Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
Philipp Bamberger

Distributed Systems, Summer Term 2020
Exercise Sheet 4

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.

Sample Solution

Given some schedule S, the happens before relation contains:

1. All events (e, e') where e precedes e' in S and e and e' are events of the same process/node.
2. All events (e, e') where e precedes e' in S and which access the same shared variable i, and one of the events accessing variable i in between e and e' (including e and e') is a write event.
3. The transitive closure of the relation defined by 1 and 2.

2. Unique Maximal Cut Preceding a Given Cut

Given a schedule S with a cut C. Show that there is a unique consistent cut C' of S which precedes the cut C.

Sample Solution

Proof Sketch: Start with the (non-consistent) cut C. Whenever there is an unmatched receive event in one local view remove from the cut that event and all events after it in that local view. This process terminates and will succeed in finding the unique consistent maximal cut. The defined cut precedes the cut C and it is consistent because it does not contain any unmatched receive event.
It is maximal because if it is increased (within C) it will be increased by a receive event contradicting the consistency.
It is easy to see that the maximal cut is unique. Let C' and C'' be two maximal consistent cuts preceding cut C. Assuming that, w.l.o.g., there is some event e in C' and $e \notin C''$. Then $C'' \cup \{e\} \cup \{e' | e' \not\Rightarrow_S e\}$ is a consistent cut preceding C and strictly larger than C'', a contradiction. Hence $C' = C''$.

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.

Sample Solution

The formal proof is a little bit tedious, but the high level idea is the same as in the lecture Causality. The proof is easiest to be understood if you draw the same picture as given on the slides. For every node i let $e_i \in S|i$ be the event s.t. $a \Rightarrow_S e_i$ and for all $e \in S|i$ with $a \Rightarrow_S e$ we have $e_i \Rightarrow_S e$ (e_i might not exist for some i, but then this local view does not matter). Let i_a, i_b and i_c be the indices of the local views in which a, b and c occur and associate global (discrete) times $\iota(e)$ with every event $e \in S$ s.t. no two events have the same time and the time is consistent with the schedule S. Note that $e_{i_a} = a$. Define $\Delta = \max\{\iota(e_{i_b}), \iota(e_{i_c})\} - \iota(e_{i_a})$. Now we obtain a causal shuffle of S by shifting all elements in view i with $e_i \Rightarrow_S e$ by $\Delta + 1$ time units (to the later time). We can deduce a new schedule from these times which is a causal shuffle of S with the desired properties.

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

- the largest Lamport clock value in A is as small as possible, and
- the largest Lamport clock value in B is as large as possible.

Sample Solution

- At every node there are $n-1$ send and $n-1$ receive events, so $2n-2$ is a lower bound on the clock value. This bound is matched if all messages are sent before any message is received.
- The largest clock value is as large as possible if the messages are sent one after another along an Eulerian tour/cycle of the graph where each edge is replaced with two edges of opposite direction. (An Eulerian cycle is a cycle of the given graph that traverses all the edges of the graph exactly once). In such a graph, there always exists an Eulerian cycle, since it satisfies that each node v has $\deg_{in}(v) = \deg_{out}(v)$ and there is a single strongly connected component. For every message across a (directed) edge the clock value is increased by one. So the maximal Lamport clock value will be $2 \times$ (nr. of directed edges) $= 2n(n-1)$.