

2G1516 Formal Methods

REFERENCE SOLUTIONS
8 June 2006, 2pm–7pm

Mads Dam
KTH/IMIT

-
1. A simple solution alternates between two states and assigns a and $\neg b$ to the first state and $\neg a$ and b to the second. Formally:

15p

- $T = (Q, R, Q_0)$
- $Q = \{q_0, q_1\}$
- $R = \{(q_0, q_1), (q_1, q_0)\}$
- $Q_0 = \{q_0\}$
- $\eta(a) = \{q_0\}$
- $\eta(b) = \{q_1\}$

(Obs: η should have been ξ to correspond with the course notes.)

-
2. A suitable Buchi automaton A alternates between two states and has an a -labelled transition from the initial state to the alternate state, and both an a and a b transition from the alternate state back to the initial state. Any or both of the two states can be accepting. Formally:

15p

- $A = (Q, \Sigma, \Delta, I, F)$
 - $Q = \{q_0, q_1\}$
 - $\Sigma = \{a, b\}$
 - $\Delta = \{(q_0, a, q_1), (q_1, a, q_0), (q_1, b, q_0)\}$
 - $I = \{q_0\}$
 - $F = \{q_0\}$ (for instance)
-

3.

1. $\Box(a \vee b)$. The formula is valid since each state validates either a or b .
2. $aU(O(b \wedge \neg c))$. The formula is invalid. Consider the trace that always has a valid. This trace never visits a state which has $b \wedge \neg c$ valid.
3. $bU(\neg bU(c \wedge Oc))$. The formula is invalid. Consider the path π that visits states as follows:

$$a \wedge b \wedge c \rightarrow a \wedge \neg b \wedge \neg c \rightarrow a \wedge b \wedge c \rightarrow a \wedge \neg b \wedge \neg c \rightarrow \dots$$

For the formula to be true, since b alternates between the two states in the path, the formula $c \wedge Oc$ must hold in one of π^0 , π^1 or π^2 which it does not.

4. $A(bU(E(\neg bU(c \wedge AXc))))$. The formula is invalid. For the formula to be valid the state validating $\wedge \neg b \wedge \neg c$ must validate $E(\neg bU(c \wedge AXc))$, and for that to be the case, the initial state must validate $c \wedge AXc$, but this property fails.
5. $EFA((\neg b \wedge \neg c)Ub)$. The formula is valid. Consider any path that visits the state $a \wedge \neg b \wedge \neg c$ and some point i . That path validates b at point $b + 1$ as required.

-
4. Let C be the set of all programs. Let C^\dagger be the set of all nonterminating programs, i.e.

$$C^\dagger = \{c \mid \forall \sigma. \neg \exists \sigma'. (c, \sigma) \rightarrow \dots \rightarrow \sigma'\}.$$

1. $\phi = \psi = \text{true}$: Answer is C
2. $\phi = \text{true}, \psi = \text{false}$: Answer is C^\dagger
3. $\phi = \text{false}, \psi = \text{true}$: Answer is C
4. $\phi = \text{false}, \psi = \text{false}$: Answer is C

-
5. The transition diagram has three states, s_0, s_1, s_2 . s_0 is initial and s_2 is final. The transition relation is

$$R = \{(s_0, x > y \rightarrow x := x - 2, s_1), (s_1, y := y - 1, s_0), (s_0, x \leq y, s_2)\}$$

An assertion network assigns true to each node s_i . This network is trivially inductive and consistent. It is also deadlock-free. Define then the assignment w by:

$$\begin{aligned} w(s_0) &= x - y \\ w(s_1) &= x - (y - 1) \\ w(s_2) &= x - y \end{aligned}$$

For the progress conditions (Floyd-Hoare slides p. 15) condition 1 is a domain check, trivially valid. For condition 2 do the arithmetic check, and for condition 3 note that there is only one strongly connected subset = $\{s_0, s_1\}$, and that $x > y \rightarrow x - y > x - 2 - (y - 1) = x - y - 1$.

6. Leaving out symmetric cases:

15p

- P_1, P_2 : Observational congruence fails, since $P_1 + b$ and $P_2 + b$ are not weakly bisimilar.
- P_2, P_3 : Fails for same reason.
- P_1, P_3 : It suffices to show that $\tau.(b.0 + c.0)$ and $\tau.(b.0 + c.0) + b.0$ are weakly bisimilar. A suitable weak bisimulation relation is $\{(\tau.(b.0 + c.0), \tau.(b.0 + c.0) + b.0)\} \cup I$ where I is the identity relation.

7. Let:

15p

- $Chan_{1,2}^1 = Chan_{1,2}[out_1 \mapsto int_1, in_2 \mapsto int_2]$
- $Chan_{1,2}^2 = Chan_{1,2}[in_1 \mapsto int_1, out_2 \mapsto int_2]$

Here is a candidate 2-place channel:

$$Chan_{2,2} = (Chan_{1,2}^1 \mid Chan_{1,2}^2) \setminus \{int_1, int_2\}$$

After an in_1 -transition followed by an in_2 -transition, $Chan_{2,2}$ evolves into

$$(\overline{int_1}.Chan_{1,2}^1 \mid \overline{int_2}.Chan_{1,2}^2) \setminus \{int_1, int_2\}$$

which is deadlocked. Here is a variant of $Chan_{1,2}$ which is deadlock-free:

$$\begin{aligned} Chan_{1,2}^{0,0} &= in_1.Chan_{1,2}^{1,0} + in_2.Chan_{1,2}^{0,1} \\ Chan_{1,2}^{1,0} &= \overline{out_1}.Chan_{1,2}^{0,0} + in_2.Chan_{1,2}^{1,1} \\ Chan_{1,2}^{0,1} &= in_1.Chan_{1,2}^{1,1} + \overline{out_2}.Chan_{1,2}^{0,0} \\ Chan_{1,2}^{1,1} &= \overline{out_1}.Chan_{1,2}^{0,1} + \overline{out_2}.Chan_{1,2}^{1,0} \end{aligned}$$
