Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
Prof. Dr. Peter M. Fischer
Y. Maus
May 20, 2015

Distributed Systems, Summer Term 2015
Problem Set 3

The assignment is due on Tuesday, June 09, 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: Clock Synchronisation

Consider the following variation of the clock synchronisation algorithm $\mathcal{A}^{max}$ from the lecture. In the lecture (and in the last exercise), on the event of receiving a neighbors time, a node updates its local time to the maximum of its local time and the received time. The variation of the algorithm cannot directly update its time to the maximum, but it stores the maximal time it has received so far and increases the speed of its clock by a factor of two as long as its actual clock value is less than the stored maximum.
The maximal skew of $\mathcal{A}^{max}$ is $Dd(1+\rho)+2\rho T$, where D is the diameter of the network, d is the maximal message delay, T is the interval in which each clock value is at least forwarded once and all hardware clocks run at speeds in the interval $[1-\rho, 1+\rho]$ (note that we assumed $d=1$ in the lecture).
Show that the skew of the considered variation of $\mathcal{A}^{max}$ is at most $(Dd+T)(1+\rho)$.
Hint: A proof is similar to the proof for the skew of the clock synchronisation given in the lecture.

Exercise 2: Weak Agreement

You are given the following problem for two processes to find a weak agreement.

- Input/Output: Both processors have inputs and compute outputs from the set $\{0, 1\}$.
- Agreement: Possible outputs are $\{0, 1, *\}$, anything except the combination $\{0, 1\}$ is allowed.
- Validity: If both inputs are the same, they need to output this value in the case of no error.
- Termination: Non-failing processes need to terminate after a finite number of steps.

Design a wait-free algorithm/protocol that solves the given problem using only atomic read/write registers and prove the correctness of your algorithm (*hint:* there is a solution which uses only two registers).

Exercise 3: Consensus I

Alice and Bob live in the same town. Once a year they want to meet but they do not want to be seen together in public. So they want to meet at a secret place which one of them chooses. They know a wall in town which is painted white. In addition they know a painter who paints the wall in the color they wish and sends the person who gave him the order a *'before and after'* picture of the wall. (Of course they color-coded each possible meeting place with a single color in advance.)

1. Design an algorithm which ensures that Alice and Bob meet at the same place.

2. Can you expand your algorithm in such a way that it still works if Charlie wants to meet them as well?
3. How many persons can meet each other, if the wall is in front of the painters' shop and why? (You can assume that the painter immediately starts painting after receiving an order)

Exercise 4: Consensus II

A friend of yours is convinced to have found a great algorithm to solve consensus for 13 processes. His algorithm relies on a method called *'Fetch and Multiply'* which is described below. What would you tell him, if he asked you for your opinion?

```
public class RMW {
    private int value;
    public synchronized int FAM(int x) {
        int prior=this.value; prior = this.value;
        this.value=this.value*x;
        return prior;
    }
}
```