Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
Prof. Dr. Peter M. Fischer
Y. Maus
April 23, 2015

Distributed Systems, Summer Term 2015
Problem Set 1

The assignment is due on Tuesday, May 5, 2015, 14.15h. You can either hand it in electronically (*yannic.maus@cs.uni-freiburg.de*) or hand it in at the tutorial session itself.

Exercise 1: Schedules

Given are three nodes v_1, v_2 and v_3 which are connected via FIFO channels, that is, (two) messages, which are sent from some node i to the some node j, will arrive at node j in the order in which node i released the messages.
Devise **one** possible schedule S which is consistent with the following local restrictions to the three nodes.

- $S|1 = s_{1,3}\ s_{1,3}\ r_{1,2}\ r_{1,3}\ s_{1,2}\ r_{1,2}\ s_{1,3}$,
- $S|2 = s_{2,3}\ s_{2,1}\ r_{2,1}\ s_{2,1}$,
- $S|3 = r_{3,2}\ r_{3,1}\ s_{3,1}\ r_{3,1}\ r_{3,1}$.

$s_{i,j}$ denotes the send event from node i to node j and $r_{j,i}$ denotes the event that node j receives a message from node i.

Exercise 2: (Variations) of Two Generals

In the lecture we considered the (deterministically unsolvable) **Two Generals** consensus problem:

- two deterministic nodes, synchronuous communication, unreliable messages,
- **input**: 0 or 1 for each node,
- **output**: each node needs to decide either 0 or 1,
- **agreement**: both nodes must output the same decision (0 or 1),
- **validity**: if both nodes have the same input $x \in \{0, 1\}$ and no messages are lost, both nodes output x,
- **termination**: both nodes terminate in a bounded number of rounds.

In this exercise we consider three modifications of the model. For each of them, either give a (deterministic) algorithm or state a proof which shows that the variation cannot be solved deterministically.

a) There is the guarantee that within the first 7 rounds at least *one* message in *each* direction succeeds.

b) There is the guarantee that within the first 7 rounds at least *one* message succeeds.

c) Let $k \in \mathbb{N}$ be a natural number. The input for each node is a number $x_i \in \{0, \ldots, k\}$.

   Goal: If no message gets lost *and* both have the same input $x \in \{0, \ldots, k\}$, both have to output x. In all other cases the nodes should output numbers which do not differ by more than one. The algorithm still has to terminate in a finite number of rounds.

Hint: This problem is solvable. Ideas from the lecture might help.

Exercise 3: Asynchronous Message Passing, BFS Algorithm

In the lecture we introduced the distributed Bellman-Ford Algorithm for constructing a BFS tree in an asynchronous message passing system. The time complexity of the algorithm is $\mathcal{O}(D)$ and the message complexity is $\mathcal{O}(mn)$, where D is the diameter of the graph, n the number of nodes and m the number of edges.

The number of edges in any graph is in $\mathcal{O}(n^2)$, which implies an upper bound of $\mathcal{O}(n^3)$ on the number of messages. Find a graph and an execution of the algorithm in which $\Theta(n^3)$ messages are sent and explain your solution.