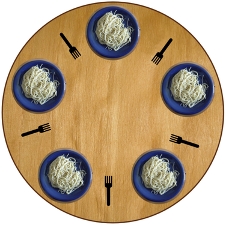
4. (20 pts) (Ex 1.8.2) The dining philosophers problem was invented by E. W. Dijkstra, a concurrency pioneer, to clarify the notions of deadlock and starvation freedom. Imagine five philosophers who spend their lives just thinking and feasting. They sit around a circular table with five chairs. The table has a big plate of rice. However, there are only five chopsticks (in the original formulation forks) available, as shown in Fig. 1.5. Each philosopher thinks. When he gets hungry, he sits down and picks up the two chopsticks that are closest to him. If a philosopher can pick up both chopsticks, he can eat for a while. After a philosopher finishes eating, he puts down the chopsticks and again starts to think.



a. What is wrong with everybody doing the following - other than that the philosophers never get up from the table?

1. think for a while
2. get left chopstick
3. get right chopstick
4. eat for a while
5. return left chopstick
6. return right chopstick
7. return to 1

The issue with everyone doing the following is they can reach a deadlock. For example, if at the start every philosopher stops thinking at the same time and picks up their left chopstick then all 5 philosophers will have a left chopstick but there will be no right chopstick and they will be stuck infinitely in the 2nd state.

b. How can the above be fixed to avoid deadlocks?

A simple solution to avoid deadlocks is the following:

1. Think for a while
2. Pick up left and right chopstick if none available return to 1
3. Eat for a while
4. Return left and right chopstick
5. return to 1

c. Is your solution starvation free? Literally!

This solution is not starvation free. There is a possible timing issue. There is the possibility that each time a philosopher is ready to eat one of the adjacent philosophers has began eating.

1. (10 pts) Describe the environment in which a wait () call is legal?

A wait call must always be placed in a synchronized block. Also, it must accompany a notify() so that the intercommunication between threads works fine!