Distributed Algorithms 2015
Midterm – Solution

23rd of November, 2015

Name:

Sciper number:

Question 1 *(9 points)*

We consider a distributed system with processes that can crash. Mark each of the following properties with:

S, if it is a safety property, or

L, if it is a liveness property.

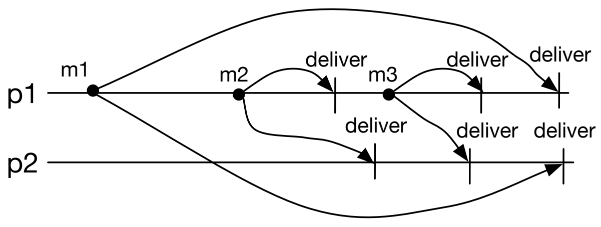
1. If a process p decides m, then no process decides on m', where $m' \neq m$. *(3 points)*
   (S)
2. If a correct process p delivers a message m, then no other correct process delivers m. *(2 points)*
   (L)
3. If a process broadcasts message $m1$ and then broadcasts message $m2$, then no correct process delivers $m2$ before delivering $m1$. *(2 points)*
   (L)
4. If a process decides m, then a correct process previously proposed m. *(2 points)*
   (S)

Question 2 *(8 points)*

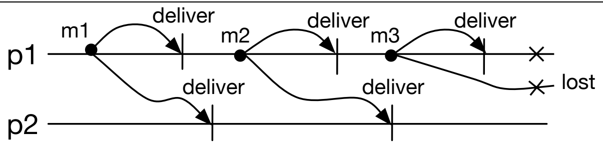
Consider a distributed system with two processes $p1$ and $p2$, where process $p1$ broadcasts three messages $m1$, $m2$, and $m3$. Both processes deliver at least the first two messages ($m1$ and $m2$).

Consider the following two cases:

1. Suppose that $p1$ and $p2$ use an algorithm that ensures total-order broadcast and no process crashes. Sketch an execution history which satisfies total-order but breaks the specification of causal-order. *(4 points)*



2. Suppose that $p1$ and $p2$ use an algorithm that ensures reliable broadcast and process $p1$ crashes at some point. Sketch an execution which satisfies reliable broadcast but does not satisfy uniform reliable broadcast. *(4 points)*



Question 3 *(10 points)*

1. Provide the properties of consensus and uniform consensus along with a brief and concise description of each property. *(4 points)*

   (a) Consensus properties are as follows.

   i. Validity: Any value decided is a value proposed.
   ii. Agreement: No two correct processes decide differently.
   iii. Termination: Every correct process eventually decides.
   iv. Integrity: No process decides twice.

   (b) Uniform consensus properties are as follows.

   i. Validity: Any value decided is a value proposed.
   ii. Uniform Agreement: No two processes decide differently.
   iii. Termination: Every correct process eventually decides.
   iv. Integrity: No process decides twice

2. Assuming that the integrity property always holds, show that for uniform consensus each of the other properties is necessary to make the problem non-trivial (i.e, omitting one of the remaining three properties makes the problem trivial to solve even if processes may crash). For each of the three remaining properties, provide a simple algorithm to solve uniform consensus where that property is omitted. *(6 points)*

   (a) Consensus defined by *integrity*, *uniform agreement*, and *validity*:
   Trivial solution: no process decides.
   Code of process p: empty.

   (b) Consensus defined by *integrity*, *uniform agreement*, and *termination*:
   Trivial solution: every process decides on a predefined value.
   Code of process p: DECIDE(0).

   (c) Consensus defined by *integrity*, *validity*, and *termination*:
   Trivial solution: every process decides on its initial value.
   Code of process p: DECIDE(v_p) (v_p is the initial value of p).

Question 4 *(6 points)*

1. Provide the properties for Terminating Reliable Broadcast (TRB) along with a brief and concise description of each property. *(2 points)*

   TRB properties are as follows.

   (a) Integrity: If a process delivers a message m, then either m is ϕ or m was broadcast by src.

   (b) Validity: If the sender src is correct and broadcasts a message m, then src eventually delivers m.

   (c) Agreement: For any message m, if a correct process delivers m, then every correct process delivers m.

   (d) Termination: Every correct process eventually delivers exactly one message.

2. You are provided with the following.

   (a) A synchronous system S where at most f crash failures can occur.

   (b) An algorithm A that solves consensus in $f + 1$ rounds for S.

   (c) An algorithm B that solves reliable broadcast in 1 round for S.

   Now, propose an algorithm T to solve Terminating Reliable Broadcast for S in $f + 2$ rounds and provide its correctness in terms of the properties for TRB. Note that the delivery of message is executed in the same round in which the decision of consensus is reached. *(4 points)*
   Hint: Think about the requirement for the consensus algorithm to start.

   Basic idea. The protocol starts with the source (src) broadcasting a message m in the 1^{st} round. All processes wait for the receipt of m till the end of round 1. Next, all the processes run the consensus algorithm with their proposed value either m (if they received it) or ϕ (if they did not receive it). Finally, when the consensus algorithm terminates they deliver whatever is decided by the consensus algorithm. The delivery of message is executed in the same round in which the decision of consensus is reached.
   Number of rounds. The total number of rounds: 1 (broadcast operation) + f+1 (execution of consensus algorithm and message delivery)= f+2.

   Algorithm T.

```
(a) If (p == src) broadcast message $m$ using algorithm $B$.
(b)   i. If received $m$, then set $v$ = $m$
         else set $v$ = $\phi$
     ii. Propose $v$ for consensus.
    iii. Set $v$ to the decision of consensus algorithm
         Deliver($v$)
```

   Correctness proofs.

   Integrity. Every process proposes v (received from src) or ϕ for consensus.

   Validity. If the sender src is correct, then it decides m in consensus and hence also delivers m.

   Agreement. If a process delivers, then it delivers whatever is decided by Consensus. Hence, if a correct process delivers m, then all correct processes deliver m.

   Termination. Deliver(v) is executed only once in step 2(c) and it delivers only 1 message. Hence, every correct process delivers exactly one message.

Question 5 *(8 points)*

Consider the fail-stop algorithm for 1-N regular registers using failure detector P. Given the following cases, does the algorithm still satisfy regularity? Explain your answer.

1. P is Strong Complete but Eventually Strong Accurate. *(4 points)*
2. P is Strong Accurrate but Completeness is violated. *(4 points)*

In the first case, when a correct process p_i is suspected, the writer does not wait for the Ack from p_i and v_i is not updated by the time of the end of write. So, p_i may read the old value and it violates the regularity.

In the case of violation of completeness, the writer waits infinitely to get Ack from some failed process (which was not suspected by the failure detector). So the write operation never finishes.

Question 6 *(9 points)*

For each of the executions below, indicate whether the register is regular, whether it is atomic, or if it is neither regular nor atomic. Give a short explanation of your answer.

Note: The initial value of each register is 0.

1. *(3 points)*

```
P1: [  W(1)               ]
P2:       [R()->0]    [R()->0]
P3:            [    R()->1   ]
```

(Atomic)

2. *(3 points)*

```
P1:    [   R()->2        ]
P2:      [ R()->1]  [    W(2)   ]
P3:                          [  R()->1 ]
```

(Neither regular nor atomic)

3. *(3 points)*

```
P1:  [ R()->2 ]        [     R()->0    ]
P2:        [     W(2)     ]
P3:                     [  R()->2   ]
```

(Regular)