# System Validation (IN4387) Final Examination
# January 30, 2014, 9:00–12:00

**Important Notes.** It is not allowed to use any study material, computers, or calculators during examination. The examination comprises 5 exercises in 5 pages. Please give complete explanations, and do not confine yourself to giving the final answer. A table with attainable scores can be found at the bottom of page 2. **Good luck!**

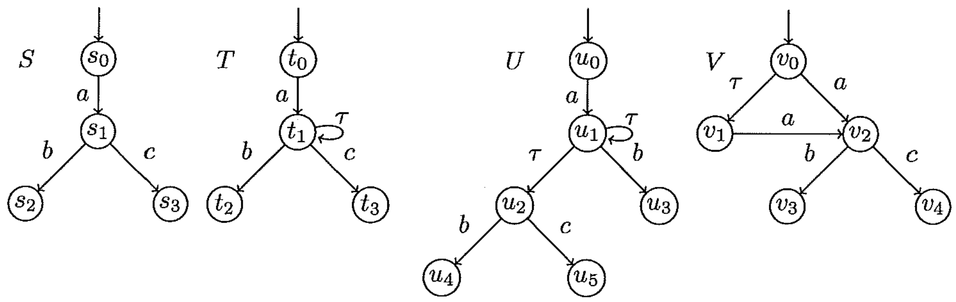
**Exercise 1.** Consider the labelled transition systems given below. Determine whether each of the following equivalences holds. For each and every item provide a complete line of reasoning why a certain equivalence does or does not hold:

(a) $S$ and $T$ are strongly bisimilar,

(b) $S$ and $U$ are branching bisimilar,

(c) $T$ and $U$ are trace equivalent,

(d) $T$ and $V$ are rooted branching bisimilar.




**Exercise 2.** Assume that the sort *Stack* of stacks over natural numbers is defined as follows:

| | |
|---|---|
| **sort** | *Stack*; |
| **cons** | *empty*: *Stack*; |
| | *push*: $Nat \times Stack \to Stack$; |
| **map** | *eq*: $Stack \times Stack \to Bool$; |

(a) Define equality *eq* on stacks. You may assume that *eq* on natural numbers is defined, and that standard Boolean operators may be used.

(b) Using your definition of *eq* prove that *empty* cannot be the same as $push(i, s)$ for arbitrary natural number $i$ and stack $s$.

(c) Define the operation *sum*, which takes a stack, and returns the sum of all values on the stack. You may assume that the operation *plus* on natural numbers is defined, and that *zero* is the constructor for natural number 0.

**Exercise 3.** Prove the following equations using the axioms provided in the appendix. At each step state precisely which axioms you have used.

(a) $((a \mid b) \setminus (b \mid a)) \cdot (c + c) = \tau \cdot c$.

(b) $(b + c) \mid\mid a = b \cdot a + c \cdot a + a \cdot (b + c) + a \mid b + a \mid c$.

(c) $\nabla_{r_1, c_2, s_3}(\Gamma_{s_2 \mid r_2 \to c_2}(r_1 \cdot (s_2 \mid r_2) \cdot s_3)) = r_1 \cdot c_2 \cdot s_3$

**Exercise 4.** Give an mCRL2 specification of a lock controller with the following informal specification. A lock is a device for raising and lowering boats between stretches of water of different levels on river and canal waterways, see Figure 1. The controller is supposed to control the entrance to a lock which can allow for at most one ship at a time. A ship announces its arrival using *arrive*$(i)$, where $i$ is the ship's identifier. If the lock is not occupied, and the waterlevel is down, the ship is allowed to enter the lock using *allow*$(i)$. If the lock if not occupied and the waterlevel is up, then first it is lowered using *lower*, before allowing the ship to enter. If the lock is occupied, the identifier of the ship will be recorded in the list of waiting ships. Upon departure of a ship from the lock, denoted by action *depart*, the first ship in the waiting list will be allowed into the lock. A ship can only depart if the waterlevel in the lock has been raised using the *raise* action. You may assume that the *raise* and *lower* actions take care of opening and closing the gates.

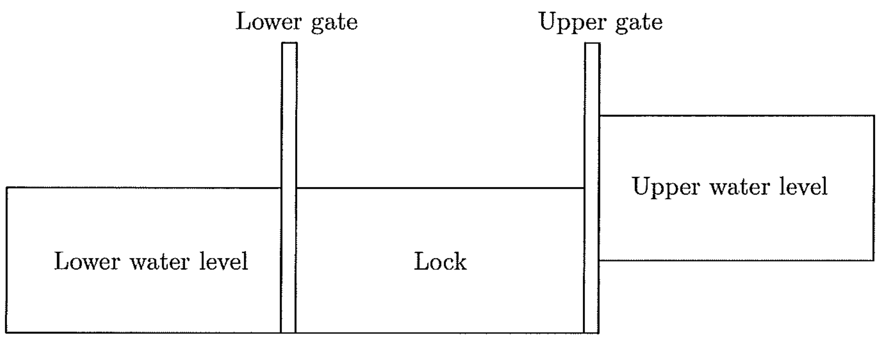



Figure 1: Situational sketch of the lock

**Exercise 5.** Express the following properties in the modal $\mu$-calculus. Assume that the set of actions $Act = \{a, b, c\}$.

(a) The system is deadlock free.

(b) Directly after each $a$ action in the initial state, a $b$ action cannot be done.

(c) Invariantly it is impossible to do two consecutive $c$ actions, without an intermediate $b$ action.

(d) Each $c$ action is inevitably followed by an $a$ action within a finite number of steps.

Score: $(10 + n)/10$ where $n$ is the cumulative judgement given by the following table:

| question | (a) | (b) | (c) | (d) | total |
|---|---|---|---|---|---|
| 1 | 5 | 5 | 5 | 5 | 20 |
| 2 | 3 | 7 | 5 | | 15 |
| 3 | 5 | 5 | 5 | | 15 |
| 4 | 20 | | | | 20 |
| 5 | 5 | 5 | 5 | 5 | 20 |