Deadlock does not occur as at least one philosopher can get both chopsticks

Q1a) False. One of the conditions that must hold simultaneously with other three conditions is circular wait for deadlock to possibly occur. With only 4 philosophers being hungry simultaneously, circular wait is not fulfilled.

SC2005 Operating Systems

Hold and wait condition cannot be satisfied.

b) True. A deadlock is a set of blocked processes each holding a resource and waiting to acquire a resource held by another process in the set.

TUTORIAL SIX

c) False. If a resource allocation graph has a cycle, deadlock only occurs if only one instance per resource type. If several instances per resource type, deadlock is possible.

Deadlocks

1. Indicate whether the following statements are true or false. Justify your answers.

   (a) In the dining-philosopher problem with five philosophers, if we allow at most four of them to be hungry simultaneously, deadlock may still occur.

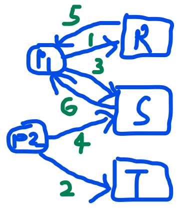
   (b) It is impossible to have a deadlock involving only one single process.

   (c) If a resource allocation graph contains a cycle, then a deadlock has occurred.

2. Use resource-allocation graphs to model the following situations, and determine if deadlock occurs in each case. There are three resource types, R, S, and T, each having a single instance.

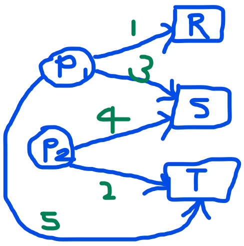
Case 1	Case 2
P1 requests R	P1 requests R
P2 requests T	P2 requests T
P1 requests S	P1 requests S
P2 requests S	P2 requests S
P1 releases R	P1 requests T
P1 releases S	

Case 1:



no deadlock as P2 waiting for S only

Case 2:



deadlock as P1 waiting for T while holding S and P2 waiting for S while holding T

3. A resource-allocation state is given below. Assume Available = 2. −1

	Allocation	Max	Need
PROCESS 1	1	6	5
PROCESS 2	1	5	4
PROCESS 3	2 +1	4	2 −1
PROCESS 4	4 +1	7	3 −1

3a) unsafe since Need(i) > Available for all i

b)safe since sequence <Process 3, Process 2m Process 1,Process 4> satisfies safety requirement

   (a) If PROCESS 4 requests for one more unit of the resource, does this lead to a safe state or an unsafe one?

   (b) If PROCESS 3 requests for one more unit of the resource, does this lead to a safe state or an unsafe one?

4. Consider the following snapshot of a system's state, with four processes (P0, P1, P2 and P3) and three resource types (A, B and C). The current Allocation, Need and Available matrices are shown in the below table. Compute the minimum value for x, so that this system state is safe. Justify your answer.

Work = Available +Allocation

Process	Allocation A B C	Need A B C	Available A B C
			0 1 x
P0	2 1 1	0 1 0	
P1	1 1 0	2 1 2	
P2	1 1 1	2 0 1	
P3	1 1 1	4 1 0	

4

Q3a)
Work = Allocation + Available
need = 2 = available(work), P3 done 2 + 2 =4
need = 4 = work, P2 done, 4+1 = 5
need =4 <work=5, P4 done, 5+4 = 9
need =5 < work=9,P1 done, 9+1 = 10
Thus, safe state with sequence<P3,P2,P4,P1>

b)
need = 3 > available = 2, P3 not done
need = 5 > available = 2, P1 not done
need = 4 > available = 2, P2 not done
need = 3 > available = 2, P4 not done
Thus, unsafe state

P0,P1 and P2 have to execute before P3 as Available A is 0 and P1,P2,P3 together allocate just enough A that P3 needs.

With x =0, P0 is done as need =available, increasing work to 2,2,1 for A,B,C respectively. before P1 ~~P1 and~~ P2 can then finish as work is more than each of the process needs. P3 can then execute as work is now 4,4,2 for A,B,C more than P3 needs of 4,1,0 for A,B,C