Homework 2

CS425/ECE428 Spring 2020

Due: Thursday, Feb 27 at 11:59 p.m.

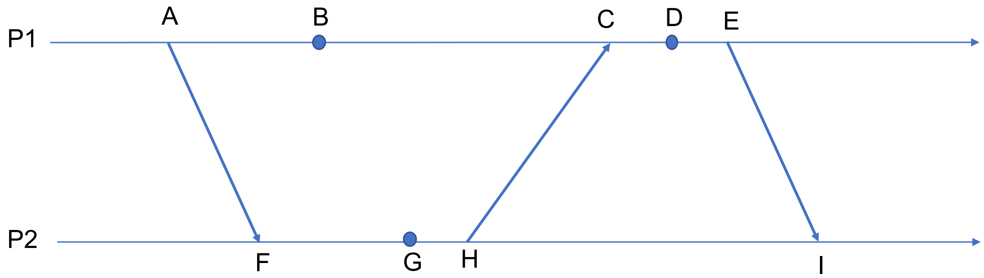

Figure 1: Figure for question 1

1. (a) (6 points) Consider the events {A,B,...I} across two processes as shown in Figure 1. Each event is shown with the *real time* that it occurred. List all possible linearizations for this system that includes each event.

   (b) (2 points) What is the total number of consistent global states that can be possibly captured for the above system?

   (c) (2 points) Provide an example of unstable safety (or non-safety) property. How can it be made stable?

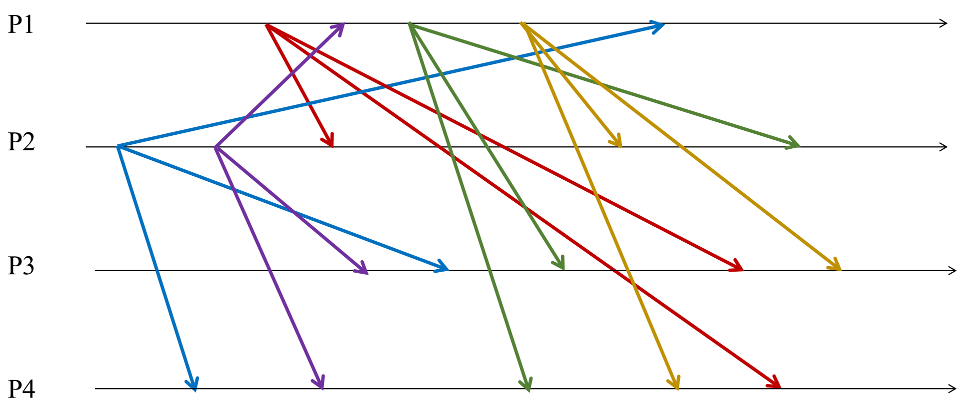

Figure 2: Figure for question 2(a)

2. (a) (5 points) In the execution in Figure 2, processes send messages to each other to implement *FIFO ordered* multicast. To simplify the picture, messages sent by each process to itself are not shown, but you may assume that such messages are received instantaneously. For the questions below, you may use printed or hand-drawn figure with hand-drawn responses, or digitally edit the figure from the homework PDF.

   (i) Identify the messages that are buffered at the processes to ensure FIFO multicast delivery. (Circle the receive event for the buffered messages to identify those messages.) *(2.5 points)*

(ii) For each message buffered as above, determine the earliest instant of time at which the message may be delivered, while ensuring FIFO multicast. (To identify the instant of time draw an arrow that begins at the time when the message is received to the time at which the message may be delivered.) *(2.5 points)*

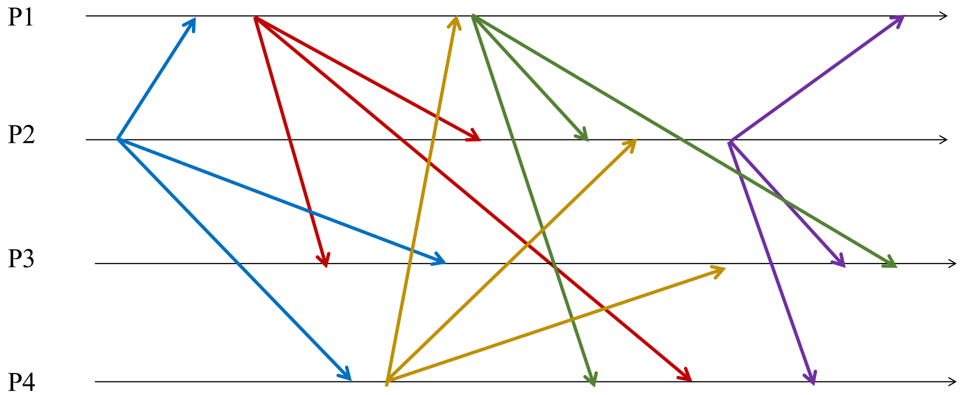

Figure 3: Figure for question 2(b)

(b) (5 points) In the execution in Figure 3, processes send messages to each other to implement *causal multicast*. To simplify the picture, messages sent by each process to itself are not shown, but you may assume that such messages are received instantaneously. For the questions below, you may use printed or hand-drawn figure with hand-drawn responses, or digitally edit the figure from the homework PDF.

(i) Identify the messages that are buffered at the processes to ensure causally-ordered multicast delivery (Circle the receive event for the buffered messages to identify those messages.) *(2.5 points)*

(ii) For each message buffered as above, determine the earliest instant of time at which the message may be delivered, while ensuring causally-ordered multicast. (To identify the instant of time draw an arrow that begins at the time when the message is received to the time at which the message may be delivered.) *(2.5 points)*

3. For each of the statements below, identify whether it is *true* or *false*. If it is false, present a counter-example. If it is true, prove why.

(a) (2 points) A causal multicast is also FIFO ordered.

(b) (2 points) A FIFO + total multicast is also causal.

(c) (2 points) A causal multicast is total ordered.

(d) (2 points) We can implement the ISIS algorithm for total ordering on top of (or using) CO-multicast (causal-ordered multicast), to achieve a total causal multicast.

(e) (2 points) If processes use R-multicast, and each *channel* follows FIFO order, then causal ordering is satisfied.

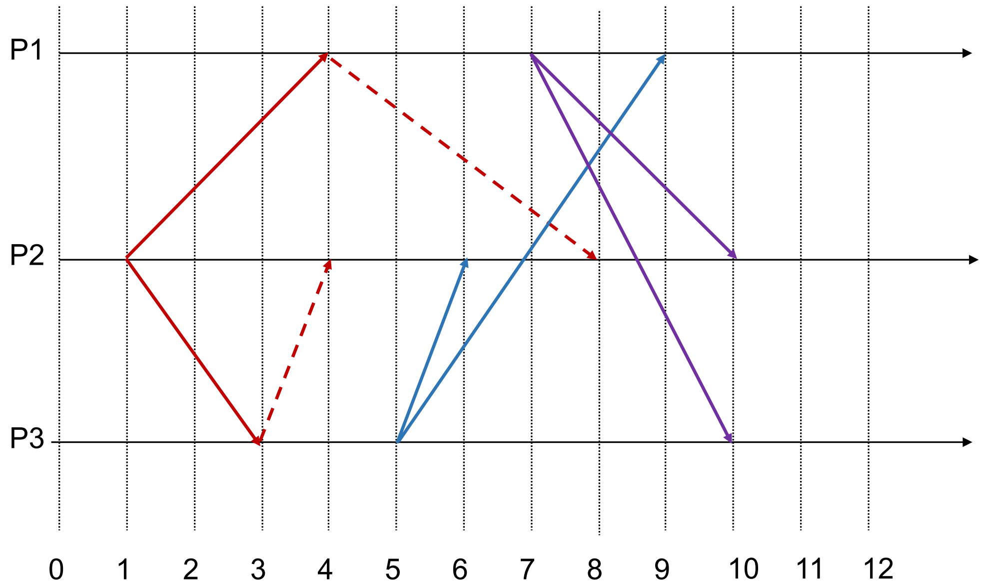

Figure 4: Figure for question 4(a)

4. (a) (8 points) Figure 4 shows three process P1, P2, and P3 implementing the Ricard-Agrawala (RA) algorithm for mutual exclusion. The solid lines indicate requests for accessing the critical section (CS) made by each process – purple, red, and blue requests are from P1, P2, and P3 respectively. Only a subset of replies are shown; specifically, the dashed red arrows indicate P1's and P3's replies to P2's request. Other than the (possibly) missing replies to CS requests, no other messages are exchanged between the processes. Refer to this figure to answer the following questions:

   (i) What is P2's state (as per the RA algorithm) when it receives CS request from P3? How will P2 handle P3's request upon receiving it – will it immediately send back a reply or will it queue the request? Why? [2 points]

   (ii) What is P2's state (as per the RA algorithm) when it receives CS request from P1? How will P2 handle P1's request upon receiving it – will it immediately send back a reply or will it queue the request? Why? [2 points]

   (iii) What is P1's state (as per the RA algorithm) when it receives CS request from P3? How will P1 handle P3's request upon receiving it – will it immediately send back a reply or will it queue the request? Why? [2 points]

   (iv) What is P3's state (as per the RA algorithm) when it receives CS request from P1? How will P3 handle P1's request upon receiving it – will it immediately send back a reply or will it queue the request? Why? [2 points]

   (b) (2 points) Four processes, P1, P2, P3, and P4, participate in Maekawa's algorithm for mutual exclusion. Which of the following voting sets (Vi for process Pi) will satisfy the safety condition for mutual exclusion? Why or why not?

   - Option 1: V1 = [P1, P2], V2 = [P2, P3], V3 = [P3, P4], V4 = [P4, P1]
   - Option 2: V1 = [P1, P2, P3], V2 = [P1, P2, P4], V3 = [P1, P3, P4], V4 = [P2, P3, P4]
   - Option 3: V1 = V2 = V3 = V4 = [P1, P2, P3, P4]