Homework 1 Theory

Michael Tang

April 16, 2019

1 Exercise 2

1. Safety. Bad thing of interest would be patrons being served out of order.
2. Safety. Bad thing: something that goes up doesn't come down.
3. Liveness. Good thing: at least one process succeeds.
4. Safety. Bad thing: message printed after more than one second.
5. Liveness. Good thing: message getting printed.
6. Liveness. Good thing: cost of living not increasing.
7. Liveness. Good thing: death and taxes occurring.
8. Liveness. Good thing: telling a Harvard man.

2 Exercise 3

Instead of just one can being on Alice's end, put a can on each end of the string, one with Alice and one with Bob. If Alice's can is down, when she resets it to request food the for the pets she yanks the string to knock down Bob's can; Bob then knows Alice has reset her can. When Bob has finished placing the food and returned to his house, he can reset his can and yank the string to knock over Alice's.
For mutual exclusion, the same state-based proof applies except with the addition of Bob's can that is always in the state opposite that of Alice's can. With this always-opposite state Bob is able to get the same message as if it were just Alice's can visible on the windowsill and he were looking at it.
For starvation-freedom, it is not possible for Alice and Bob's can to be up at the same time. Therefore either Bob is providing food or the pets don't need any.
For producer-consumer, the states are again the same except with Bob's can the state opposite of Alice's can.

3 Exercise 4

4 Exercise 5

5 Exercise 7

$S_2 = \frac{1}{1-p+p/2}$ where p is the fraction of the program that is parallelizable
$1 - p + p/2 = \frac{1}{S_2}$

$p = -2(\frac{1}{S_2} - 1)$
$S_n = \frac{1}{1-p+p/n} = \frac{1}{1+(2/S_2)-2-\frac{2}{nS_2}+(2/n)}$
$S_n = \frac{1}{\frac{nS_2+2n-2nS_2-2+2S_2}{nS_2}}$
Result: $S_n = \frac{nS_2}{2n-nS_2+2S_2-2}$

6 Exercise 8

Speedup must be greater than 1 for multiprocessor to be worth it.
If 1 is the time for one one-zillion-instruction-per-second processor to process the task, then for a five-zillion-instruction processor that task should take 1/5. Using Amdahl's Law:
$S = \frac{serial5zillion}{parallel1zillion} = \frac{1/5}{1-p+p/10} > 1$
$1 - p + p/10 < 1/5$
$-\frac{-9p}{10} < -4/5$
$9p > 40/5$
$p > 40/45$
$p > 8/9$
I would buy the multiprocessor if more than 8/9 of the code can be executed in parallel.

7 Exercise 11

Mutual exclusion: yes. Assume for contradiction that second thread manages to pass the lock before the first thread currently in the critical section has called $Flaky.unlock()$. Since the second thread had to enter the $while(busy)$ loop first, that means it broke free because $busy == False$. However, the last time $busy$ was set was when the first thread passed through $Flaky.lock()$, and it was set to $True$. But since $busy == False$ the first thread must have called $Flaky.unlock()$, which violates the condition that the first thread is still in the critical section.
Starvation-free: no. If the first thread were so much faster than the second thread that it finished the critical section, called $Flaky.unlock()$, then called $Flaky.lock()$, set $busy = True$, then looped back because $turn$ is set to the second thread ALL BEFORE the first thread left the $while(busy)$ loop, then both threads would end up stuck in that loop.
Deadlock-free: no. See example for starvation-free.

8 Exercise 14

9 Exercise 15

10 Exercise 16