University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Fuchs, A. Mályusz, Z. Parsaeian

Theory of Distributed Systems
Sample Solution Exercise Sheet 4

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.

Sample Solution

Given some schedule S, we give the happens before relation as set of tuples R_S (ie., $(e, e') \in R_S$ iff $e \Rightarrow_S e'$). We define R_S such that all orders of events that some node knows about somehow, are reflected in the relation (that, in turn, makes it possible to talk about concepts such as a causal shuffle in the parallel setting). Principally, a processor learns about the order of events in two ways. Either they happen on the same processor anyway, or the processor reads a shared variable that some other process has written before. More precisely we define:

1. $(e, e') \in R_S$ if e precedes e' in S and e and e' are events of the same processor.
2. $(e, e') \in R_S$ if e precedes e' in S, and e is a write event and e' is a read event which access the same shared variable and that variable is not overwritten by some other event between e and e'.
3. The transitive closure of the relation defined by 1 and 2.

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.

Sample Solution

Intuition: From the lecture we know that only message events may violate the consistency (we have seen that a cut is consistent if and only if for every *receive* event in the cut the according *send* event is also in the cut). If we want to construct the largest *consistent* cut $C' \subseteq C$ from C, then we have no chance but to remove any such "unmatched" receive events (and all subsequent events) from C until there are no such violating event pairs anymore. By constructing C' this way, the resulting cut is consistent and there can not be any other cut the is larger, but also none that is equally large.

Proof: 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 output a set $C' \subseteq C$. It is a cut, because we have removed all events after a removed event. It is consistent because all event-pairs that violate consistency have been removed (compare for

lecture). It is maximal because if another event $e \in C \setminus C'$ is added to C', by construction of C' this event e is an receive event for which the corresponding send event $e' \notin C \supseteq C'$, thus $C' \cup \{e\}$ can not be consistent. It also is unique. For a contradiction, assume that C' and C'' are two different maximal consistent cuts preceding C (i.e., $|C'| = |C''|$). Then there is some event e in C' and $e \notin C''$. Then $C'' \cup \{e\} \cup \{e' \mid e' \Rightarrow_S e\}$ is a consistent cut preceding C and strictly larger than C'', a contradiction.

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.

Sample Solution

Intuition: We use the same approach as in the proof of the claim that $e \Rightarrow_S e'$ is equivalent to the statement that e precedes e' in all causal shuffles S' of S. In particular for one direction of the equivalence we showed that $e \not\Rightarrow_S e'$ implies that there is a causal shuffle S' of S where e' *precedes* e (contrapositive). The idea was that, given $e \not\Rightarrow_S e'$, if e precedes e' in S then e can be shifted by an amount of time Δ to occur after e' without violating causality. The generalisation to three events a, b, c as given above is then straight forward.

Proof: Assume a occurs before b or c in S (otherwise $S' := S$ already does the trick). Let τ be a logical clock for S that assigns each event e a point in time $\tau(e)$. Let $\Delta := max(\tau(b) - \tau(a), \tau(c) - \tau(a)) + 1$, i.e., the largest of the time differences between a and b or c (plus 1) respectively. We obtain a new logical clock τ' by shifting event a and all events e with $a \Rightarrow_S e$ by an amount Δ, i.e. $\tau'(e) := \tau(e) + \Delta$. This clock τ' defines a new schedule of events S' where b, c occur before a.

It remains to argue that the local views of S' and S are the same, i.e., that S' is in fact a causal shuffle. Let i be a node with local view $S|i$. Let $e, e' \in S|i$ be two events with $e \Rightarrow_S e'$. If e gets moved by Δ, then $a \Rightarrow_S e$. But then, by transitivity, we also have $a \Rightarrow_S e'$, so e' also gets moved by Δ, therefore the order of e and e' in $S'|i$ remains the same. If e does not get moved by Δ, then we do not care if e' gets moved or not, because the order for both events in $S'|i$ remains the same. Since this is now true for any two events $e, e' \in S|i$ with $e \Rightarrow_S e'$, we conclude that $S'|i = S|i$.

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.

Sample Solution

a) 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.

b) The largest clock value is at most four times the number of edges (at most two timer increases for each direction of an edge). This is happens 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) $= 2n(n - 1)$.