HY486: Principles of Distributed Systems

Professor: Panagiota Fatourou

2nd Set of Exercises: Theory Part

Deadline: January 15, 2021

Graduate students should answer all the exercises in order to get the grade of 100. Undergraduate students do not have to answer the parts that are labelled as [Graduate] although they may want to do so for those exercises that are references as bonus for them.

Exercise 1 [Undergraduates: 25%, Graduates: 18%]

You are given an asynchronous distributed system, represented as an undirected graph, where processes have unique identifiers and communicate via message passing. Design an algorithm that constructs a spanning tree for this system, given a distinguished node p_r. The algorithm returns TRUE if it ends up that all leaves of the constructed tree are in the same level (i.e. they have the same distance from the root). Furthermore, for each node of the constructed spanning tree, the algorithm calculates the number of leaves in each of its sub-trees. At the end of the computation, all processes should know whether the leaves are in the same level and each process should know the number of leaves in its sub-trees.

i. Present event-driven pseudocode for your algorithm and describe its function. [15%, 10%]

ii. What is the communication complexity of your algorithm? Justify your answer. [5%, 4%]

iii. What is the time complexity of your algorithm? Justify your answer. [5%, 4%]

Exercise 2 [Undergraduates: 25%, Graduates: 37%]

You are given an asynchronous distributed system, represented as an undirected graph, where processes have unique identifiers and communicate via message passing.

i. **[Both]** Present event-driven pseudo-code for counting the number of nodes in an Asynchronous System given an Unrooted Spanning Tree of the system (see slide 6 in Section Advanced Graph Algorithms). What is the communication complexity and the time complexity of your algorithm? Justify your answer. [10%, 7%]

ii. **[Both]** Present event-driven pseudo-code that checks whether all nodes at distance k from a distinct node p_r have the same number of neighbors. Assume that $1 \leq k \leq dist$, where dist is the longest distance of p_r from any other node in the graph. What is the communication complexity and the time complexity of your algorithm? Justify your answer. Your algorithm should not assume knowledge of dist. [15%, 15%]

iii. **[Graduates]** Present event-driven pseudo-code that does the following: For **all** k, $1 \leq k \leq dist$, it counts the number of nodes that are in distance k from a distinct node p_r, where dist is the longest distance of p_r from any other node in the graph. When the algorithm terminates, p_r should know dist and for every k, it should also know how many nodes are in distance k from it. [15%]

Exercise 3 [25% + Bonus for both: 15%]

Consider a ring network in an asynchronous system, that is, a collection of n processes, where each process is bi-directionally connected to its neighbors. Assume that each process p_i is distinguished by its unique id, id_i, and that it can distinguish its left from its right neighbor. Assume however that all processes can send messages only

to one direction (i.e. messages in the ring flow only towards the left or towards the right). Without loss of generality, assume that messages in the ring flow towards the right.

A. An algorithm for leader election in such a unidirectional ring simulates two-way communication as follows: Each process can either be in active or relay state, and has a current value v_i, which initially has the value id_i. Initially, each process is in active state. The algorithm works in phases and each process locally keeps track of the phase it is currently executing.

   In the first phase, each active process sends a pair, containing its value and its phase number, two steps to the right, i.e. it sends a message containing this pair to the first two active processes on its right in the ring. Subsequently, p_i compares its value to the values received by its two left active predecessors in the ring. If the first active neighbor on the left of p_i has the highest value among the three, then p_i remains in the active state, but changes its value to the value of that left neighbor. Otherwise, p_i switches to relay state and remains in that state for the rest of the execution of the algorithm.

   Each subsequent phase works in a similar way. Each active process p_i sends again its value v_i to the next and second-next active processors to its right on the ring, and waits to hear from the two immediate active processes on its left. Then, it applies a similar rule as in the first phase to decide whether it will remain active or will become a relay. If it remains active, it updates its value accordingly.

   If at any phase, p_i detects that the value that it has received from its first active left neighbor, is the same as its own value, then p_i determines that it is the only active process left in the ring and thus, it elects itself as the leader.

   i. Present event-driven pseudocode for the aforementioned algorithm. [15%, 15%]

   ii. What is the communication complexity of the algorithm? Justify your answer. [5%, 5%]

   iii. What is the time complexity of your algorithm? Justify your answer. [5%, 5%]

B. **[Bonus (for both graduates and undergraduates): 15%]** Design a version of the algorithm for the case of a bidirectional asynchronous system. In this version of the algorithm, the values of the active processes do not need to be forwarded around the ring. Instead, each process can collect the values of its two active neighbors in each phase. Try to design the most efficient algorithm you can both in terms of the number of messages and the number of rounds.

   i. Present event-driven pseudocode for your algorithm and describe its function. [9%]

   ii. What is the communication complexity of your algorithm? Justify your answer. [3%]

   iii. What is the time complexity of your algorithm? Justify your answer. [3%]

Exercise 4 [Undergraduates: 25%, Graduates: 20%]

A. In Figure 1, you are given an execution in an asynchronous system where processes communicate via message passing. [10%, 8%]

   i. Describe which events of process p_1 are related to events of process p_3 through the happens-before relationship.

   ii. Use the simple logical timestamp algorithm that was studied in class in order to assign logical timestamps to the evens of the execution in Figure 1.

   iii. Use the vector timestamps that was studied in class in order to assign vector timestamps to the events of Figure 1.

B. The Chandy-Lamport Algorithm that you have studied in the lectures assumes that processes communicate over reliable FIFO channels. This means that if a message M_1 is sent before message M_2 by some process p_i to some neighboring process p_j, then p_j must receive M_1 before it receives M_2. [15%, 12%]

i. Use a counterexample to show that the correctness of the algorithm does not hold when the channels are not FIFO.

ii. Propose a modification of the Chandy-Lamport Algorithm that works correctly when the channels are not FIFO. Provide pseudocode for your solution and explain your reasoning.

Note: For answering ii., you are allowed to have each process delaying the processing of some of the messages (until some specific event has occurred) to ensure correctness.

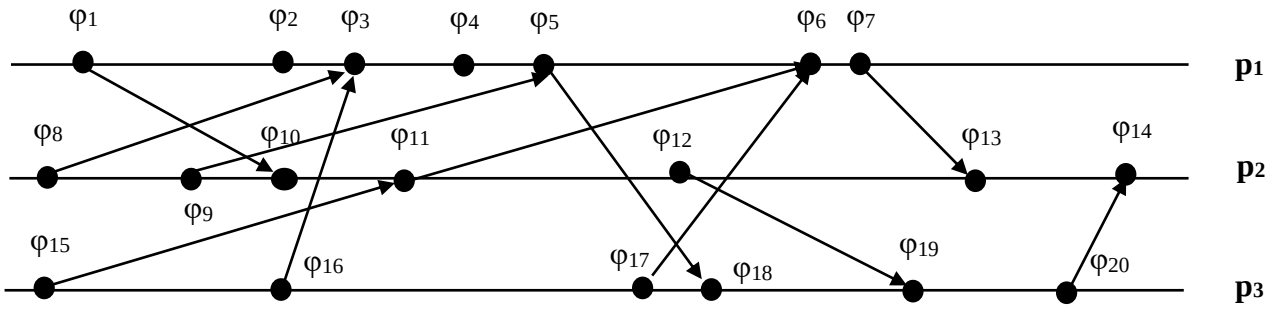

Figure 1