Philosophers!

Homework #4, CSCI 322, Winter 2015
Geoffrey Matthews

February 10, 2015

Description: This is based on Exercise 4.17 from the text.

Solve the dining philosophers problem by focusing on the state of the philosophers rather than the forks. In particular, let `eating[n]` be a boolean array; `eating[i]` is true if `Philosopher[i]` is eating and is false otherwise.

(a) Specify a global invariant, then develop a coarse-grained solution, and finally develop a fine-grained solution that uses semaphores for synchronization. Your solution should be deadlock-free, but an individual philosopher might starve. Use the technique of passing the baton. Describe how your fine-grained solution matches the semantics of the coarse-grained solution.

Implement your fine-grained solution in either **Racket**, **Python** or **Pthreads**. Be sure to use only semaphores for synchronization and mutual exclusion. Name the semaphores well. Insert your invariant at relevant places as an assertion, *i.e.*, the evaluation of a boolean expression or expressions which, if false, halts the program with an error message.

(b) Modify your answer to (a) to avoid starvation. In particular, if a philosopher wants to eat, eventually he or she gets to eat. Describe your strategy in clear language. If it lends itself to an invariant (not all strategies do), state that clearly.

Develop both coarse-grained and fine-grained solutions, and implement the fine-grained solution in your chosen language.

Writeup: There should be three files submitted:

- The writeup, which includes both parts, justifications, *etc.* It should look something like the examples in the textbooks. You can use Andrews-like pseudocode here.
- A runnable (or compilable on unix) solution to part (a).
- A runnable (or compilable on unix) solution to part (b).

Due date: Wednesday, February 25, midnight.