Discrete Event Simulation Case Study: Dining Philosophers Problem

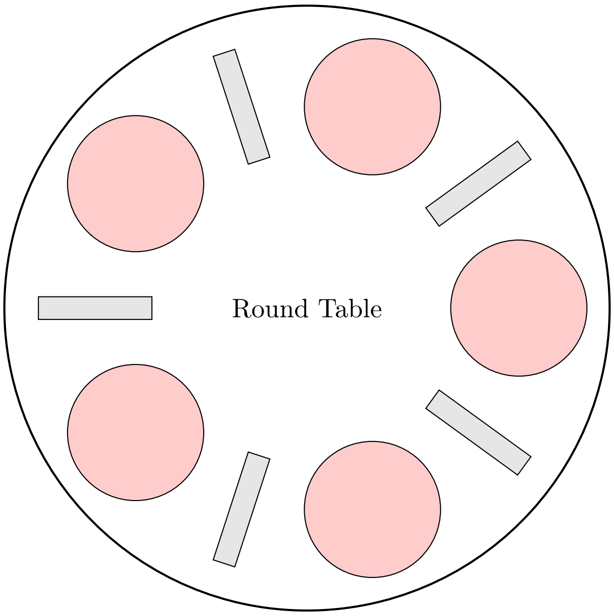

Figure 1: Dining philosophers sitting at a round table, each with a bowl of food.

Research Question

A group of philosophers is known for their rigorous thinking and minimal needs. However, their practice of sharing chopsticks while eating has led to conflicts and inefficiencies, which may even cause starvation. Your task is to model their behavior, simulate their interactions, and develop strategies to address inefficiencies and prevent starvation.

Modelling Approaches

The dining philosophers problem is a classic example of resource contention in systems. This project explores how discrete event simulation can model, analyze, and address the dynamics of the dining philosophers. The problem will be modeled with the following features:

- Philosophers alternate between **thinking**, **hungry**, and **eating**.
- Each philosopher requires two chopsticks to eat: one from their left and one from their right.
- Chopsticks are shared between neighboring philosophers, and contention may arise.
- The primary objective is to prevent deadlocks and starvation while ensuring fair access to chopsticks.

The model will incorporate individual philosopher behavior and system-wide rules.

Philosopher Behavior

Each philosopher alternates between three states:

- **Thinking:** The philosopher is not hungry and does not need chopsticks.
- **Hungry:** The philosopher attempts to acquire two chopsticks.
- **Eating:** The philosopher has acquired two chopsticks and eats before returning them.

State transitions, e.g. how long a philosopher thinks before becoming hungry, are determined probabilistically.

Discrete Event Simulation

The system is modeled as a discrete event simulation with the following components:

- **Entities:** Philosophers and chopsticks.
- **Events:** Changes in state, such as a philosopher transitioning from thinking to hungry or acquiring/releasing chopsticks.
- **Schedule:** A timeline of events determines the order in which philosophers act.

Initial Setup

- Five philosophers are seated around a round table.
- Each philosopher starts in the **thinking** state.
- Five chopsticks are placed between neighboring philosophers.
- Random delays are introduced for thinking, attempting to acquire chopsticks, and eating.

ToDo's

Task 1: Basic Simulation

Implement a discrete event simulation of the dining philosophers problem with the following assumptions:

- Philosophers have three states: **thinking**, **hungry**, and **eating**.
- Chopsticks can only be held by one philosopher at a time.
- Simulate 100 time units and track how often each philosopher eats.

Task 2: Deadlock Detection

Extend the simulation to identify deadlock situations. A deadlock occurs when all philosophers are hungry but unable to acquire both chopsticks, e.g. when each philosopher holds the chopstick from their left. Implement a detection mechanism and log deadlock events.

Task 3: Deadlock Prevention

Modify the simulation to prevent deadlocks using strategies such as:

- Assigning priorities to chopsticks to ensure an ordering rule (e.g., philosophers always pick the lower-priority chopstick first).
- Imposing timeouts for acquiring chopsticks, after which a philosopher releases any held chopstick and retries.

Simulate the system with these strategies and compare their effectiveness.

Task 4: Starvation Avoidance

Ensure that all philosophers have a chance to eat by introducing fairness mechanisms:

- Implement a mechanism to track and prioritize the h philosophers.
- Track the time since each philosopher last ate and prioritize philosophers who have waited the longest.

Simulate and measure the performance of this approach.

Task 5: Complex Scenarios

Experiment with additional scenarios:

- Increase the number of philosophers and chopsticks.
- Introduce variability in thinking and eating times.
- Simulate "selfish" philosophers who do not follow the rules.
- Explore conspiracies among philosophers, where a subset intentionally blocks others from eating.

Analyze the impact of these changes on system dynamics and performance.

Discussion

- Compare the performance of the different strategies implemented in Tasks 3 and 4.
- Discuss the trade-offs between simplicity, efficiency, and fairness in the chosen strategies.
- Reflect on the relevance of the dining philosophers problem to modern systems, such as deadlock prevention in operating systems or resource contention in distributed computing.