CALIFORNIA INSTITUTE OF TECHNOLOGY
Computing and Mathematical Sciences

CS 142

R. Murray, M. Chandy — **Homework Set #5** — Issued: 6 Nov 2019
Fall 2019 — Due: 13 Nov 2019

Note: Please keep track of the number of hours that you spent on this homework set (including reading) and enter this when you turn in your homework on Moodle.

1. Properties of unless.

   (a) Show that the following statements are all equivalent ways of specifying that the value of a variable x never decreases:

   - $(\forall k :: x = k \textbf{ unless } x > k)$
   - $(\forall k :: x \geq k \textbf{ unless } x > k)$
   - $(\forall k :: x \geq k \textbf{ unless false})$

   (b) Show that $\textbf{stable}(P) \equiv P \textbf{ unless false}$.

2. Consider the safety property for the dining philosopher's problem:

$$P = \big(\forall u, v :: \textbf{invariant}(\neg(E(u,v) \wedge u.e \wedge v.e))\big).$$

   Provide a short rationale that the following property is a refinement of that property:

$$Q = \big(\forall u, v :: \textbf{invariant}(u.e \wedge E(u,v) \implies \text{fork}(u,v) = u)\big)$$

   (that is, provide an argument for why $Q \implies P$).

3. A key idea in conflict resolution algorithms, as illustrated by the dining philosophers problem, is to maintain a priority graph structure among agents which has the following properties:

   1. The graph is acyclic, and
   2. The graph changes in a fair way, so that every agent that needs a resource gets higher priority than its neighbors eventually. In the case of the dining philosophers problem: every hungry agent gets to eat eventually.

   One class of algorithms for conflict resolution uses each agent's local clock time. The local clocks are not synchronized. If $t[i]$ is the time for agent i and T is the real time, then the maximum drift is a constant value, ϵ, i.e.,

$$\forall i : |t[i] - T| \leq \epsilon$$

   We don't know the value of ϵ but we know it exists. So, clocks cannot drift apart by arbitrarily large amounts.

   Each agent's clock moves forwards (never backwards and never stays still). Assume that clock values and ϵ are integers so that we can carry out induction on their values. (If clock values were real numbers then we can have a situation where a clock ticks forward by 0.5^n on the n-th step, and thus the clock never increases by more than 2.)

Also, when an agent makes a state transition (e.g., thinking to hungry) the agent's clock ticks forward. An agent does not make a state transition while its clock remains unchanged (but the agent's clock can continue to tick forward while it remains in its current state).

This question is about the relationship, if any, between acyclic priority graphs and time-based conflict resolution.

(a) Which of the following methods would you use to define the priority in terms of local clock values so that the two conditions (listed above) for priority graphs are met? In particular, could you solve the dining philosophers problem by allowing a hungry philosopher with higher priority than all its neighbors to eat, where priority was defined in terms of local clock time?

1. An agent's priority is the ordered pair [agent's current clock value, agent id].
2. An agent's priority is the agent's current clock value.
3. An agent's priority is defined as follows. If the agent is waiting to enter a critical section (i.e., hungry) or in a critical section (eating) then the priority is the ordered pair [the agent's local clock value at the point at which the agent last started waiting, agent id]. If the agent is executing outside the critical section (thinking) then the priority is [agent's current clock value, agent id].

Note: Lower numbers indicate higher priority.

(b) Show that the method you chose satisfies the safety property that the priority graph is acyclic.

(c) Show that the method you chose satisfies the progress property that every hungry agent in dining philosophers gets to have higher priority than all its neighbors eventually (and therefore gets to eat eventually).