Review for 22C:166 Exam 1

Model conversion exercises
Liveness and safety properties: Proving (or at least reasoning about) them
Writing and analyzing simple algorithms for various problems
Understand atomicity, fairness and their consequences
Logical clocks, physical clocks, and vector clocks
Distributed snapshots: Recognizing consistent cuts

Sample questions

1. Consider the problem of phase synchronization, where a set of processes (in a completely connected topology) executes their actions in phases, and no process is allowed to execute phase \((k+1)\) until every process has completed its phase \(k\) \((k \geq 0)\). This is useful in parallelizing loop computations. Assuming each process to have a non-negative integer variable \(p\) that represents its phase, implement phase synchronization using the shared memory model.

2. A distributed system consists of a completely connected network of three processes. Each process wants to pick a unique identifier from the domain \(\Sigma = \{0, 1, 2\}\). Let \(\text{name}[i]\) represent the identifier chosen by process \(i\). The initial values of name are arbitrary. Here is a proposed solution:

\[
\text{program} \ \text{pick-a-name} \ \{\text{for process } i\}\n\text{define} \ \text{name}: \ \text{integer} \in \{0, 1, 2\}\n\text{do} \ \exists j \neq i: \ \text{name}[j] = \text{name}[i] \land x \in \Sigma \setminus \{\text{name}[j]: j \neq i\} \rightarrow \text{name}[i] := x
\text{od}
\]

Comment on the correctness of the above solution.

3. Consider a tree \((V, E)\), where each vertex \(v \in V\) represents a process, and each edge \(e \in E\) represents an undirected edge. Each process \(v\) has a color \(c[v] \in \{0,1\}\). Starting from an arbitrary initial configuration, the nodes have to acquire a color such that no two neighboring nodes have the same color. We propose the following algorithm for each process \(i\):

\[
\text{program} \ \text{twocolor}
\text{define} \ c[i]: \ \text{color of process } i \ \{c = 0 \text{ or } 1\}
\text{do} \ \exists j \in \text{neighbor}(i): \ c[i] = c[j] \rightarrow c[i] := 1 - c[i] \ \text{od}
\]

Assume that the scheduler is weakly fair. Will the algorithm terminate? Provide brief arguments.
4. The following computation runs on a unidirectional ring of N processes 0,1,2, ... N-1 (N>3). Processes 0 and N-1 are neighbors. Each process j has a local integer variable x[j] whose value is in the range 0..K-1 (K > 1).

\{process 0\} \quad \textbf{do} \ x[0] = x[N-1] \rightarrow x[0] := x[0] + 1 \mod N \ \textbf{od}

\{process j > 0\} \quad \textbf{do} \ x[j] \not= x[j-1] \rightarrow x[j] := x[j-1] \ \textbf{od}

Prove that the above computation will not deadlock.

5. \(a, b, c\) are three events in a distributed system, and no two events belong to the same process. Using Lamport's definition of sequential and concurrent events, comment on the truth of the following statements:

(a) \((a \parallel b) \land (b \prec c) \Rightarrow a \prec c\)
(b) \((a \parallel b) \land (b \parallel c) \Rightarrow a \parallel c\)

(Notation: \(a \parallel b\) denotes that \(a, b\) are concurrent events)

7. Compute vector clock values for a set of events