The University for world-class professionals

Manchester Metropolitan University

Deadlock Avoidance

Emma Norling
John Dalton room E128
Email: E.Norling@mmu.ac.uk
Telephone: 0161 247 3884

Aims

At the end of this podcast, you should be able to:

- Explain the principle of deadlock avoidance
- Describe the concepts of the *banker's algorithm* used for deadlock avoidance

Deadlock Avoidance

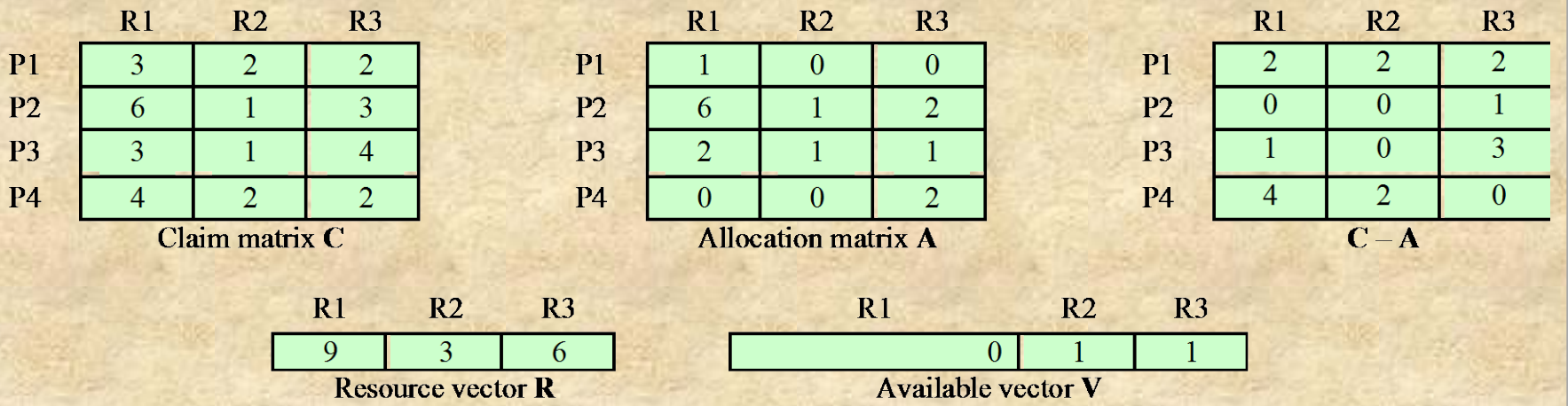
- A decision is made dynamically whether the current resource allocation request will, if granted, potentially lead to a deadlock
- Requires knowledge of future process requests

Two Approaches

- Resource allocation denial
- Process initiation denial

Resource Allocation Denial

- Referred to as the *banker's algorithm*
- ***State*** of the system reflects the current allocation of resources to processes
- ***Safe state*** is one in which there is at least one sequence of resource allocations to processes that does not result in a deadlock
- ***Unsafe state*** is a state that is not safe



(a) Initial state

Figure 6.7 Determination of a Safe State

	R1	R2	R3
P1	3	2	2
P2	0	0	0
P3	3	1	4
P4	4	2	2

Claim matrix **C**

	R1	R2	R3
P1	1	0	0
P2	0	0	0
P3	2	1	1
P4	0	0	2

Allocation matrix **A**

	R1	R2	R3
P1	2	2	2
P2	0	0	0
P3	1	0	3
P4	4	2	0

C – A

R1	R2	R3
9	3	6

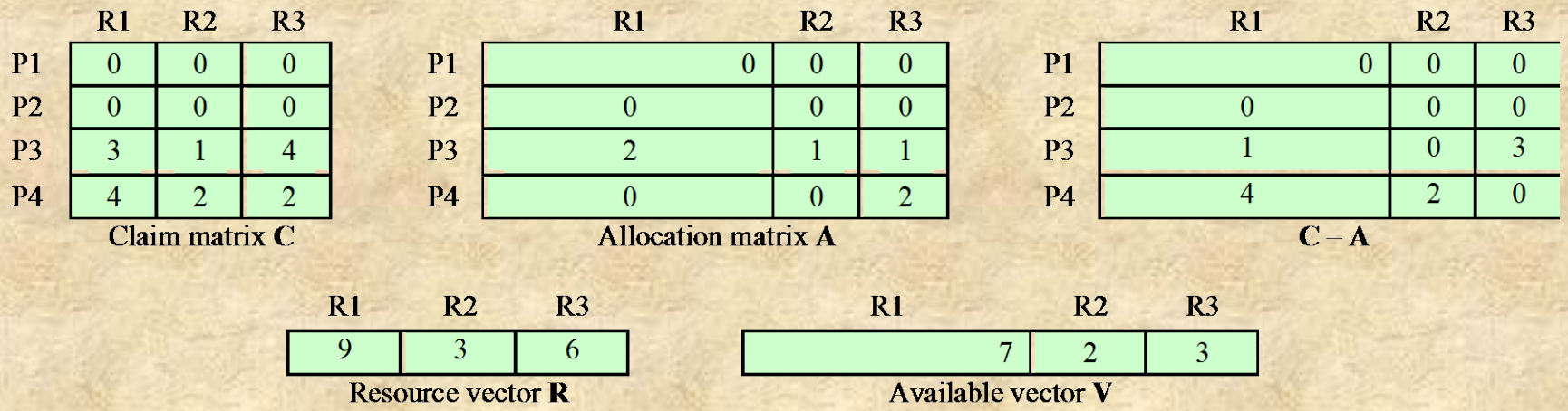
Resource vector **R**

R1	R2	R3
6	2	3

Available vector **V**

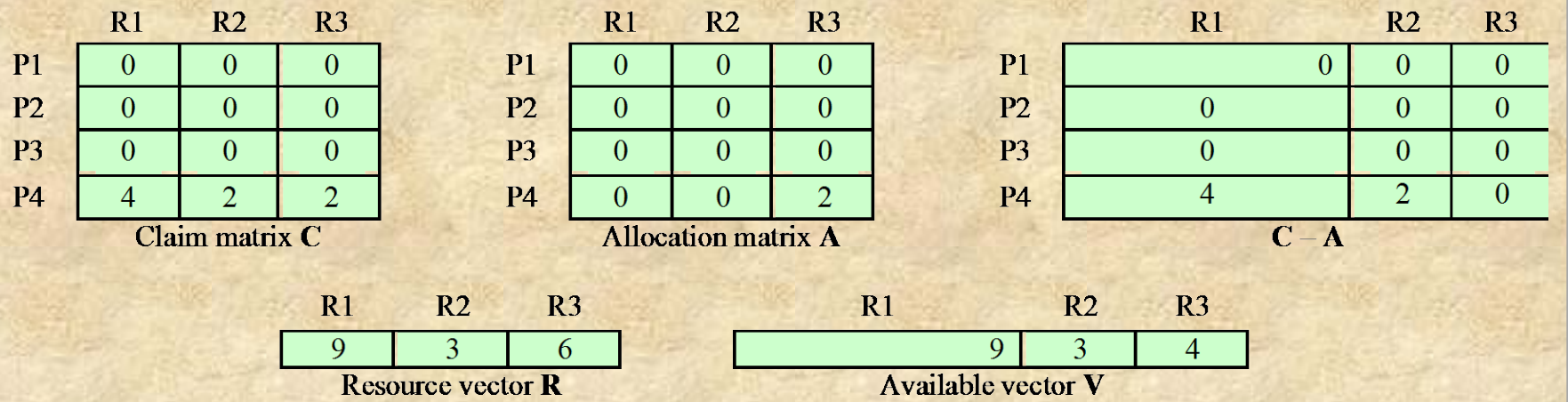
(b) P2 runs to completion

Figure 6.7 Determination of a Safe State



(c) P1 runs to completion

Figure 6.7 Determination of a Safe State



(d) P3 runs to completion

Figure 6.7 Determination of a Safe State

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

Claim matrix **C**

	R1	R2	R3
P1	1	0	0
P2	5	1	1
P3	2	1	1
P4	0	0	2

Allocation matrix **A**

	R1	R2	R3
P1	2	2	2
P2	1	0	2
P3	1	0	3
P4	4	2	0

C – A

R1	R2	R3
9	3	6

Resource vector **R**

R1	R2	R3
1	1	2

Available vector **V**

(a) Initial state

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

Claim matrix **C**

	R1	R2	R3
P1	2	0	1
P2	5	1	1
P3	2	1	1
P4	0	0	2

Allocation matrix **A**

	R1	R2	R3
P1	1	2	1
P2	1	0	2
P3	1	0	3
P4	4	2	0

C – A

R1	R2	R3
9	3	6

Resource vector **R**

R1	R2	R3
0	1	1

Available vector **V**

(b) P1 requests one unit each of R1 and R3

Figure 6.8 Determination of an Unsafe State

Banker's Algorithm

- Banker's algorithm is presented in the textbook
  - You need to know the principles (as illustrated by the previous example); you don't need to implement the algorithm
- Advantages of the banker's algorithm:
  - It is not necessary to preempt and rollback processes, as in deadlock detection
  - It is less restrictive than deadlock prevention

Disadvantages

- Maximum resource requirement for each process must be stated in advance
- Processes under consideration must be independent and with no synchronization requirements
- There must be a fixed number of resources to allocate

Summary

- Deadlock avoidance provides a slightly less conservative strategy than deadlock prevention
- Is only applicable for some problems
- Both under commit resources
  - If deadlocks are uncommon, can lead to significant degradation of performance with little gain.

What next?

- Read the relevant section of the textbook (6.3)
- Move on to the next podcast: deadlock detection and recovery.