System Validation

Mohammad Mousavi

8. Model Examination with Solutions

System Validation (IN4387) Model Examination
October 24, 2012

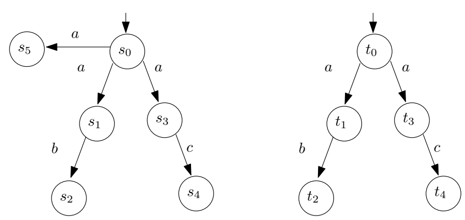
Important Notes. It is not allowed to use study material, computers, or calculators during the examination. The examination comprises 5 question and 3 pages. Please check beforehand whether your copy is properly printed. Give complete explanation and do not confine yourself to giving the final answer. The answers may be given in Dutch or in English. **Good luck!**

Exercise 1 (20 points) For each of the following items, give a pair of LTSs which are

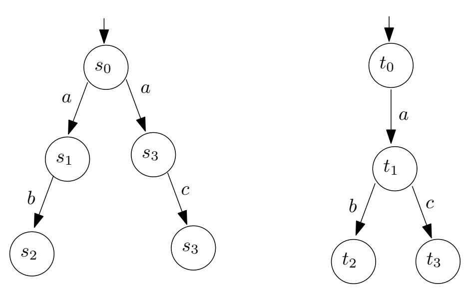
1. trace equivalent but not language equivalent,
2. language equivalent but not strongly bisimilar,
3. branching bisimilar but not strongly bisimilar, and
4. branching bisimilar but not rooted branching bisimilar.

Answer 1

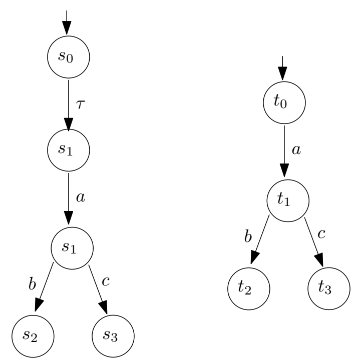
1.



2.



3. and 4.



Exercise 2 (20 points) Assume that the sort iStack of stacks of natural numbers defined below:

`sort`	iStack;
`cons`	empty: iStack;
	push: Nat # iStack $\rightarrow$ iStack;
`map`	eq: iStack $\times$ iStack $\rightarrow$ Bool;

1. Define the function eq on stacks with the above-given signature; when applied on two stacks, eq is equal to true if the stacks contain the same (possible empty) sequence of natural numbers. Assume that eq on natural numbers is already defined.

2. Prove, based on your definition of eq, that empty is different push(i,s) for each natural number i and iStack s.

Answer 2 1. The specification of eq is given below.

`var`	i,j: Nat;	
	s, sp: iStack;	
`eqn`	eq(s, s)= true;	(1)
	eq(empty, push(i, s))= false;	(2)
	eq(push(i, s), empty)= false;	(3)
	eq(push(i, s), push(j, sp))= eq(i,j) && eq(s, sp);	(4)

2. Assume towards a contradiction that empty = push(n, st) for some natural number n and some iStack st. Then, we have:

true	=	(1)
eq(empty, empty)	=	(assumption)
eq(empty, push(n, st))	=	(2)
false		

Exercise 3 (20 points) Prove the following equations using the axioms provided in the appendix.

1. $a \cdot c + b \cdot (c + \delta) = (a + b) \cdot c + a \cdot c$,

2. $a \parallel b = \delta \cdot (a \mid b) + a \cdot b + b \cdot a + a \mid b$,

3. $a \cdot \delta \parallel b = a \cdot b \cdot \delta + b \cdot a \cdot \delta + (a|b) \cdot \delta$,

4. $a \parallel (b + c) = (b + c) \cdot a + a \cdot (b + c) + a|b + a|c$.

Note that sequential composition binds stronger than nondeterministic choice.

Answer 3

1.

$$
\begin{array}{lcl}
a \cdot c + b \cdot (c + \delta) & = & \text{(A6)} \\
a \cdot c + b \cdot c & = & \text{(A3)} \\
(a \cdot c + a \cdot c) + b \cdot c & = & \text{(A2)} \\
a \cdot c + (b \cdot c + a \cdot c) & = & \text{(A2)} \\
(a \cdot c + b \cdot c) + a \cdot c & = & \text{(A1)} \\
a \cdot (c + b) + a \cdot c & = & \text{(A4)} \\
a \cdot (b + c) + a \cdot c & &
\end{array}
$$

2.

$$
\begin{array}{lcr}
a \parallel b & = & \text{(M)} \\
a \mathbin{\lfloor\!\lfloor} b + b \mathbin{\lfloor\!\lfloor} a + a \mid b & = & \text{(LM1)}\times 2 \\
a \cdot b + b \cdot a + a \mid b & = & \text{(A6,A1)} \\
\delta + a \cdot b + b \cdot a + a \mid b & = & \text{(A7)} \\
\delta \cdot (a \mid b) + a \cdot b + b \cdot a + a \mid b & &
\end{array}
$$

3.

$$
\begin{array}{lcr}
a \cdot \delta \parallel b & = & \text{(M)} \\
a \cdot \delta \mathbin{\lfloor\!\lfloor} b + b \mathbin{\lfloor\!\lfloor} a \cdot \delta + (a|b) \cdot \delta & = & \text{(LM3,LM1)} \\
a \cdot (\delta \parallel b) + b \cdot a \cdot \delta + (a|b) \cdot \delta & = & \text{(M)} \\
a \cdot (\delta \mathbin{\lfloor\!\lfloor} b + b \mathbin{\lfloor\!\lfloor} \delta + \delta|b) + b \cdot a \cdot \delta + (a|b) \cdot \delta & = & \text{(LM2,S1,S4)} \\
a \cdot (\delta + b \mathbin{\lfloor\!\lfloor} \delta + \delta) + b \cdot a \cdot \delta + (a|b) \cdot \delta & = & \text{(A6)} \times 2 \\
a \cdot (b \mathbin{\lfloor\!\lfloor} \delta) + b \cdot a \cdot \delta + (a|b) \cdot \delta & &
\end{array}
$$

4.

$$
\begin{array}{lcr}
a \parallel (b + c) & = & \text{(M)} \\
a \mathbin{\lfloor\!\lfloor} (b + c) + (b + c) \mathbin{\lfloor\!\lfloor} a + a|(b + c) & = & \text{(LM1)} \\
a \cdot (b + c) + b \mathbin{\lfloor\!\lfloor} a + c \mathbin{\lfloor\!\lfloor} a + a|(b + c) & = & \text{(LM1)} \times 2 \\
a \cdot (b + c) + b \cdot a + c \cdot a + a|(b + c) & = & \text{(S1,S7)} \\
a \cdot (b + c) + b \cdot a + c \cdot a + b|a + c|a & = & \text{(A4)} \\
a \cdot (b + c) + (b + c) \cdot a + b|a + c|a & = & \text{(S1)} \times 2 \\
a \cdot (b + c) + (b + c) \cdot a + a|b + a| & = & \text{(A1,A2} \\
(b + c) \cdot a + a \cdot (b + c) + a|b + a|c & &
\end{array}
$$

Exercise 4 (20 points) Give an mCRL2 specification of a traffic light controller. It starts of by showing the red signal, denoted by the action *show* with parameter *red* and remains showing it until it receives a signal that a car has arrived, denoted by the action *arrive*. Then, it moves it state to green and shows green, at some stage, it (nondeterministically) decides to change state to yellow, showing the yellow signal and later (again nondeterministically) decides to change state to red. It remains red until a new arrival is noticed.

Answer 4

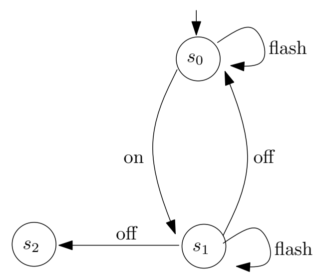
`sort` Color = `struct` red | yellow | green;

`act` show, moveTo : Color;
arrive ;

```
proc  TrLight (c: Color)  =
        show(c) · TrLight(c)
        arrive · (c ≈ red) → moveTo(green).TrLight(green) ◇ TrLight(c) +
        (c ≈ green) → moveTo(yellow).TrLight(yellow) +
        (c ≈ yellow) → moveTo(red).TrLight(red) ;

init  TrLight (red) ;
```

Exercise 5 (20 points) Consider the following LTS of a flash light.



Assume that the set *Act* is defined as $\{on, off, flash\}$. Specify which of the following properties are satisfied by the above-given LTS. Explain the reason.

1. $[off]\langle on\rangle true$
2. $[on][on]false$
3. $\langle Act\rangle true$
4. $\nu X.(\langle Act\rangle true \wedge [Act]X)$

Answer 5

1. The formula states that after all *off* transitions an *on* transition must be enabled. This formula trivially holds, because there is no *off* transition enabled initially.
2. The formula states that no two *on* transitions should be enabled in a row. This formula holds, because after the only enabled *on* transition there is no subsequent *on* transition enabled.
3. The formula states that some action must be enabled initially. The formula holds because initially both *on* and *flash* actions are enabled.
4. The formula states that an action must be enabled initially and after all other actions. In other words, this formula requires that no deadlock must exist. This formula does not hold because after performing an *on* transition there exists an *off* transition (from s_1 to s_2) after which no action is enabled.

A1	$x + y = y + x$
A2	$x + (y + z) = (x + y) + z$
A3	$x + x = x$
A4	$(x + y) \cdot z = x \cdot z + y \cdot z$
A5	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
A6	$x + \delta = x$
A7	$\delta \cdot x = \delta$
Cond1	$\mathit{true} \to x \diamond y = x$
Cond2	$\mathit{false} \to x \diamond y = y$
SUM1	$\sum_{d:D} x = x$
SUM3	$\sum_{d:D} X(d) = X(e) + \sum_{d:D} X(d)$
SUM4	$\sum_{d:D}(X(d) + Y(d)) = \sum_{d:D} X(d) + \sum_{d:D} Y(d)$
SUM5	$(\sum_{d:D} X(d)) \cdot y = \sum_{d:D} X(d) \cdot y$

Table 1: Axioms for the basic operators

M	$x \parallel y = x \mathbin{⌊\!⌊} y + y \mathbin{⌊\!⌊} x + x \vert y$
LM1	$\alpha \mathbin{⌊\!⌊} x = \alpha \cdot x$
LM2	$\delta \mathbin{⌊\!⌊} x = \delta$
LM3	$\alpha \cdot x \mathbin{⌊\!⌊} y = \alpha \cdot (x \parallel y)$
LM4	$(x + y) \mathbin{⌊\!⌊} z = x \mathbin{⌊\!⌊} z + y \mathbin{⌊\!⌊} z$
LM5	$(\sum_{d:D} X(d)) \mathbin{⌊\!⌊} y = \sum_{d:D} X(d) \mathbin{⌊\!⌊} y$
S1	$x \vert y = y \vert x$
S2	$(x \vert y) \vert z = x \vert (y \vert z)$
S3	$x \vert \tau = x$
S4	$\alpha \vert \delta = \delta$
S5	$(\alpha \cdot x) \vert \beta = \alpha \vert \beta \cdot x$
S6	$(\alpha \cdot x) \vert (\beta \cdot y) = \alpha \vert \beta \cdot (x \parallel y)$
S7	$(x + y) \vert z = x \vert z + y \vert z$
S8	$(\sum_{d:D} X(d)) \vert y = \sum_{d:D} X(d) \vert y$
TC1	$(x \mathbin{⌊\!⌊} y) \mathbin{⌊\!⌊} z = x \mathbin{⌊\!⌊} (y \parallel z)$
TC2	$x \mathbin{⌊\!⌊} \delta = x \cdot \delta$
TC3	$(x \vert y) \mathbin{⌊\!⌊} z = x \vert (y \mathbin{⌊\!⌊} z)$

Table 2: Axioms for the parallel composition operators