Brock Sauvage
2796131
Lab Report

1) The asymmetric solution is to essentially have the philosophers around the table be alternating in which order they pick up the chopsticks. More plainly stated, the philosophers that are placed at even-numbered indexes in the philosopher array pick up the chopsticks from left to right, while the odd-numbered indexes pick them up from right to left. This guarantees that deadlock is never reached because when an even numbered philosopher utilizes their left chopstick, and the odd numbered philosopher to the right utilizes their right chopstick, there will be one chopstick still available between the two. Thus, even if there is a race to the chopstick, the winning philosopher will eventually release the chopstick for the other philosopher to utilize.
2) Starvation may still occur because if a set of philosophers begins to utilize and surrender chopsticks around the same time (get into a rhythm) then another set of philosophers may get put at a disadvantage and never get to acquire the chopsticks.
3) Essentially, the waiter solution utilizes a mutex lock (the "waiter") in order to manage how and when the philosophers get their chopsticks. When a waiter acquires the lock, they will check to see if both chopsticks are available. If not, then they will wait until the right conditions are met and a chopstick is available. If they are available, the philosopher with the lock will get to eat. Once complete, the lock will be released and the other philosophers signaled.
4) The waiter solution does not prevent deadlock, as some philosophers, depending on when they get signaled, may never acquire the lock if other philosophers acquire if first. They would prevent philosophers that never acquire the lock from eating, therefore reaching starvation.
5) Phil might find that both of the chopsticks are not free if another philosopher is also waiting and acquires the chopsticks before Phil is able to. It would be under this circumstance that Phil would find the chopsticks unavailable upon checking.