HW 6 – Modeling Systems

CS 477 – Spring 2014
Revision 1.1

Assigned April 16, 2014
Due April 23, 2014, 9:00 pm
Extension 48 hours (20% penalty)

1 Change Log

1.1 Added a solution to the extra credit problem.

1.0 Initial Release.

2 Objectives and Background

The purpose of this HW is to test your understanding of

- The use of program transition systems in describing systems.
- The use of LTL formulae in describing desired behavioral properties.

Another purpose of HWs is to provide you with experience answering non-programming written questions of the kind you may experience on the final.

3 Turn-In Procedure

The pdf for this assignment (`hw6.pdf`) should be found in the `assignments/hw6/` subdirectory of your svn directory for this course. Your solution should be put in that same directory. Using your favorite tool(s), you should put your solution in a file named `hw6-submission.pdf`. If you have problems generating a pdf, please seek help from the course staff. Your answers to the following questions are to be submitted electronically from within `assignments/hw6/` subdirectory by committing the file as follows:

```
svn add hw6-submission.pdf
svn commit -m "Turning in hw6"
```

4 Problems

1. (10 pts) This is a modeling problem. Consider how a simple lock on a waterway works. There are two gates, an upstream gate and downstream gate, and there are two valves (typically panels in the gates), an inlet valve on the upstream side and an outlet valve on the downstream side. When the upstream gate and the inlet valve are both closed, no water may flow from upstream into the lock, and when the downstream gate and outlet valve are both closed no water may flow out of the lock. If either the upstream gate or the upstream valve is open and the water level upstream is higher than the water level in the lock, then the water flows into the lock, increasing the level in the lock (one unit per time interval (an abstraction)), and if either the downstream gate or the downstream valve is open and the water level in the lock is higher than the water level downstream , then water flows out of the lock,

decreasing the level in the lock (one unit per time interval (an abstraction)). You may assume (an abstraction) that the water level upstream and downstream are constant with upstream always strictly higher than downstream.

The initial position of the lock is with the downstream gate open, and the upstream gate and both valves closed, and the water level equal to the downstream level. If a boat approaches from the upstream side (headed downstream), if the upstream gate is open, then the boat goes into the lock, while if the upstream gate is closed, it waits for it to open. If a boat is waiting on the upstream side (headed downstream) and the upstream gate is closed and the downstream gate is open, then the downstream gate closes, and when it is closed, the inlet valve opens. If the inlet valve is open and the water level in the lock is the same as the upstream water level, the inlet valve closes, and when it is closed, the upstream gate opens. When the upstream gate is open, if a boat is waiting on the upstream side (headed downstream), it enters the lock. Once a boat headed downstream has entered the lock, the gate closes and when it is closed, the outlet valve opens. If the outlet valve is open and the water level is equal to the downstream level, then the outlet valve closes, then the downstream gate opens, and then the boat exists the lock to the downstream side. A boat that is downstream of the lock headed downstream may either continue to head downstream or turn and head upstream approaching the lock from the downstream side.

When a boat approaches from the downstream side (headed upstream), if the downstream gate is open, the boat goes into the lock and if the downstream gate is closed, it waits for it to open. If a boat is waiting on the downstream side (headed upstream) and the downstream gate is closed and the upstream gate is open, then the upstream gate closes, and when it is closed the outlet valve opens. If the outlet valve is open and the water level in the lock is the same as the downstream water level, the outlet valve closes, and when it is closed, the downstream gate opens. When the downstream gate is open, if a boat is waiting on the downstream side (headed upstream), it enters the lock. Once a boat headed upstream has entered the lock, the gates close and when they are closed, the inlet valve opens. If the inlet valve is open and the water level is equal to the upstream level, then the inlet valve closes and the upstream gate opens. If there is a boat headed upstream in the lock and the upstream gate is open, then the boat exists the lock to the upstream side. A boat that is upstream of the lock headed upstream may either continue to head upstream or turn and head downstream approaching the lock.

Write a program transition system to model the boat, the water, lock gates and valves. You may assume just one boat exists, and that water levels are measured in integers. You will need to introduce a collection of variables, but their names should clearly indicate their meaning. For data, you may use the booleans true and false, and the integers, and any enumeration type you clearly define. Your program transition system should satisfy the properties listed in the next problem.

Solution:

$$F = \{\, \texttt{open}, \texttt{closed}, \texttt{upstream}, \texttt{downstream}, \texttt{in\_lock}, \texttt{upstream\_water\_level}, \texttt{downstream\_water\_level}, \texttt{+}, \texttt{-}, \texttt{1} \}$$

$$R = \{\texttt{=}, \texttt{>}\}$$

$$V = \{\, \texttt{inlet\_valve\_position}, \texttt{outlet\_valve\_position}, \texttt{upstream\_gate\_position}, \texttt{downstream\_gate\_position}, \texttt{boat\_position}, \texttt{boat\_direction}, \texttt{lock\_direction}, \texttt{lock\_water\_level}\}$$

$$\begin{aligned} init = \; & (\texttt{downstream\_gate\_position} = \texttt{open}) \wedge (\texttt{upstream\_gate\_position} = \texttt{closed}) \wedge \\ & (\texttt{inlet\_valve\_position} = \texttt{closed}) \wedge (\texttt{outlet\_valve\_position} = \texttt{closed}) \wedge \\ & (\texttt{lock\_water\_level} = \texttt{downstream\_water\_level}) \end{aligned}$$

$T = \{$ *Water behavior:*

$$\begin{aligned} & ((\texttt{upstream\_gate\_position} = \texttt{open}) \vee (\texttt{inlet\_valve\_position} = \texttt{open})) \wedge \\ & (\texttt{upstream\_water\_level} > \texttt{lock\_water\_level}) \\ & \to \texttt{lock\_water\_level} := \texttt{lock\_water\_level} + 1; \\ & ((\texttt{downstream\_gate\_position} = \texttt{open}) \vee (\texttt{outlet\_valve\_position} = \texttt{open})) \wedge \\ & (\texttt{lock\_water\_level} > \texttt{downstream\_water\_level}) \\ & \to \texttt{lock\_water\_level} := \texttt{lock\_water\_level} - 1; \end{aligned}$$

Boat behavior, headed downstream:

```
((boat_position = upstream) ∧ (boat_direction = downstream)∧
(upstream_gate_position = open))
→ (boat_position := in_lock);
((boat_position = in_lock) ∧ (boat_direction = downstream)∧
(downstream_gate_position = open))
→ (boat_position := downstream);
((boat_position = downstream) ∧ (boat_direction = downstream))
→ (boat_direction := upstream);
((boat_position = downstream) ∧ (boat_direction = downstream))
→ (boat_direction := downstream); − − −This is not necessary
```

Boat behavior, headed upstream:

```
((boat_position = downstream) ∧ (boat_direction = upstream)∧
(downstream_gate_position = open))
→ (boat_position := in_lock);
((boat_position = in_lock) ∧ (boat_direction = upstream)∧
(upstream_gate_position = open))
→ (boat_position := upstream);
((boat_position = upstream) ∧ (boat_direction = upstream))
→ (boat_direction := downstream);
```

Lock behvavior

```
((((boat_position = upstream) ∧ (boat_direction = downstream))∨
 ((boat_position = in_lock) ∧ (boat_direction = upstream)))∧
(downstream_gate_position = open))
→ (lock_direction, downstream_gate_position) := (upstream, closed);
((downstream_gate_position = closed) ∧ (upstream_gate_position = closed)∧
(outlet_valve_position = closed) ∧ (inlet_valve_position = closed)∧
(upstream_water_level > lock_water_level) ∧ (lock_direction = upstream))
→ inlet_valve_position := open;
((outlet_valve_position = closed) ∧ (inlet_valve_position = open)∧
(lock_water_level = upstream_water_level))
→ inlet_valve_position := closed;
((upstream_gate_position = closed) ∧ (inlet_valve_position = closed)∧
(lock_water_level = upstream_water_level) ∧ (lock_direction = upstream))
→ upstream_gate_position := open;
(((boat_position = in_lock) ∧ (boat_direction = downstream))∨
 (boat_position = downstream) ∧ (boat_direction = upstream))∧
(upstream_gate_position = open)
→ (lock_direction, upstream_gate_position) := (downstream, closed);
((downstream_gate_position = closed) ∧ (upstream_gate_position = closed)∧
(outlet_valve_position = closed) ∧ (inlet_valve_position = closed)∧
(lock_water_level > downstream_water_level ∧ (lock_direction = downstream))
→ outlet_valve_position := open;
((outlet_valve_position = open) ∧ (inlet_valve_position = closed)∧
(lock_water_level = downstream_water_level))
→ outlet_valve_position := closed;
((downstream_gate_position = closed) ∧ (outlet_valve_position = closed)∧
(lock_water_level = downstream_water_level) ∧ (lock_direction = downstream))
→ downstream_gate_position := open}
```

2. (10 pts) Using the variables and types you introduced in the previous problem, give LTL formulae expressing each

of the following:

1. (2pts) We never have both gates open at the same time.
   Solution: $\neg\Diamond((\texttt{downstream\_gate\_position} = \texttt{open}) \wedge (\texttt{upstream\_gate\_position} = \texttt{open}))$
2. (2pts) Neither valve is ever open when either gate is open.
   Solution: $\Box((\texttt{downstream\_gate\_position} = \texttt{open}) \vee (\texttt{upstream\_gate\_position} = \texttt{open})) \Rightarrow \neg((\texttt{outlet\_valve\_position} = \texttt{open}) \vee (\texttt{inlet\_valve\_position} = \texttt{open}))$
3. (3pts) The boat only enters the lock when the water level in the lock is the same as the water level on the side where the boat is.
   Solution: $\Box((((\texttt{boat\_position} = \texttt{upstream}) \wedge \circ(\texttt{boat\_position} = \texttt{in\_lock})) \Rightarrow (\texttt{lock\_water\_level} = \texttt{upstream\_water\_level})) \wedge (((\texttt{boat\_position} = \texttt{downstream}) \wedge \circ(\texttt{boat\_position} = \texttt{in\_lock})) \Rightarrow (\texttt{lock\_water\_level} = \texttt{downstream\_water\_level})))$
4. (3pts) A boat headed in a given direction will get to that side of the lock.
   Solution: $\Box((\texttt{boat\_direction} = \texttt{downstream}) \Rightarrow \Diamond(\texttt{boat\_position} = \texttt{downstream}))$

5 Extra Credit

3. (5 pts) Modify your program transition system in the first problem so that there are two boats. Your new system should still satisfy the properties in the second problem.

Solution:

$$F = \{\texttt{open}, \texttt{closed}, \texttt{upstream}, \texttt{downstream}, \texttt{in\_lock}, \texttt{upstream\_water\_level}, \texttt{downstream\_water\_level}, \texttt{+}, \texttt{-}, \texttt{1}\}$$

$$R = \{\texttt{=}, \texttt{>}\}$$

$$V = \{\texttt{inlet\_valve\_position}, \texttt{outlet\_valve\_position}, \texttt{upstream\_gate\_position}, \texttt{downstream\_gate\_position}, \texttt{boat\_position0}, \texttt{boat\_direction0}, \texttt{boat\_position1}, \texttt{boat\_direction1}, \texttt{lock\_direction}, \texttt{lock\_water\_level}\}$$

$$\begin{aligned} init = \ & (\texttt{downstream\_gate\_position} = \texttt{open}) \wedge (\texttt{upstream\_gate\_position} = \texttt{closed}) \wedge \\ & (\texttt{inlet\_valve\_position} = \texttt{closed}) \wedge (\texttt{outlet\_valve\_position} = \texttt{closed}) \wedge \\ & (\texttt{lock\_water\_level} = \texttt{downstream\_water\_level}) \end{aligned}$$

$T = \{$ *Water behavior:*

$((\texttt{upstream\_gate\_position} = \texttt{open}) \vee (\texttt{inlet\_valve\_position} = \texttt{open})) \wedge$
$(\texttt{upstream\_water\_level} > \texttt{lock\_water\_level})$
$\rightarrow \texttt{lock\_water\_level} := \texttt{lock\_water\_level} + 1;$

$((\texttt{downstream\_gate\_position} = \texttt{open}) \vee (\texttt{outlet\_valve\_position} = \texttt{open})) \wedge$
$(\texttt{lock\_water\_level} > \texttt{downstream\_water\_level})$
$\rightarrow \texttt{lock\_water\_level} := \texttt{lock\_water\_level} - 1;$

Boat 0 behavior, headed downstream:
$((\texttt{boat\_position0} = \texttt{upstream}) \wedge (\texttt{boat\_position0} = \texttt{downstream}) \wedge$
$(\texttt{upstream\_gate\_position} = \texttt{open}))$
$\rightarrow (\texttt{boat\_position0} := \texttt{in\_lock});$

$((\texttt{boat\_position0} = \texttt{in\_lock}) \wedge (\texttt{boat\_position0} = \texttt{downstream}) \wedge$
$(\texttt{downstream\_gate\_position} = \texttt{open}))$
$\rightarrow (\texttt{boat\_position0} := \texttt{downstream});$

$((\texttt{boat\_position0} = \texttt{downstream}) \wedge (\texttt{boat\_position0} = \texttt{downstream}))$
$\rightarrow (\texttt{boat\_position0} := \texttt{upstream});$

Boat 0 behavior, headed upstream:
$((\texttt{boat\_position0} = \texttt{downstream}) \wedge (\texttt{boat\_position0} = \texttt{upstream}) \wedge$
$(\texttt{downstream\_gate\_position} = \texttt{open}))$
$\rightarrow (\texttt{boat\_position0} := \texttt{in\_lock});$

$((\texttt{boat\_position0} = \texttt{in\_lock}) \wedge (\texttt{boat\_position0} = \texttt{upstream}) \wedge$
$(\texttt{upstream\_gate\_position} = \texttt{open}))$
$\rightarrow (\texttt{boat\_position0} := \texttt{upstream});$

$((\texttt{boat\_position0} = \texttt{upstream}) \wedge (\texttt{boat\_position0} = \texttt{upstream}))$
$\rightarrow (\texttt{boat\_position0} := \texttt{downstream});$

Boat 1 behavior, headed downstream:
$((\texttt{boat\_position1} = \texttt{upstream}) \wedge (\texttt{boat\_direction1} = \texttt{downstream}) \wedge$
$(\texttt{upstream\_gate\_position} = \texttt{open}))$
$\rightarrow (\texttt{boat\_position1} := \texttt{in\_lock});$

$((\texttt{boat\_position1} = \texttt{in\_lock}) \wedge (\texttt{boat\_direction1} = \texttt{downstream}) \wedge$
$(\texttt{downstream\_gate\_position} = \texttt{open}))$
$\rightarrow (\texttt{boat\_position1} := \texttt{downstream});$

$((\texttt{boat\_position1} = \texttt{downstream}) \wedge (\texttt{boat\_direction1} = \texttt{downstream}))$
$\rightarrow (\texttt{boat\_direction1} := \texttt{upstream});$

Boat 1 behavior, headed upstream:
$((\texttt{boat\_position1} = \texttt{downstream}) \wedge (\texttt{boat\_direction1} = \texttt{upstream}) \wedge$
$(\texttt{downstream\_gate\_position} = \texttt{open}))$
$\rightarrow (\texttt{boat\_position1} := \texttt{in\_lock});$

$((\texttt{boat\_position1} = \texttt{in\_lock}) \wedge (\texttt{boat\_direction1} = \texttt{upstream}) \wedge$
$(\texttt{upstream\_gate\_position} = \texttt{open}))$
$\rightarrow (\texttt{boat\_position1} := \texttt{upstream});$

$((\texttt{boat\_position1} = \texttt{upstream}) \wedge (\texttt{boat\_direction1} = \texttt{upstream}))$
$\rightarrow (\texttt{boat\_direction1} := \texttt{downstream});$

Raising the lock level, closing the downstream gate
((((((boat_position0 = upstream) ∧ (boat_direction0 = downstream))∨
((boat_position0 = in_lock) ∧ (boat_direction0 = upstream)))∧
((boat_position1 ≠ downstream) ∨ (boat_direction1 = downstream))∧
((boat_position1 ≠
in_lock) ∨ (boat_direction1 = upstream)))∨
((((boat_position1 = upstream) ∧ (boat_direction1 = downstream))∨
((boat_position1 = in_lock) ∧ (boat_direction1 = upstream)))∧
((boat_position0 ≠ downstream) ∨ (boat_direction0 = downstream))∧
((boat_position0 ≠
in_lock) ∨ (boat_direction0 = upstream)))) ∧ (downstream_gate_position = open))
→ (lock_direction, downstream_gate_position) := (upstream, closed);

Opening the inlet valve
((downstream_gate_position = closed) ∧ (upstream_gate_position = closed)∧
(outlet_valve_position = closed) ∧ (inlet_valve_position = closed)∧
(upstream_water_level > lock_water_level) ∧ (lock_direction = upstream))
→ inlet_valve_position := open;

Closing inlet valve((outlet_valve_position = closed) ∧ (inlet_valve_position = open)∧
(lock_water_level = upstream_water_level))
→ inlet_valve_position := closed;

Opening the upstream gate((upstream_gate_position = closed) ∧ (inlet_valve_position = closed)∧
(lock_water_level = upstream_water_level) ∧ (lock_direction = upstream))
→ upstream_gate_position := open;

Lowering the lock level, closing the upstream gate
((((((boat_position0 = downstream) ∧ (boat_direction0 = upstream))∨
((boat_position0 = in_lock) ∧ (boat_direction0 = downstream)))∧
((boat_position1 ≠ upstream) ∨ (boat_direction1 = upstream))∧
((boat_position1 ≠
in_lock) ∨ (boat_direction1 = downstream)))∨
((((boat_position1 = downstream) ∧ (boat_direction1 = upstream))∨
((boat_position1 = in_lock) ∧ (boat_direction1 = downstream)))∧
((boat_position0 ≠ upstream) ∨ (boat_direction0 = upstream))∧
((boat_position0 ≠
in_lock) ∨ (boat_direction0 = downstream)))) ∧ (upstream_gate_position = open))
→ (lock_direction, upstream_gate_position) := (downstream, closed);

Opening the outlet valve
((upstream_gate_position = closed) ∧ (downstream_gate_position = closed)∧
(outlet_valve_position = closed) ∧ (inlet_valve_position = closed)∧
(lock_water_level > downstream_water_level) ∧ (lock_direction = downstream))
→ outlet_valve_position := open;

Closing the outlet valve
((outlet_valve_position = open) ∧ (inlet_valve_position = closed)∧
(lock_water_level = downstream_water_level))
→ outlet_valve_position := closed;

Opening the downstream gate
((downstream_gate_position = closed) ∧ (outlet_valve_position = closed)∧
(lock_water_level = downstream_water_level) ∧ (lock_direction = downstream))
→ downstream_gate_position := open}