Exercise 1: Happens Before in Shared Memory

Consider \( n \) processors and \( m \) shared variables. Every processor can access every shared variable with atomic read and write operations (i.e., a process can either read from or write to a shared variable and the system guarantees that such accesses of different processes to the same variable happen atomically). Define a happens before relation similar to the one for message passing.

Exercise 2: Unique Maximal Cut Preceding a Given Cut

Given a schedule \( S \) with some cut \( C \). Show that there is a unique, maximal consistent cut \( C' \) of \( S \) which precedes the cut \( C \).

Remarks: A cut \( C' \) precedes \( C \) if \( C' \subseteq C \). A cut is maximal with respect to a given property if it contains the most events among all cuts with that property.

Exercise 3: Happens Before Relation

Let \( S \) be a schedule with events \( a \), \( b \), and \( c \). Show that if \( a \not\Rightarrow_S b \) and \( a \not\Rightarrow_S c \) holds, then there exists some causal shuffle \( S' \) of \( S \) in which \( b \) and \( c \) occur before \( a \).

Exercise 4: Logical Clocks

You are given a clique graph on \( n \) nodes. Find two executions \( A \) and \( B \), in which each node sends exactly one message to every other node, such that

a) the largest Lamport clock value in \( A \) is as small as possible, and

b) the largest Lamport clock value in \( B \) is as large as possible.