EE 360P **TEST 2** **Vijay K. Garg** **Spring'17**

NAME:

UT EID:

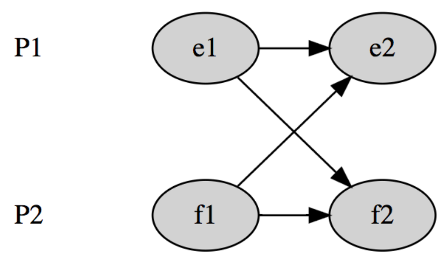
Honor Code: I have neither cheated nor helped anybody cheat in this test.
Signature:

Time Allowed: 75 minutes
Maximum Score: 75 points
Instructions: This is a CLOSED book exam. Attempt all questions.

Q. 1 **(15 points)** Please be concise in your answers.

(a, 2 points) Let $(E, \rightarrow)$ be a distributed computation. Define a *consistent global state* (or *a consistent cut*) for this computation in the event-based model.

(b, 5 points) Draw the lattice of all the consistent global states of the following computation $(E, \rightarrow)$ (valid subsets of $E = \{e1, e2, f1, f2\}$ that are consistent cuts)



(c, 5 points) Give an example of a binary relation on $X = \{a, b, c\}$ such that the relation is nonempty, symmetric, transitive but not reflexive.

(d, 3 points) Give one advantage and one disadvantage of using Java RMI compared to using TCP. What is the role of stub routines generated by Java compiler for remote objects?

Q. 2 **(10 points)** Suppose that a programmer has proposed the following algorithm for mutual exclusion in a distributed system. There is a unique token in the system and only the process with the token can enter the critical section. The token carries with it a queue of requests. Any process that wants the critical section broadcasts its request. On receiving a request, the process with the token adds that request to the token. If a process does not have the token, it ignores the request. Whenever a process with the token is done with the critical section, it checks the queue in the token. If the queue is empty, it waits for the queue to become nonempty. Otherwise, it deletes the request at the head of the queue and sends the token to the process that made that request.

(a, 5 points) Does this algorithm satisfy the safety property (two processes can never be in the critical section at any time)? Justify your answer.

(b, 5 points) Does this algorithm satisfy the progress property (every request is eventually granted)?

Q. 3 **(4 points)**

(a, 2 points) If the logical clock value for event e is 40 and for event f is 50, then what can be inferred about happened-before order relationship between e and f?

(b, 2 points) Suppose that the vector clocks for events g,h and k are $(4, 3, 5)$, $(6, 5, 5)$, and $(3, 2, 7)$, respectively. What can be inferred about happened-before relationship among events g, h and k?

Q. 4 **(12 points)** Suppose that a distributed system with three processes P_1, P_2, and P_3 has *two* tokens that move around these processes. A process P_0, that is external to the system, sends a request to each of P_i (where $i = 1..3$) to respond back with the number of tokens it currently has. It them sums up the counts it receives from all three processes.

(a) Is it possible for P_0 to conclude that the total number of tokens in the system is 4? If yes, show a computation (process-time diagram) that results in P_0 concluding that the total number of tokens in the system is 4. If not, justify your answer.

(b) Is it possible for P_0 to conclude that the total number of tokens in the system is 0? If yes, show a computation (process-time diagram) that results in P_0 concluding that the total number of tokens in the system is 0. If not, justify your answer.

Q. 5 **(10 points)**
(a, 1 point) When is a predicate B defined on a consistent global state *stable*?

Which of the following predicates are stable? Justify your answer.
(b, 2 points) P_1 is waiting for P_2 to release a resource and P_2 is waiting for P_1 to release a different resource.

(c, 2 points) There are no "request" messages in the channel from P_1 to P_2 (i.e. there are no "request" messages in-transit from P_1 to P_2).

(c, 2 points) When does a node in Chandy and Lamport's algorithm know that it has finished its part of the global snapshot algorithm?

(d, 3 points) Suppose that a message takes one unit time to traverse any link. Assume that computation on any node takes zero time. Give the maximum time Chandy and Lamport's algorithm will take on a distributed systems with the following topology when it initiated by one or more nodes in the algorithm. Justify your answer.

```
p1 ---------->   p2  --------> p3 ------------>  p4
/|\               |                              |
 |                |                              |
 |                |                              |
 |               \|/                            \|/
 p5  <---------  p6 <------- p7  <-----------  p8
```

Q. 6 **(12 points) Dining Philosopher Algorithm**

Consider the dining philosopher algorithm for resource allocation in distributed systems. Assume that the conflict graph for resources may not be complete, i.e., there may not be a direct edge from every philosopher to every other philosopher. Assume that all philosophers have a very short thinking time, i.e., as soon as they finish eating, they become hungry again. Let $numEat(u)$ denote the number of times the philosopher u gets to eat when we follow the rules of the dining philosopher algorithm. Assume that philosophers u and v are connected by a path of length k. Show that the difference between $numEat(u)$ and $numEat(v)$ is at most k.

Q. 7 **(10 points)** (a, 6 points) How will you detect the predicate *possibly* : B in a distributed computation? The variables x, y and z are integers on different processes. $B = \neg((x \geq 3) \vee ((y \geq 0) \wedge (z \geq 0)))$.

(b, 3 points) Suppose that your friend has implemented a mutual exclusion algorithm for 3 processes. Give the predicate B that you will use to detect violation of mutual exclusion.
(Hint: Define appropriate local predicates and a boolean expression of those local predicates that corresponds to violation of mutual exclusion.)

(c, 3 points) How will you detect if the computation has a consistent global state that satisfies B in part (b)? You do not have to give the details of any algorithm covered in the class. Just outline which algorithm you will use and how.