

VU University Amsterdam, 26th May 2014

At this exam, you may use copies of the slides without handwritten comments. Answers must be given in English.

The exercises in this exam sum up to 90 points; each student gets 10 points bonus.

Please note that 34 points are available for Question 5. The points available for each remaining question are as follows: Question 1 (8pts), Question 2 (17pts); Question 3 (19pts); and, Question 4 (12pts).

QUESTION 1. Given the sorts *Bool*, with the constructors $T, F : \rightarrow Bool$, and *Nat*, with constructors $0 : \rightarrow Nat$ and $S : Nat \rightarrow Nat$. Given also sort *D* with non-constructor $eq : D \times D \rightarrow Bool$ and the sort *List* over *D* with constructors $[] : \rightarrow List$ and $in : D \times List \rightarrow List$.

Specify a non-constructor $count : D \times List \rightarrow Nat$ that, given a data item and an arbitrary unordered list, returns the number of occurrences of the data item in the list.

You may use help functions, which must then also be defined. (8pts)

QUESTION 2. Consider the following μCRL specification:

$$\begin{aligned} Y(b : Bool) &= r(b).Z(\neg b).Y(b) \\ Z(b : Bool) &= s(b).Z(\neg b) + s(\neg b) \end{aligned}$$

(a) Linearise the specification, using the algorithm underlying **mcrl** (8pts)

(b) Does the algorithm underlying **mcrl -regular** terminate on this specification? If so perform the linearization, if not then justify your answer. (9pts)

QUESTION 3. (a) Specify, in μCRL , a process modelling a stack of size N , which can receive as input (i.e., push onto the stack) elements of some sort *D* via action name *rcv*, but only if the stack contains less than N data items. Also specify that the stack outputs the data items (i.e., pops from the stack) via action name *snd*; both *rcv* and *snd* target the top of the stack. Finally, specify an action *empty* that empties the stack only when it is non-empty. You are required only to specify the action declaration **act** and process declaration **proc**. (9pts)

(b) Give an abstraction of the stack, whereby the abstracted state space consists of two states, representing that the stack is empty and is non-empty, respectively. Your abstraction must contain three actions: a single action $\hat{r}$ to abstract all *rcv* actions; a single action $\hat{s}$ to abstract all *snd* actions; and, the action *empty*. (10pts)

QUESTION 4.

Consider the following convergent linear process equations. Assume non-constructors $\neg$ and $\vee$ have been specified for $Bool$ in the usual way, that the sort K has constructors $1, 2, 3 : \rightarrow K$ and that equality has been specified such that each constructor is unique.

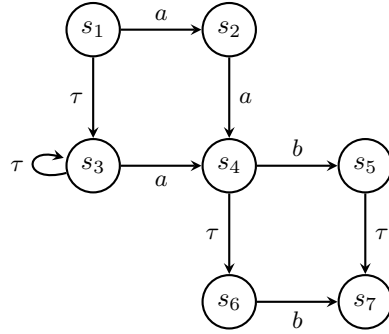
$$\begin{aligned} X(k : K) = & \\ & \tau.X(2) \triangleleft k = 1 \triangleright \delta \\ + & a.X(3) \triangleleft k = 1 \vee k = 2 \triangleright \delta \\ + & b.X(1) \triangleleft k = 3 \triangleright \delta \end{aligned}$$

$$\begin{aligned} Y(b : Bool) = & \\ & a.X(F) \triangleleft b \triangleright \delta \\ + & b.X(T) \triangleleft \neg b \triangleright \delta \end{aligned}$$

Prove they are branching bisimilar using the Cones & Foci method. To do so, first formulate the matching criteria. Then, give a state mapping $\phi : K \rightarrow Bool$ such that the matching criteria are satisfied with respect to X and Y . Finally, prove that the matching criteria are satisfied indeed. (12pts)

QUESTION 5.

Consider the following state space:



(a) Calculate for each of the following two formulas, which states in the state space satisfy the formula. (Provide also the intermediate steps of your calculation.) (8pts)

- (i) $\nu X.(\langle T \rangle X)$
- (ii) $\mu X.(\langle T \rangle T \wedge [\neg b] X)$

(b) State for the following formula, which states in the state space satisfy the formula. (Explain your answer) (4pts)

- (i) $\langle (\neg \tau)^* . b \rangle T$

(c) Apply the minimization algorithm modulo branching bisimilarity to this state space. Describe the subsequent splits that you perform, and the results of those splits. Also draw the resulting minimized state space. (12pts)

(d) What is the maximal confluent set in the original state space illustrated? (4pts)

(e) Again reduce the state space, but this time using τ -prioritisation. (4pts)

(f) Give an advantage and disadvantage of a reduction based on confluent τ -transitions in comparison to minimization modulo branching bisimulation? (2pts)