Distributed Systems
29/01/2020
Exam A

Family Name_____Name_____Student ID_____

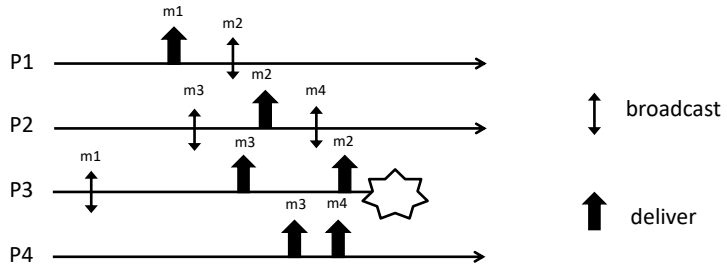
Please, tick the appropriate option:

☐ Master of Science in Engineering in Computer Science ☐ Erasmus

☐ Master of Science in Artificial Intelligence and Robotics ☐ Other

Ex 1: Provide the specification of a perfect failure detector, describe how it can be implemented and discuss why moving from a synchronous system to an eventually synchronous one it is no longer possible to guarantee its correctness.

Ex 2: Consider the partial execution depicted in the Figure

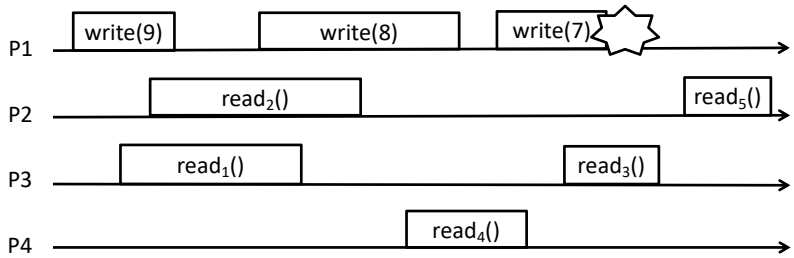


Answer to the following questions:

1. Provide ALL the possible delivery sequences that satisfies causal order and TO (UA, SUTO)
2. Complete the execution in order to have a run satisfying TO (UA WNUTO), FIFO order Broadcast but not Causal Order Broadcast
3. Complete the execution in order to have a run satisfying Regular Reliable Broadcast but not Uniform Reliable Broadcast and not satisfying Total Order.

NOTE: In order to solve the exercise, you can only add broadcast, deliveries and failures.

Ex 3: Consider the partial execution depicted in the Figure



Answer to the following questions:

1. Define ALL the values that can be returned by read operations (Rx) assuming that the run refers to a regular register.

2. Define ALL the values that can be returned by read operations (Rx) assuming that the run refers to an atomic register.
3. Assign to each read operations (Rx) a return value that makes the execution linearizable.

Ex 4: Let us consider the following algorithm

```
upon event ⟨ frb, Init ⟩ do
lsn := 0;
pending := ∅;
next := [1]^N;

upon event ⟨ frb, Broadcast | m ⟩ do
lsn := lsn + 1;
for each p ∈Π do
        trigger ⟨ Send | [DATA, self, m, lsn] ⟩ to p;

upon event ⟨ Deliver | p, [DATA, s, m, sn] ⟩ do
pending := pending ∪ {(s, m, sn)};

while exists (s, m', sn') ∈ pending such that sn' = next[s] do
next[s] := next[s] + 1;
pending := pending \ {(s, m', sn')};
trigger ⟨ frb, Deliver | s, m' ⟩;
```

Assuming that messages are sent by using fair loss links, for each of the following properties, discuss if it satisfied or not and provide a motivation for your answer:

- ***Validity:*** If a correct process p broadcasts a message m, then p eventually delivers m.
- ***No duplication***: No message is delivered more than once.
- ***No creation:*** If a process delivers a message m with sender s, then m was previously broadcast by process s.
- ***Agreement***: If a message m is delivered by some correct process, then m is eventually delivered by every correct process.
- ***FIFO delivery***: If some process broadcasts message m_1 before it broadcasts message m_2, then no correct process delivers m_2 unless it has already delivered m_1.

Ex 5: Consider a distributed system constituted by *n* processes $\prod=\{p_1, p_2 \ldots p_n\}$ with unique identifiers that exchange messages through perfect point-to-point links and are structured through a ring (i.e., each process p_i can exchange messages only with processes and $p_{i+1(\text{mod } n)}$). Processes may crash and each process is equipped with a perfect oracle (having the interface *new_next(p)*) reporting a new neighbor when the previous one is failing.
Write the pseudo-code of an algorithm implementing a FIFO Order Broadcast communication primitive.

According to the Italian law 675 of the 31/12/96, I authorize the instructor of the course to publish on the web site of the course results of the exams.

Signature: _____