CS & IT ENGINEERING

Operating Systems

Deadlock

Lecture No. 2

By- Dr. Khaleel Khan Sir

- Deadlock Avoidance
- Deadlock Detection
- Deadlock Recovery

Cycle $\xrightarrow{N}$ Deadlock;

1. If the R.A.G, has resources of only Multi-Instance Type, then Cycle is only a Necessary Condition for deadlock;
2. If the R.A.G has resources of both Single & Multi-Instance, then the presence of cycle is only a necessary Condition for Deadlock.

Cycle $\xrightarrow{N}$ Deadlock

3. If the R.A.G has resources of only Single Instance Type, then cycle is a necessary and Sufficient condition for deadlock;

Cycle $\xrightarrow[S]{N}$ Deadlock

Deadlock prevention

- To design a system in such a way that the possibility of a deadlock is excluded a priori.
- Prevention philosophy: We know what the preconditions are; So prevent one or more these from occurring.
- For example: Circular wait can be prevented by linear ordering of the resource types. If a process holds resources of type Rj, then it can request resources of type Rk, k > j, but not Ri where i <= j. Similarly, any other process holding Ri can request Rj but a process holding Rj cannot request Ri.

III. Deadlock Avoidance

1) Single Instance Resource

<Resource-Allocation Graph Algorithm>

2) Multi-Instance of Resource (S.I + M.I)

Banker's Algorithm

(i) Safety Algo

(ii) Resource-Request Algo.

Note: Both the Algorithms are based on A-priori Knowledge/Information

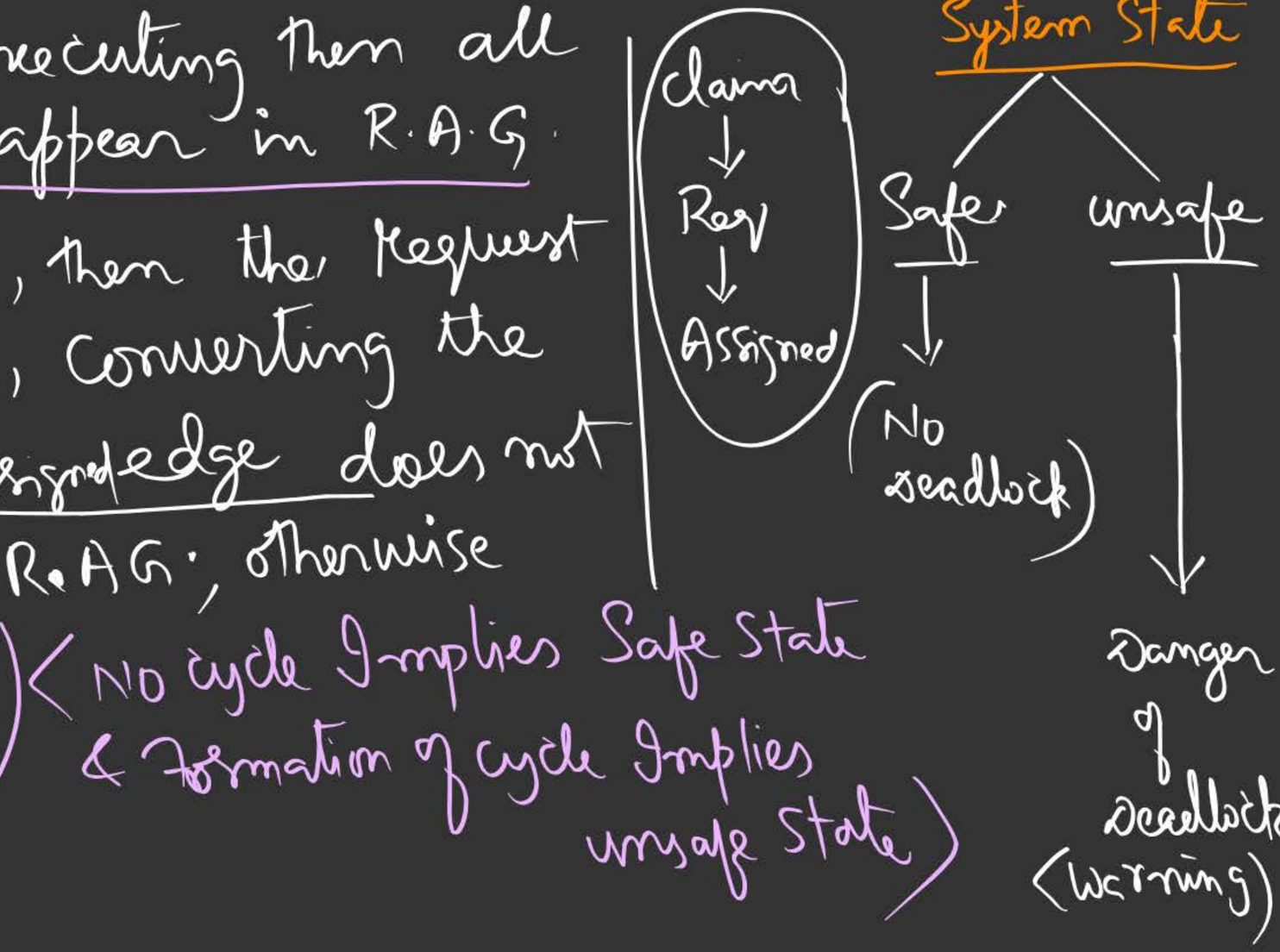
1. Resource-Allocation Graph Algorithm (Single Instance Resource)

→ Resources are claimed a-priori in the System.

→ If Process P_i starts executing then all claim edges must appear in R.A.G.

→ If P_i req's Res. R_j, then the request is granted only if, converting the request edge to assigned edge does not lead to a cycle in R.A.G.; otherwise Process is Blocked.

(No cycle Implies Safe State & Formation of cycle Implies unsafe State)

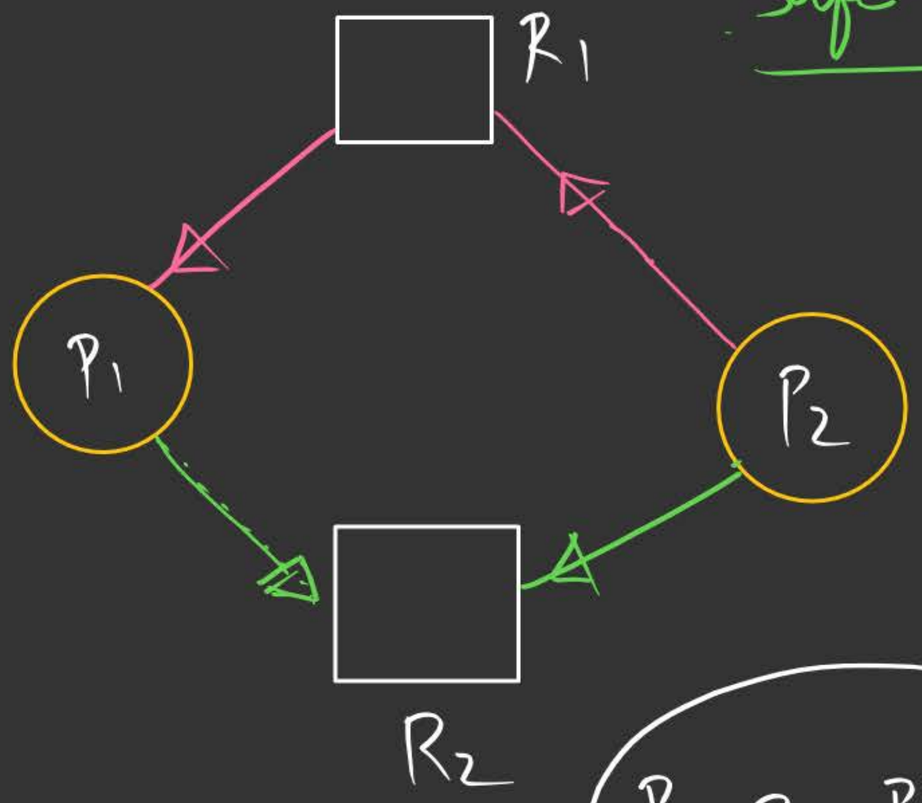


→ The basic objective of R.A.G Algo (S.I)
is to always operate the System in SAFE State;

⇒ System is said to be Safe, iff the
Conversion of Req edge to Assigned Edge does not
Lead to cycle in RAG;
otherwise if it is Leading cycle then
it is unsafe "

Safe → unsafe → deadlock

→ Current State ?



R.A.G

What if P_2 also request R_2 ?

↓

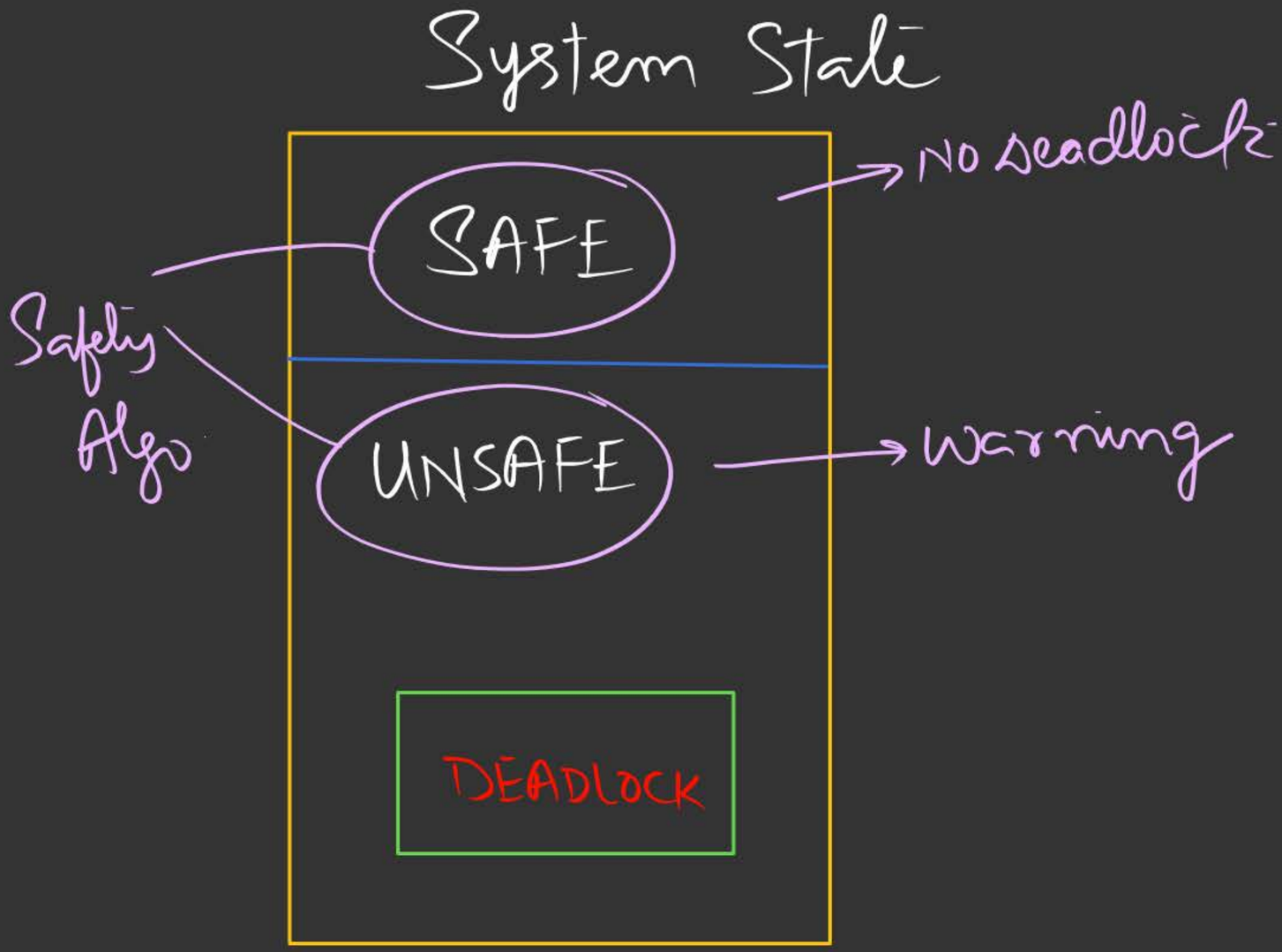
Can it be granted

$P_1 - R_2 - P_2 - R_1 - P_1$

Cycle is detected

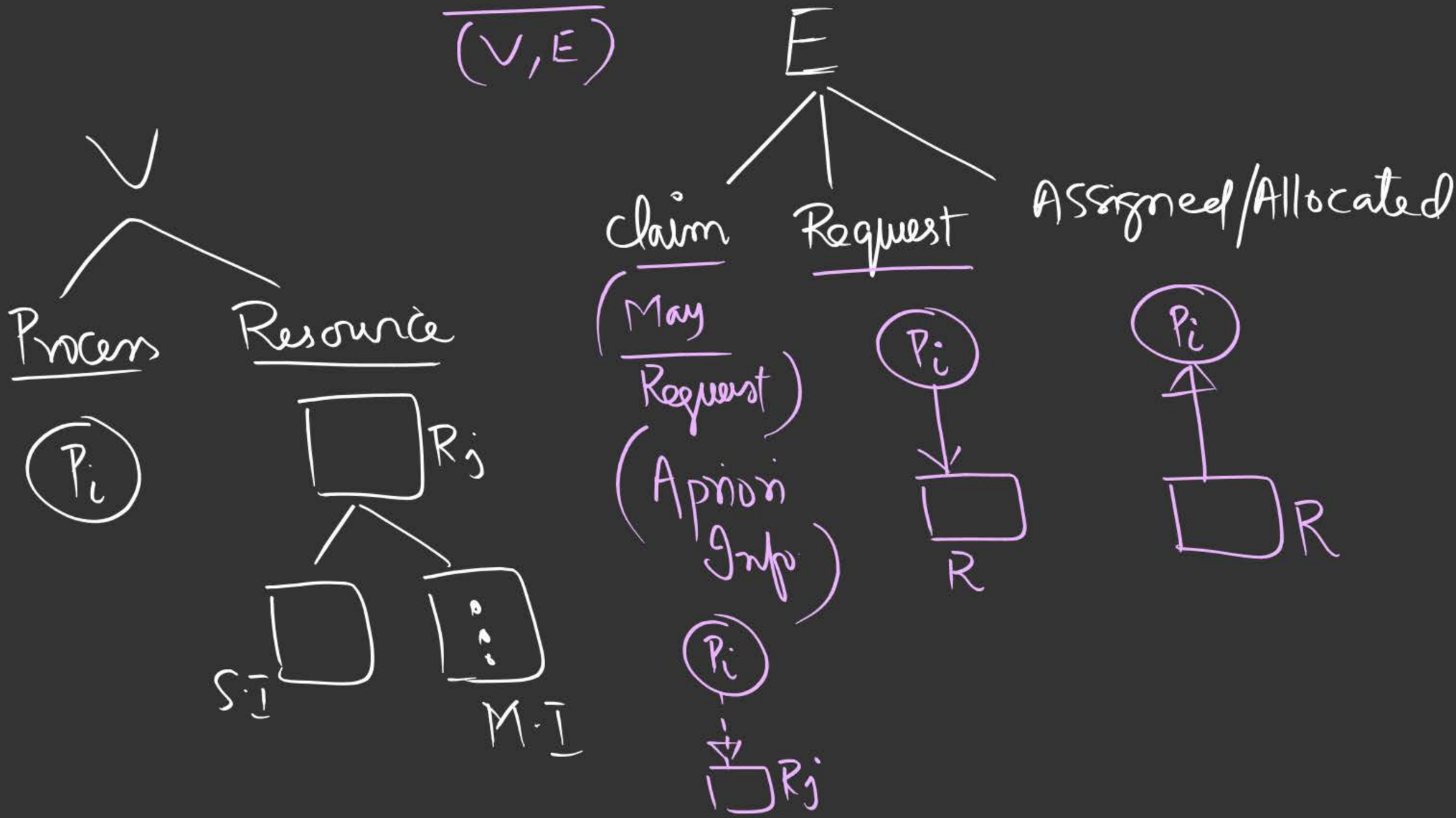
(Implies System is unsafe)

(claim edge is also part of graph)


System State
SAFE
No deadlock
Safety Algo
UNSAFE
Warning
DEADLOCK

R.A.G

(V, E)


V
Process
Resource
Pi
Rj
S.I
M.I
E
Claim
Request
Assigned/Allocated
(May Request)
(Apriori Info)
Pi
Rj
Pi
R
Pi
R

** Bankers's Algorithm (m. Instance) Dijkstra's (Avoidance)

(i) Safety Algo (ii) Resource Request Algo. (Apriori info)

Data structures (Sys. State)

1) n : no. of Processes ($P_1 \ldots P_n$)
2) m : no. of Resources ($R_1 \ldots R_m$)
3) * Maximum $[1 \ldots n, 1 \ldots m]_{n \times m}$:

(claim) $Max[i,j] = K$

$P_i \xrightarrow[max]{} K(R_j)$

Process P_i demands (apriori) 'K' copies of R_j;

4) Allocation $[1 \ldots n, 1 \ldots m]_{n \times m}$

$Alloc[i,j] = a$ $[a \le K]$

$P_i \xleftarrow[alloc]{} a(R_j)$

5) Need $[1 \ldots n, 1 \ldots m]_{n \times m} \Rightarrow$ (Max - Alloc)

$Need[i,j] = b$ $[b = K - a]$

$P_i \xrightarrow[Need]{} b(R_j)$ $[c \le b]$ ✓

6) Request $[1 \ldots n, 1 \ldots m]_{n \times m}$; $Req[i,j] = c$

$P_i \xrightarrow[req]{} c(R_j)$ @time 't'

$Req \le Need$

7) Total $[1 \ldots m]$

$Total[j] = 'z'$

There are 'z' Copies of R_j in the System

8) Availaible $[1 \ldots m]$

$Avail[j] = 'e'$ $(e \le z)$

There are $e(R_j)$ free availaible @ time 't'

$$Avail = Total - \sum_{i=1}^{n} Alloc_i$$

Ex 1)

$n = 5$; $m = 1$; $R = \underline{\text{Total}} = 22$

P.id	Max R	Alloc R	Need R	Avail R
P_1	10	5	5	3
P_2	8	4	4	
P_3	12	5	7	
P_4	5	2	3	
P_5	6	3	3	
		19		

Need = Max - Alloc

t_1:

9:00 am Tom 18/2

<5>
<8>
<13>
<17>
<22>

$\langle P_4; P_5; P_1; P_2; P_3 \rangle$

(Safe-Sequence)

(⇒ we can have Multiple Safe Sequences)

Safety Algorithm :

System is said to be "SAFE" iff the Need of all Processes can be satisfied with the available Resources in some order ; "otherwise it is unsafe"


THANK YOU
GW
SOLDIERS !