## Week 3

### H 4.4.4 dining philosophers (I)

Implement the solution as suggested in 4.4.4 such that there is no circular-waiting (take the standard 4.4 solution as your starting point).

### I 5.5 Santa Clause

Implement with slightly different requirement: with helping *at least* 3 elves, and reindeer don't have priority. Furthermore: the elf's getHelp() can only be executed together with Santa's helpElves() (so: an explicit synchronization has to be added which is not listed in the book)

Tip: please check first what goes wrong in the LBoS-solution when the reindeer/elf priority has been swapped.

Ensure that an arbitrary number of elf threads can be started (e.g. N=7)

### J 5.6 H2O

Implement without counters (but semaphores, mutexes, pipets, queues, barriers are allowed).

Ensure that an arbitrary number of H and O threads can be started (e.g. N=7).

### K 5.7 river crossing

Implement with a symmetric mutex usage (but counters, semaphores, mutexes, pipets, queues, barriers are allowed).

Ensure that an arbitrary number of hacker and serf threads can be started (e.g. N=7).

Tip: do not start with the code as given in LBoS and move some statements around until it more or less seems to work, but start with an empty sheet and write a clean implementation.

