFY IF THERE WILL BE A DEADLOCK

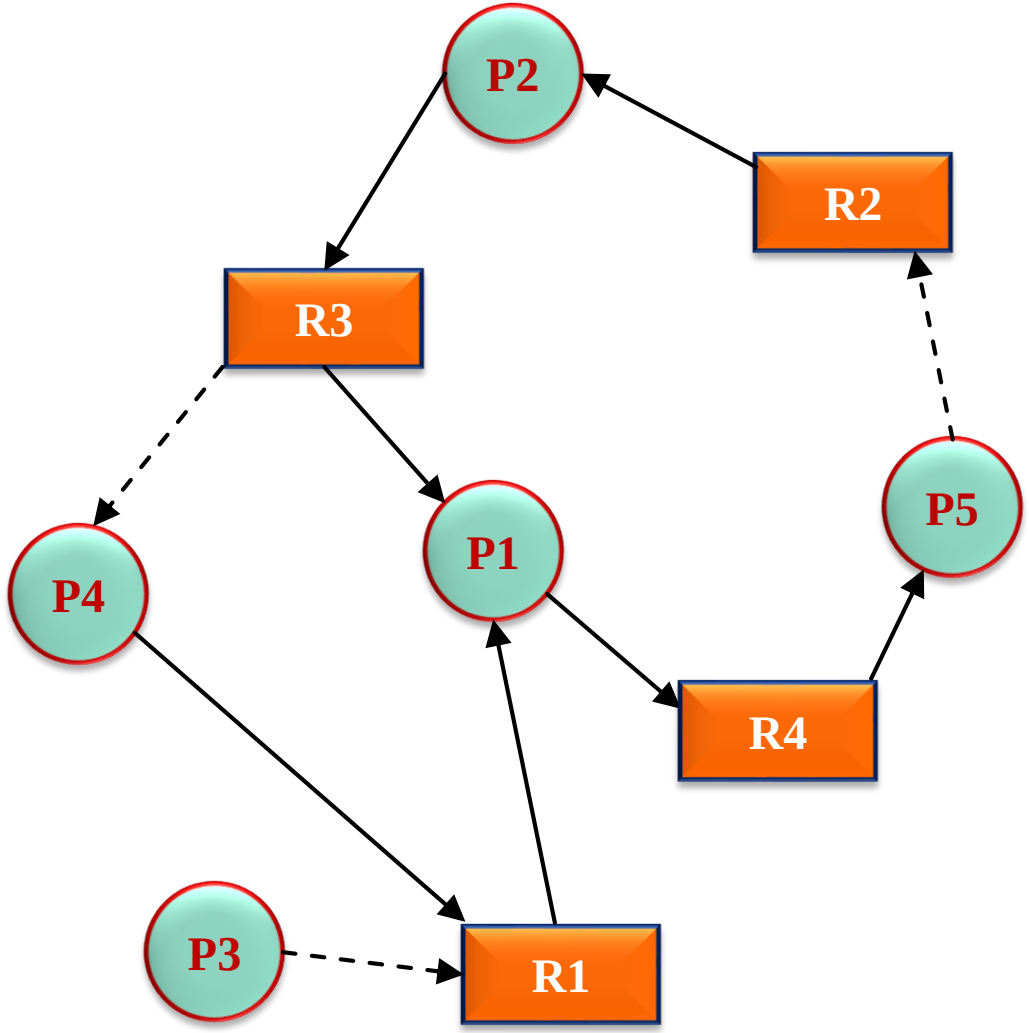


EXAMPLE 2: IDENTIFY IF THERE WILL BE A DEADLOCK



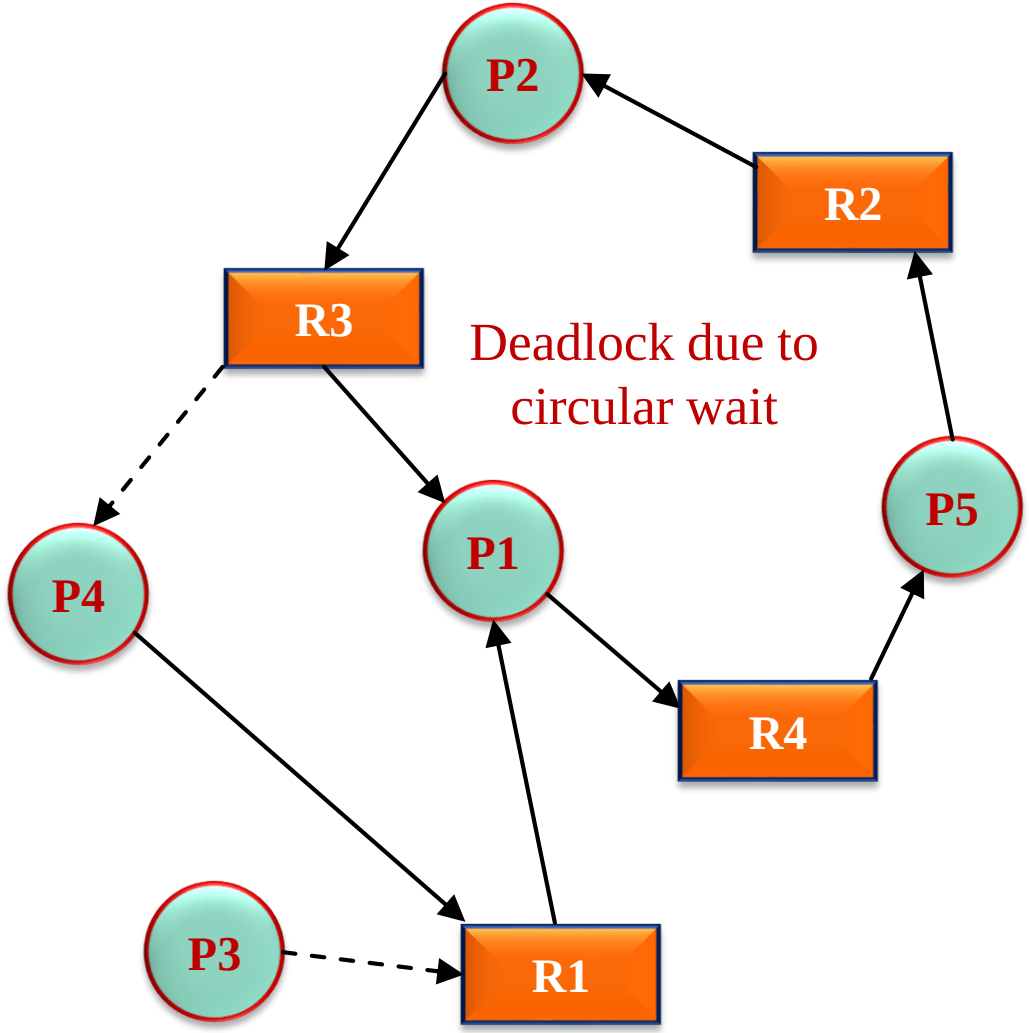
- P2 requests for R2

EXAMPLE 2: IDENTIFY IF THERE WILL BE A DEADLOCK



- P2 requests for R2
- R2 is allocated to P2

EXAMPLE 2: IDENTIFY IF THERE WILL BE A DEADLOCK



- P2 requests for R2
- R2 is allocated to P2
- P5 requests for R2

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Process	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7	2	2	5
P1	0	1	2	2	2	3			
P2	3	1	0	4	2	3			
P3	0	1	1	4	5	6			

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7	2	2	5	3	5	7
P1	0	1	2	2	2	3				2	1	1
P2	3	1	0	4	2	3				1	1	3
P3	0	1	1	4	5	6				4	4	5

- Calculate Need matrix

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7	2	2	5	3	5	7
P1	0	1	2	2	2	3				2	1	1
P2	3	1	0	4	2	3				1	1	3
P3	0	1	1	4	5	6				4	4	5

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7	2	2	5	3	5	7
P1	0	1	2	2	2	3				2	1	1
P2	3	1	0	4	2	3				1	1	3
P3	0	1	1	4	5	6				4	4	5

With the available resources, P1 and P2 can be served

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7	2	2	5	3	5	7
P1	0	1	2	2	2	3				2	1	1
P2	3	1	0	4	2	3				1	1	3
P3	0	1	1	4	5	6				4	4	5

Start with P1
<P1>

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7				3	5	7
P1	0	1	2	2	2	3	2	3	7			
P2	3	1	0	4	2	3				1	1	3
P3	0	1	1	4	5	6				4	4	5

Already done

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7				3	5	7
P1	0	1	2	2	2	3	2	3	7			
P2	3	1	0	4	2	3				1	1	3
P3	0	1	1	4	5	6				4	4	5

Already done

Next we can allocate P2

<P1, P2>

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7				3	5	7
P1	0	1	2	2	2	3						
P2	3	1	0	4	2	3	5	4	7			
P3	0	1	1	4	5	6				4	4	5

Already done

Already done

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7				3	5	7
P1	0	1	2	2	2	3						
P2	3	1	0	4	2	3	5	4	7			
P3	0	1	1	4	5	6				4	4	5

Already done

Already done

Next we can allocate to P3

<P1, P2, P3>

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7				3	5	7
P1	0	1	2	2	2	3						
P2	3	1	0	4	2	3						
P3	0	1	1	4	5	6	5	5	8			

Already done

Already done

Already done

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7				3	5	7
P1	0	1	2	2	2	3						
P2	3	1	0	4	2	3						
P3	0	1	1	4	5	6	5	5	8			

Finally we can allocate to P0
<P1, P2, P3, P0>

Already done

Already done

Already done

EXAMPLE 3: HOW TO AVOID DEADLOCK

- Total number of resources: 3
- A=7, B=7, C=8
- Process: 4 (P_0-P_3)

Available resource

Process	Allocation			Max			Available			Need		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	2	2	0	5	7	7						
P1	0	1	2	2	2	3						
P2	3	1	0	4	2	3						
P3	0	1	1	4	5	6	5	5	8			
							7	7	8			

Already done

Already done

Already done

Already done

One of the safe sequences: <P1, P2, P3, P0>

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Process	Allocation				Max				Available			
	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2				
P_1	3	0	1	1	3	1	3	4				
P_2	1	2	0	2	6	4	2	3				
P_3	1	1	3	0	5	3	7	5				
P_4	0	3	1	1	7	4	4	3				

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2	1	3	3	4	6	3	2	2
P_1	3	0	1	1	3	1	3	4					0	1	2	3
P_2	1	2	0	2	6	4	2	3					5	2	2	1
P_3	1	1	3	0	5	3	7	5					4	2	4	5
P_4	0	3	1	1	7	4	4	3					7	1	3	2

Calculate Need matrix

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2	1	3	3	4	6	3	2	2
P_1	3	0	1	1	3	1	3	4					0	1	2	3
P_2	1	2	0	2	6	4	2	3					5	2	2	1
P_3	1	1	3	0	5	3	7	5					4	2	4	5
P_4	0	3	1	1	7	4	4	3					7	1	3	2

The available resources can be allocated to P_1

<P1>

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2					6	3	2	2
P_1	3	0	1	1	3	1	3	4	4	3	4	5				
P_2	1	2	0	2	6	4	2	3					5	2	2	1
P_3	1	1	3	0	5	3	7	5					4	2	4	5
P_4	0	3	1	1	7	4	4	3					7	1	3	2

Already done

<P1>

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2					6	3	2	2
P_1	3	0	1	1	3	1	3	4	4	3	4	5				
P_2	1	2	0	2	6	4	2	3					5	2	2	1
P_3	1	1	3	0	5	3	7	5					4	2	4	5
P_4	0	3	1	1	7	4	4	3					7	1	3	2

Already done

The available resources can be allocated to P_3

<P1, P3>

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2					6	3	2	2
P_1	3	0	1	1	3	1	3	4								
P_2	1	2	0	2	6	4	2	3	5	4	7	5	5	2	2	1
P_3	1	1	3	0	5	3	7	5								
P_4	0	3	1	1	7	4	4	3					7	1	3	2

Already done

Already done

<P1, P3>

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2					6	3	2	2
P_1	3	0	1	1	3	1	3	4								
P_2	1	2	0	2	6	4	2	3	5	4	7	5	5	2	2	1
P_3	1	1	3	0	5	3	7	5								
P_4	0	3	1	1	7	4	4	3					7	1	3	2

Already done

The available resources can be allocated to P_2

Already done

<P_1, P_3, P_2>

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2					6	3	2	2
P_1	3	0	1	1	3	1	3	4								
P_2	1	2	0	2	6	4	2	3								
P_3	1	1	3	0	5	3	7	5	6	6	7	7				
P_4	0	3	1	1	7	4	4	3					7	1	3	2

Already done

Already done

Already done

<P_1, P_3, P_2>

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2					6	3	2	2
P_1	3	0	1	1	3	1	3	4								
P_2	1	2	0	2	6	4	2	3								
P_3	1	1	3	0	5	3	7	5	6	6	7	7				
P_4	0	3	1	1	7	4	4	3					7	1	3	2

The available resources can be allocated to P_0

Already done

Already done

Already done

<P_1, P_3, P_2, P_0>

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2								
P_1	3	0	1	1	3	1	3	4								
P_2	1	2	0	2	6	4	2	3								
P_3	1	1	3	0	5	3	7	5								
P_4	0	3	1	1	7	4	4	3	8	6	9	7	7	1	3	2

Already done

Already done

Already done

Already done

$<P_1, P_3, P_2, P_0>$

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2								
P_1	3	0	1	1	3	1	3	4								
P_2	1	2	0	2	6	4	2	3								
P_3	1	1	3	0	5	3	7	5								
P_4	0	3	1	1	7	4	4	3	8	6	9	7	7	1	3	2

Already done

Already done

Already done

Already done

The available resources can be allocated to P_4

$<P_1, P_3, P_2, P_0, P_4>$

EXAMPLE 4: HOW TO AVOID DEADLOCK

- Find out the safe sequence
- Total types of resources:
  - Printer (P): 8 instances
  - Scanner (S): 9 instances
  - File (F): 10 instances
  - Keyboard (K): 8 instances
- Process: 5 (P_0-P_4)

Available resources

Process	Allocation				Max				Available				Need			
	P	S	F	K	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	8	3	4	2								
P_1	3	0	1	1	3	1	3	4								
P_2	1	2	0	2	6	4	2	3								
P_3	1	1	3	0	5	3	7	5								
P_4	0	3	1	1	7	4	4	3								

Already done

Already done

Already done

Already done

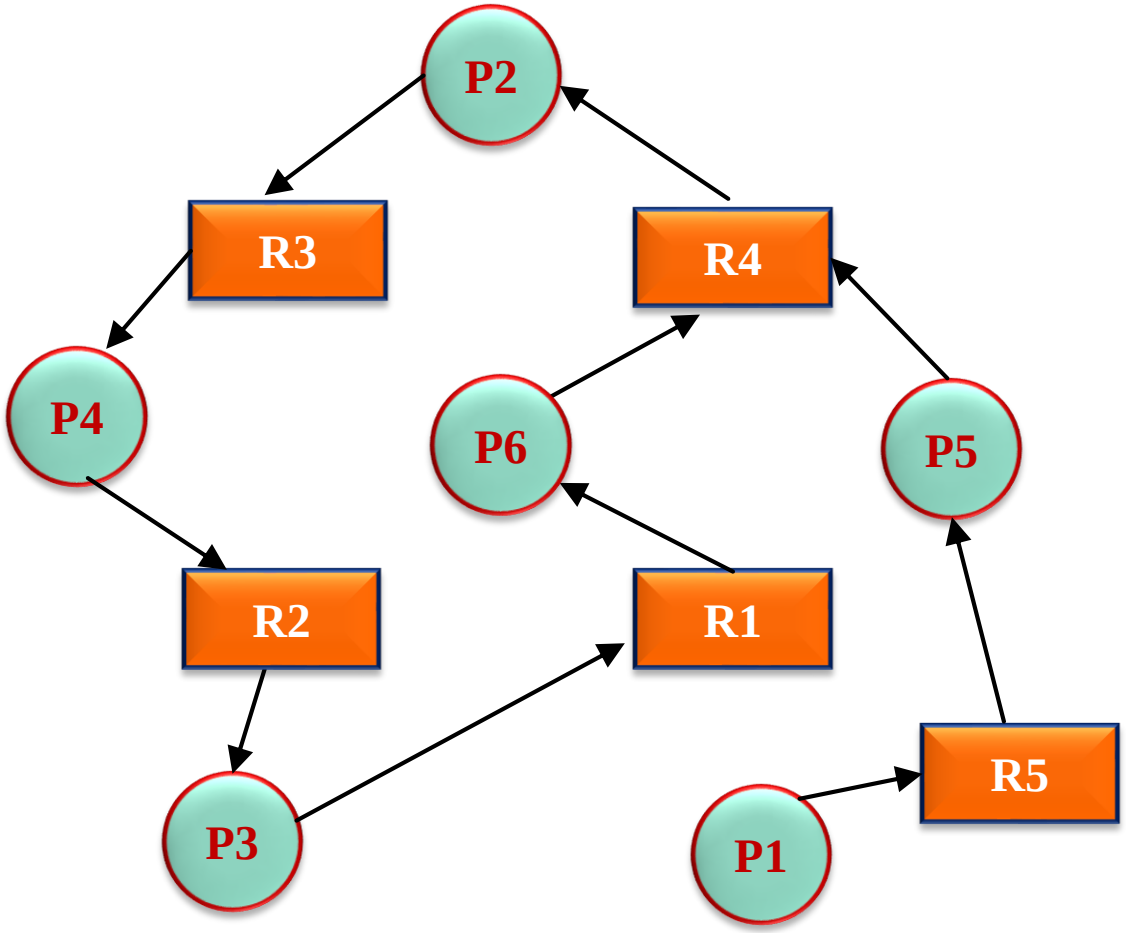
Already done

8 9 10 8

One of the safe sequences

$<P_1, P_3, P_2, P_0, P_4>$

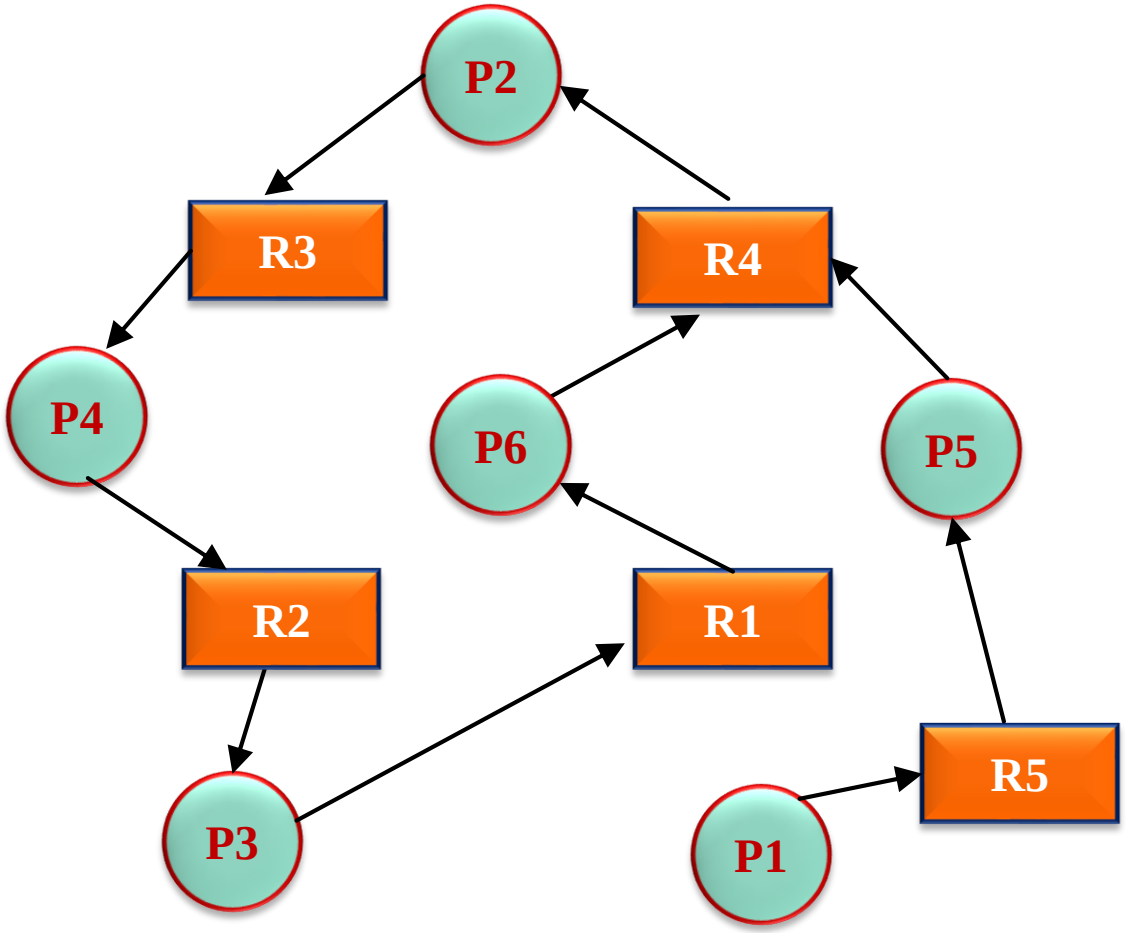
EXAMPLE 5: DETECT IF THERE IS A DEADLOCK



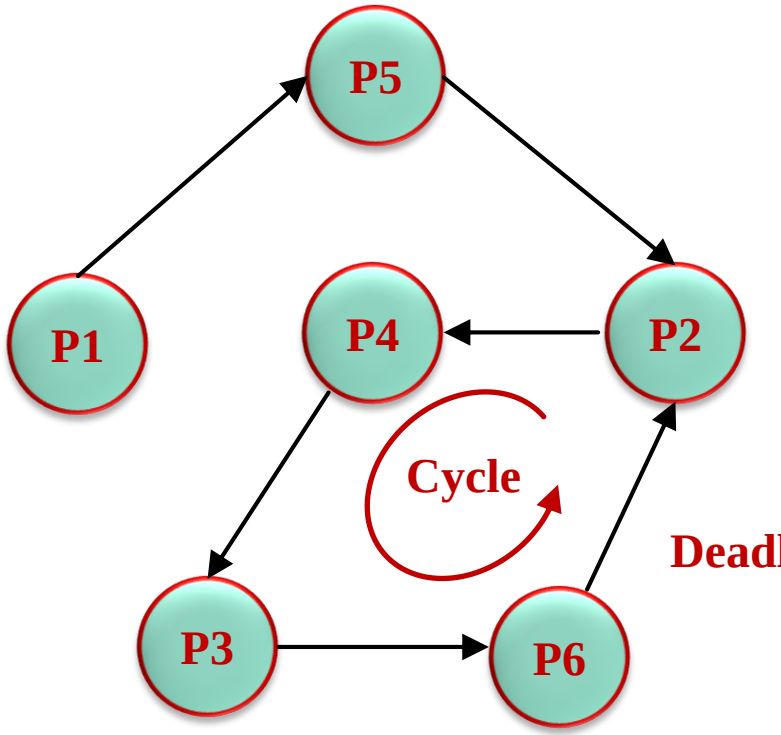
Resource-Allocation Graph (RAG)

EXAMPLE 5: DETECT IF THERE IS A DEADLOCK

We can generate Wait-for-Graph from the RAG



Resource-Allocation Graph (RAG)

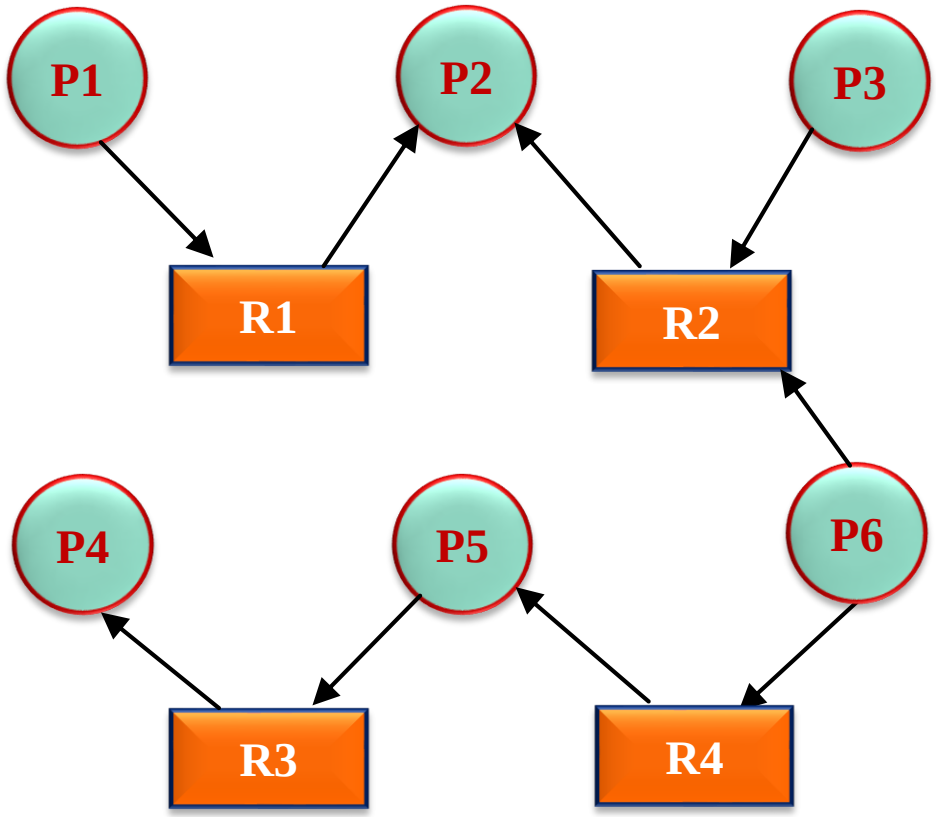


Deadlock is there!

Wait-for-Graph

EXAMPLE 6: DETECT IF THERE IS A DEADLOCK

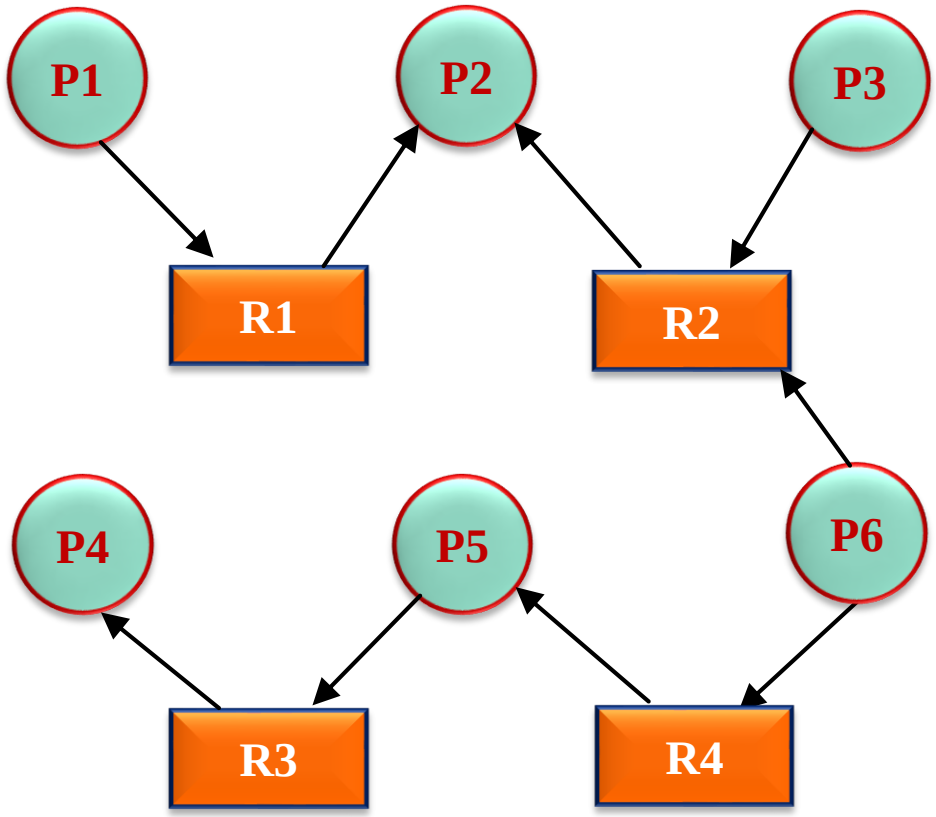
We can generate Wait-for-Graph from the RAG



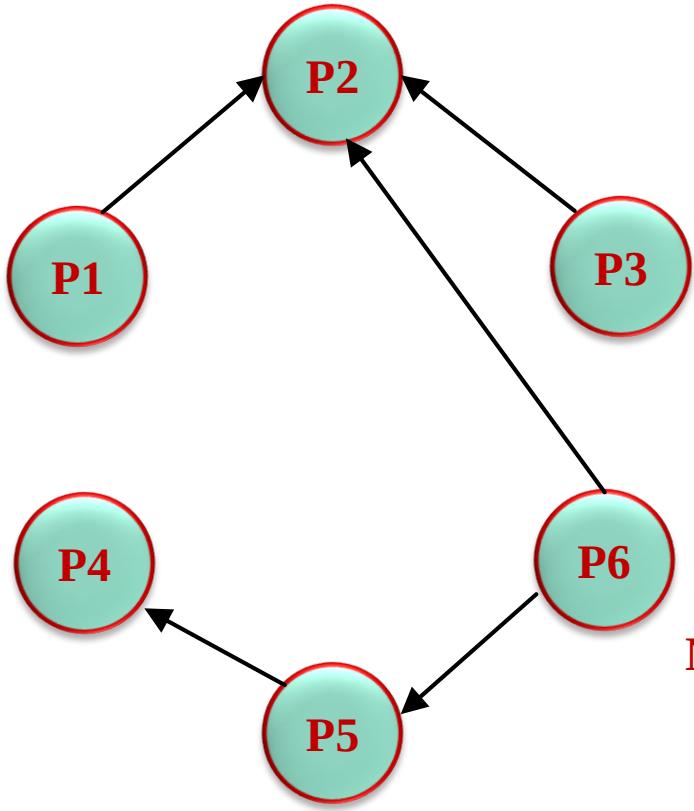
Resource-Allocation Graph (RAG)

EXAMPLE 6: DETECT IF THERE IS A DEADLOCK

We can generate Wait-for-Graph from the RAG



Resource-Allocation Graph (RAG)



Wait-for-Graph

No Deadlock is there!

EXAMPLE 7: DETECT IF THERE IS A DEADLOCK

- Need to detect if there is a deadlock when we have multiple instances of a resource
- A: 7 Instances
- B: 8 Instances
- C: 9 Instances

Process	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P0	1	3	1	5	5	7	0	2	1
P1	2	1	2	0	2	1			
P2	3	1	4	4	6	3			
P3	1	1	1	4	5	6			

EXAMPLE 7: DETECT IF THERE IS A DEADLOCK

- Need to detect if there is a deadlock when we have multiple instances of a resource
- A: 7 Instances
- B: 8 Instances
- C: 9 Instances

Process	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P0	1	3	1	5	5	7	0	2	1
P1	2	1	2	0	2	1			
P2	3	1	4	4	6	3			
P3	1	1	1	4	5	6			

P1 can execute with the available resources

<P1>

EXAMPLE 7: DETECT IF THERE IS A DEADLOCK

- Need to detect if there is a deadlock when we have multiple instances of a resource
- A: 7 Instances
- B: 8 Instances
- C: 9 Instances

Process	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P0	1	3	1	5	5	7	0	2	1
P1	2	1	2						
P2	3	1	4	4	6	3			
P3	1	1	1	4	5	6			

Let P1 is executed

<P1>

EXAMPLE 7: DETECT IF THERE IS A DEADLOCK

- Need to detect if there is a deadlock when we have multiple instances of a resource
- A: 7 Instances
- B: 8 Instances
- C: 9 Instances

Process	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P0	1	3	1	5	5	7			
P1	2	1	2				2	3	3
P2	3	1	4	4	6	3			
P3	1	1	1	4	5	6			

With the available resources, none of the processes can be executed – Deadlock occurs

<P1>

EXAMPLE 8: DETECT IF THERE IS A DEADLOCK

- Total types of resources:
  - Printer (P): 6 instances
  - Scanner (S): 9 instances
  - File (F): 12 instances
  - Keyboard (K): 5 instances
- Process: 5 (P_0-P_4)

Process	Allocation				Request				Available			
	P	**S**	**F**	**K**	**P**	**S**	**F**	**K**	**P**	**S**	**F**	**K**
P_0	2	0	2	0	3	3	4	5	0	3	6	1
P_1	3	0	1	1	0	2	4	1				
P_2	0	2	0	2	6	4	2	3				
P_3	1	1	3	0	5	3	8	6				
P_4	0	3	1	1	3	3	7	1				

EXAMPLE 8: DETECT IF THERE IS A DEADLOCK

- Total types of resources:
  - Printer (P): 6 instances
  - Scanner (S): 9 instances
  - File (F): 13 instances
  - Keyboard (K): 5 instances
- Process: 5 (P_0-P_4)

Process	Allocation				Request				Available			
	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	3	3	4	5	0	3	6	1
P_1	3	0	1	1	0	2	4	1				
P_2	0	2	0	2	6	4	2	3				
P_3	1	1	3	0	5	3	8	6				
P_4	0	3	1	1	3	3	7	1				

EXAMPLE 8: DETECT IF THERE IS A DEADLOCK

- Total types of resources:
  - Printer (P): 6 instances
  - Scanner (S): 9 instances
  - File (F): 13 instances
  - Keyboard (K): 5 instances
- Process: 5 (P_0-P_4)

Process	Allocation				Request				Available			
	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	3	3	4	5	0	3	6	1
P_1	3	0	1	1	0	2	4	1				
P_2	0	2	0	2	6	4	2	3				
P_3	1	1	3	0	5	3	8	6				
P_4	0	3	1	1	3	3	7	1				

With the available resources, P_1 can execute

EXAMPLE 8: DETECT IF THERE IS A DEADLOCK

- Total types of resources:
  - Printer (P): 6 instances
  - Scanner (S): 9 instances
  - File (F): 13 instances
  - Keyboard (K): 5 instances
- Process: 5 (P_0-P_4)

Process	Allocation				Request				Available			
	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	3	3	4	5				
P_1	3	0	1	1					3	3	7	2
P_2	0	2	0	2	6	4	2	3				
P_3	1	1	3	0	5	3	8	6				
P_4	0	3	1	1	3	3	7	1				

Let P_1 is executed

<P_1>

EXAMPLE 8: DETECT IF THERE IS A DEADLOCK

- Total types of resources:
  - Printer (P): 6 instances
  - Scanner (S): 9 instances
  - File (F): 13 instances
  - Keyboard (K): 5 instances
- Process: 5 (P_0-P_4)

Process	Allocation				Request				Available			
	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	3	3	4	5				
P_1	3	0	1	1					3	3	7	2
P_2	0	2	0	2	6	4	2	3				
P_3	1	1	3	0	5	3	8	6				
P_4	0	3	1	1	3	3	7	1				

Let P_1 is executed

With the available resources, P_4 can execute

$<P_1>$

EXAMPLE 8: DETECT IF THERE IS A DEADLOCK

- Total types of resources:
  - Printer (P): 6 instances
  - Scanner (S): 9 instances
  - File (F): 13 instances
  - Keyboard (K): 5 instances
- Process: 5 (P_0-P_4)

Process	Allocation				Request				Available			
	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	3	3	4	5				
P_1	3	0	1	1								
P_2	0	2	0	2	6	4	2	3	3	6	8	3
P_3	1	1	3	0	5	3	8	6				
P_4	0	3	1	1								

Let P1 is executed

Let P_4 is executed

$<P_1, P_4>$

EXAMPLE 8: DETECT IF THERE IS A DEADLOCK

- Total types of resources:
  - Printer (P): 6 instances
  - Scanner (S): 9 instances
  - File (F): 13 instances
  - Keyboard (K): 5 instances
- Process: 5 (P_0-P_4)

Process	Allocation				Request				Available			
	P	S	F	K	P	S	F	K	P	S	F	K
P_0	2	0	2	0	3	3	4	5				
P_1	3	0	1	1								
P_2	0	2	0	2	6	4	2	3	3	6	8	3
P_3	1	1	3	0	5	3	8	6				
P_4	0	3	1	1								

Let P1 is executed

Let P_4 is executed

<P_1, P_4>

With the available resources, none of the processes can be served. So, there is a deadlock

THANK YOU!