Exam Protocol Validation

Free University Amsterdam, 27 June 2008, 8:45-11:30

(At this exam, you may use copies of the slides without handwritten comments. Answers can be given in English or Dutch.)

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

1. Given a sort *List* over the natural numbers *Nat*, with as constructors the empty list $[] :\rightarrow List$ and $in : Nat \times List \rightarrow List$ to insert an element of *Nat* at the beginning of a list. Define the non-constructor function $add : Nat \times List \rightarrow List$, where $add(n, \ell)$ inserts element n into list ℓ, in such a way that if ℓ is a sorted list without duplicates, then $add(n, \ell)$ is again a sorted list without duplicates. (In this definition you may use help functions, which must then also be defined.) (12 pts)

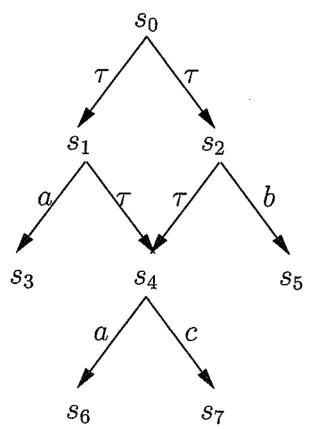
2. Consider a mutual exclusion algorithm, where actions $claim_i$ and $release_i$ mean that process p_i claims or releases the critical region. Express in the regular μ-calculus the following properties:

   (a) each occurrence of $release_1$ is preceded by an occurrence of $claim_1$; (4 pts)

   (b) each occurrence of $claim_1$ is eventually followed by an occurrence of $release_1$; (8 pts)

   (c) after an occurrence of $claim_1$, no $claim_2$ can happen until an occurrence of $release_1$. (4 pts)

3. Apply the minimisation algorithm modulo branching bisimulation to the process graph below. Describe the subsequent splits that you perform, and the results of those splits. Also draw the resulting minimized process graph.



(12 pts)

4. (a) Linearise, using the algorithm underlying `mcrl -regular`, the μCRL specification

$$\begin{aligned} Y(m{:}Nat) &= a(m)\cdot Z(S(m))\cdot Y(S(m)) \\ Z(m{:}Nat) &= b(m)\cdot Z(m) + c(S(m)) \end{aligned}$$

(8 pts)

(b) Linearise, using the algorithm underlying `mcrl`, the μCRL specification

$$\begin{aligned} Y(m{:}Nat) &= a(m)\cdot Z(S(m))\cdot Y(m) \\ Z(m{:}Nat) &= b(m)\cdot Z(S(m)) + c(m) \end{aligned}$$

with initial state $Y(0)$.

Would the algorithm underlying `mcrl -regular` terminate on this specification? (12 pts)

5. Let $even : Nat \to Bool$ with $even(n) = \mathsf{T}$ if and only if n is an even number: $even(0) = \mathsf{T}$ and $even(S(n)) = \neg(even(n))$.

(a) Consider the following LPE:

$$\begin{aligned} X(n : Nat) &= \tau\cdot X(n) \lhd even(n) \rhd \delta \\ &+ \; a(n)\cdot X(S(S(n))) \lhd \mathsf{T} \rhd \delta \end{aligned}$$

Prove that the confluence formula for the τ-summand is true, meaning that the τ-transitions generated by this summand are confluent. (9 pts)

(b) Consider the following LPE:

$$\begin{aligned} X(n : Nat) &= \tau\cdot X(S(n)) \lhd even(n) \rhd \delta \\ &+ \; a(n)\cdot X(S(S(n))) \lhd \mathsf{T} \rhd \delta \end{aligned}$$

Prove that in this case the confluence formula regarding the τ-summand is false. Also give an example consisting of transitions of $X(0)$ to show that the τ-summand does not give rise to confluent τ-transitions. (9 pts)

6. Given the state space of the μCRL specification $\partial_H(S(0) \parallel R(0) \parallel K \parallel L)$ of the Alternating Bit Protocol, where H consists of all send and read actions over the internal channels B, C, E and F. Let $\Delta = \{d, e\}$.

Consider the following abstraction. All communication actions over the internal channels and j are mapped to a special action $\hat{c}$, while the actions $r_A(d)$, $r_A(e)$, $s_D(d)$ and $s_D(e)$ are left in tact. The abstracted state space has three states: empty, d and e, where each state in the concrete state space is mapped to the datum that is currently being transported, or empty if no datum is being transported.

Give precise definitions of the mappings π and θ, and draw the state space of the abstracted ABP. (12 pts)