Distributed Systems and Algorithms—Exam 2 Solutions

Question 2

1. For any pair of processes \( p_i \) and \( p_j \), it always holds that \( T_i(j, i) \leq T_j(j, i) \).
   
   To be precise, this question should require that we examine \( T_i \) and \( T_j \) at the same wall clock time.

   **TRUE**

   Row \( j \) in \( T_i \) describes the events that site \( i \) knows that site \( j \) has in its log. \( T_i(j, i) = t \) means that site \( i \) knows that site \( j \) has the first \( t \) events that site \( i \) executed in \( L_j \).

   Row \( j \) in \( T_j \) describes the events that site \( j \) actually has in its log. For \( T_j(j, i) \) to be less than \( T_i(j, i) \), site \( i \) would have to know that some event was in \( L_j \), but site \( j \) is not aware that this event is in \( L_j \). This is impossible.

2. For any pair of processes \( p_i \) and \( p_j \), \( i \neq j \), it always holds that \( T_i(i, i) \leq T_j(j, j) \).
   
   **FALSE**

   Consider a system with two sites, \( p_1 \) and \( p_2 \), and let \( i = 1 \) and \( j = 2 \). Suppose site 1 executes three local events, and site 2 executes one local event, and no messages are sent. Then \( T_i(i, i) = 3 \) and \( T_j(j, j) = 1 \).

Question 3

1. Does Dr. Science’s proposed algorithm variation guarantee safety? Answer YES or NO and give a justification.

   NO. Consider the execution shown below. Here, \( p_1 \) and \( p_4 \) both receive GRANT messages from \( \frac{N}{2} = 2 \) other processes, and they have implicit permission from themselves. Therefore, each process has permission to access the resource from a majority of processes, and they access the resource at the same time. This is a violation of the safety property.

2. Does Dr. Science’s proposed algorithm variation guarantee liveness? Answer YES or NO and give a justification.

   NO. Consider the execution shown below. Here \( p_1 \) access the resource after receiving a GRANT message from \( p_2 \) and \( p_3 \). When it finishes with the resource, it sends a RELEASE message to \( p_2 \) and \( p_3 \). Since it received the GRANT from \( p_4 \) after it was finished with the resource, it does not send a RELEASE message to \( p_4 \). Therefore, if \( p_4 \) makes a request for the resource anytime after it receives \( p_1 \)’s request (the pink rectangle), \( p_1 \)’s request will be at the head of \( p_4 \)’s queue (and will stay there), and \( p_4 \) will never be able to access the resource. This violates the liveness property.
Question 4

1. Explain the steps of Raymond’s algorithm that are executed after site F is finished with the resource to grant the two requests.

An execution that results in the worst case number of messages is as follows:

(a) Site A sends REQUEST to site B and adds its request to its queue

(b) Site B receives REQUEST from A, adds A to its queue, and sends REQUEST to site C

(c) Site C receives REQUEST from B, adds B to its queue, and sends REQUEST to site E

(d) Site E receives REQUEST from C, adds C to its queue, and sends REQUEST to site F

(e) Site F receives REQUEST from E and adds E to its queue

(f) Site D sends REQUEST to site C and adds its request to its queue

(g) Site C receives REQUEST from site D and adds D to its queue. Since it has already sent a REQUEST to site E, it does not send another

(h) Site F finishes with resources, sends TOKEN to site E and removes E from its queue

(i) Site E receives TOKEN, sends TOKEN to site C and removes C from its queue

(j) Site C receives TOKEN, sends TOKEN to site B and removes B from its queue – its queue still contains D so it sends REQUEST to B to ask for token back

(k) Site B receives TOKEN from site C, sends TOKEN to site A and removes A from its queue

(l) Site B receives REQUEST from site C and adds C to its queue, and sends REQUEST to site A

(m) Site A receives TOKEN from site B, removes its own request from its queue, and accesses the resource

(n) Site A receives REQUEST from site B and adds B to its queue

(o) Site A finishes with resources, sends TOKEN to B, and removes B from its queue

(p) Site B receives TOKEN from site A, sends TOKEN to C, and removes C from its queue

(q) Site C receives TOKEN from site B, sends TOKEN to D, and removes D from its queue

(r) Site D receives TOKEN, removes its own request from its queue, and accesses the resource

2. What is the total number of messages that are sent to grant the requests of A and D?

There are 14 messages sent in the above execution.
Question 5 Recall that Maekawa’s original mutual exclusion algorithm is prone to deadlock. Sanders proposed a revision of the algorithm that is deadlock free. In this revision, a process must relinquish its locks only if it receives both an INQUIRE message and a FAIL message. Give an example execution where a process receives an INQUIRE message (but no FAIL message), and there is no deadlock.

Consider a system with 9 processes that uses Maekawa’s grid quorums. An example execution is as follows:

1. \( p_1 \) requests resource, sends \( req(1, 1) \) to \( p_1, p_2, p_3, p_4, \) and \( p_7 \)
2. \( p_1, p_3, \) and \( p_7 \) receive \( req(1, 1) \) and send GRANT to \( p_1 \). The \( req(1, 1) \) messages to \( p_2 \) and \( p_4 \) are in transit
3. \( p_1 \) sends message to \( p_5 \)
4. \( p_1 \) receives GRANT from \( p_1, p_3, \) and \( p_7 \), waiting on GRANT from \( p_2 \) and \( p_4 \)
5. \( p_5 \) receives message from \( p_1 \)
6. \( p_5 \) requests resource, sends \( req(5, 2) \) to \( p_2, p_4, p_5, p_6, \) and \( p_8 \)
7. \( p_2, p_4, p_5, p_6, \) and \( p_8 \) receive \( req(5, 2) \) and send GRANT to \( p_5 \)
8. \( p_5 \) receives GRANT from its entire quorum and starts accessing resource
9. \( p_2 \) and \( p_4 \) receive \( req(1, 1) \). Since \( (1, 1) < (5, 2) \), they each sends INQUIRE to \( p_5 \)
10. When \( p_5 \) receives INQUIRE messages, it will not release its locks, since it has not received a FAIL message (and is, in fact, already accessing the resource)
11. When \( p_5 \) finishes with the resource, it will send RELEASE to its quorum.
12. When \( p_2 \) and \( p_4 \) receive RELEASE from \( p_5 \), they will send GRANT to \( p_1 \)
13. \( p_1 \) now has received all of its GRANT messages and can access the resource