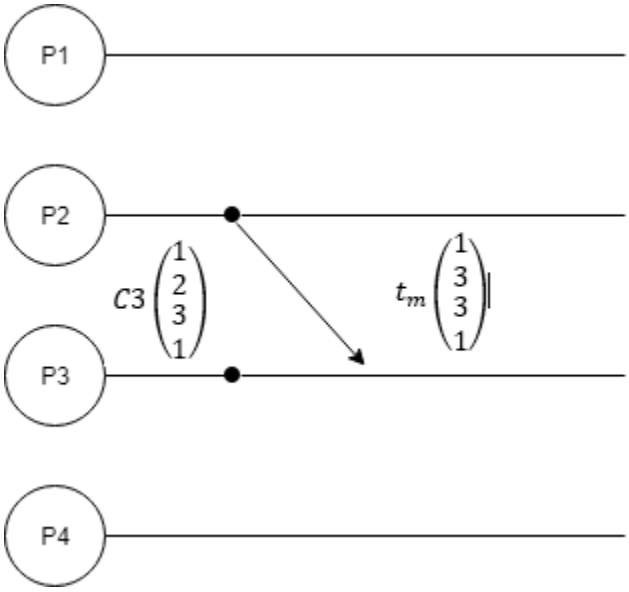
Taylor Pedretti

005488635

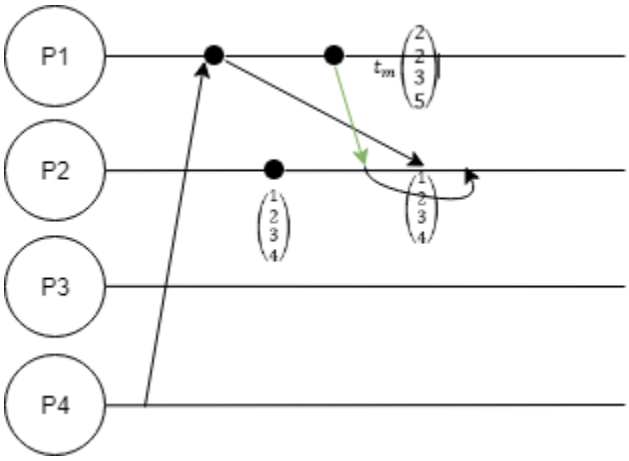
Homework 4

1. Suppose the “Birman-Schiper-Stephenson Protocol” is used to enforce “Causal Ordering of Messages” of a system that has four processes, P_1; P_2; P_3, and P_4. With the help of diagrams, explain clearly what the process would do in each of the following cases.

a) Answer: It will be delivered because P_i will receive C3[i] before P3 receives a new message from P2.



b) Answer: We start with C2[1]=tm[2]-1, but C2[4]<tm[4], then P2 should buffer the message until it receives a previous message from P1.



2. Consider a cut C:

$$C = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Where c_1, c_2, and c_3 are the cut events with vector clocks C_1, C_2, C_3 respectively:

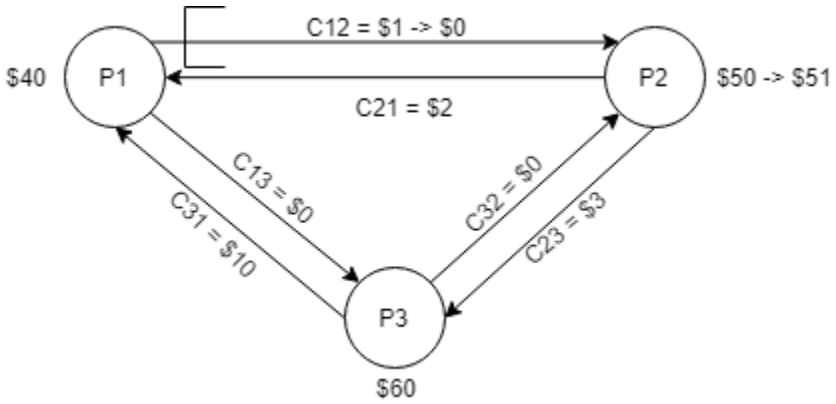
$$C_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \; C_2 = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} \; C_3 = \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix}$$

Calculate $T_C = \sup(C_1, C_2C_3)$. Is C consistent cut? Why?

Answer: C is an inconsistent cut because: $T_C > \begin{pmatrix} C1[1] \\ C2[2] \\ C3[3] \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}$

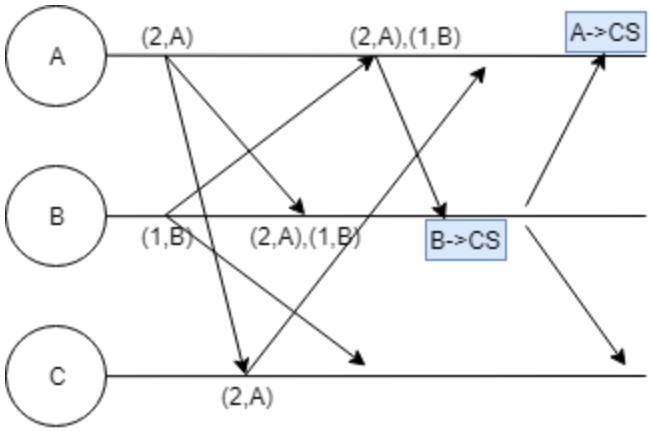
3. Answer:

State	Money
LS_1	$40
LS_2	$51
LS_3	$60
$C1_2$	$0
$C1_3$	$0
$C2_1$	$2
$C2_3$	$3
$C3_1$	$10
$C3_2$	$0



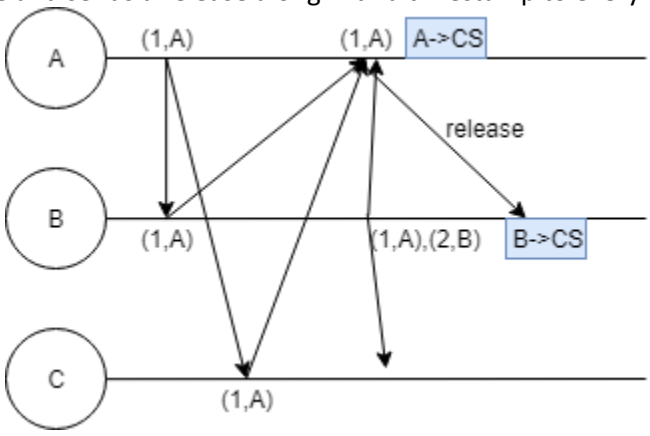
4. In Lamport's algorithm for mutual exclusion, Process P_i enters CS when the following 2 conditions are satisfied:
1) P_i's request is at the head of $requestiqueue_i$
2) P_i has received a (REPLY) message from every other process time-stamped later than t_{si}
Condition 1) can hold concurrently at several sites. Why then is 1) needed to guarantee mutual exclusion? Does the algorithm work if condition 2) is removed? Why? Give an example with illustrations (drawings) to support your argument.

Condition 1 Answer: Condition 1 is a guarantee that the mutual exclusion because all the queues that each process must be in sync since each process own queue might not be updated the same as the others. Condition 1 serves as a substitution of shared memory since they don't have a shared memory.
Condition 2 Answer: If condition 2 is removed from the algorithm it will still work under certain conditions, a release message that serves as a reply. A release message from the process that entered the CS can signal that if a process whose process is second in the queue, then that process can center the CS next.



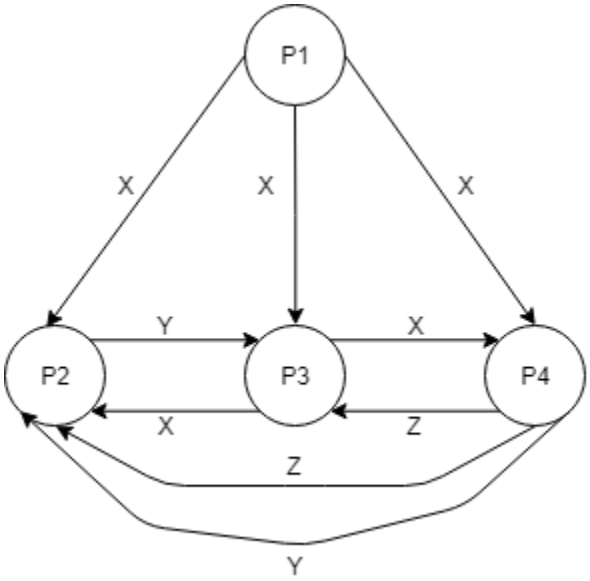
5. In Lamport`s algorithm of mutual exclusion, if a site $S_i$ is executing the critical section, is it necessary that $S_i$`s request need to be always at the top of the request-queue at another site S_j? Explain and give n example (with diagrams) to support your argument.

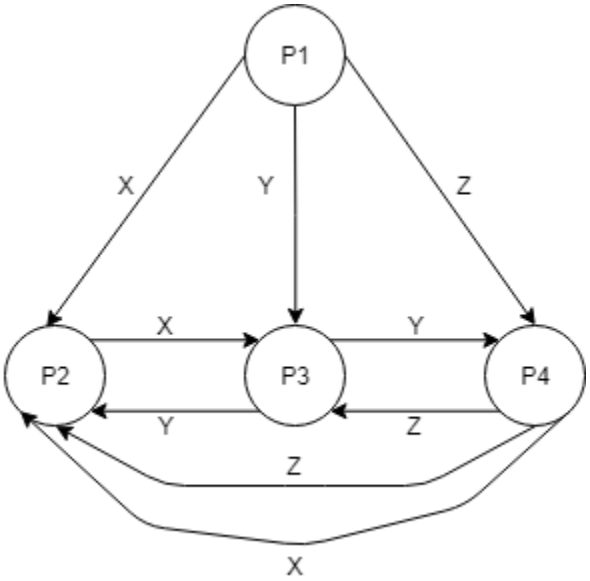
Answer: No, it is not necessary if it is executing in the CS, which means that it has already been to CS. Hence, it is irrelevant if it is on top or not. When exiting the CS, process Pi removes its request from the head of its request queue and sends a release along with a timestamp to every other process.



6. Can Byzantine agreement be always reached among four processors if two processors are faulty? With the help of diagrams, explain your answer.

Answer:





For figure in the left side,	For figure in the right side,
P2's majority = [x,x,z] =x,	P2's majority = [x,y,z] = None
but P2 is a traitor so it will retreat	P3's majority = [y,x,z] = None
P3's majority = [x,y,z] = retreat	P4's majority = [z,x,y] = None
P4's majority = [x,x,y] = x, but	
P4 is a traitor so it will retreat	

7.