**CSEP590 – Model Checking and Software Verification**
**Summer 2003**
**Solution Set 5**

1. Transition system M with fairness constraints
**Solutions:**
a) Maximal SCCs

Omitting trivial SCCs, we have two maximal SCCs:
i) $S_1, S_4, S_8$
ii) $S_2, S_5, S_6, S_7, S_9, S_{10}, S_{11}, S_{12}$

If we include trivial SCCs, then there are four (add $S_0$ and $S_3$).

b) Does M, $S_0 \models E_c Gp$ hold?

Holds:
First, p is true in all states by definition.
The SCC $S_2, S_5, S_6, S_7, S_9, S_{10}, S_{11}, S_{12}$ satisfies all the fairness constraints.
We can reach the latter SCC from state $S_0$.
Thus, M, $S_0 \models E_c Gp$ holds.

c) p false in $s_6$. Does M, $S_0 \models E_c Gp$ hold?

Does not hold:
Now there are three SCCs in the system: $S_2, S_5$; $S_7, S_{11}, S_{12}$; $S_1, S_4, S_8$
None of these SCCs satisfy the fairness constraints, so there are no fair SCCs.
Thus, M, $S_0 \models E_c Gp$ does not hold.

2. The Spring Kripke structure
**Solutions:**
LTL formulae and run $\pi = S_1\ S_2\ S_1\ S_2\ S_3\ S_3\ S_3 \ldots$
a) extended
Not satisfied because 'extended' is false in $S_1$.

b) X extended
Satisfied because 'extended' is true at $S_2$.

c) XX extended
Not satisfied because 'extended' is false in $S_1$.

d) F extended
Satisfied because 'extended' is true in some future state (e.g. $S_2$).

e) G extended
Not satisfied because 'extended' is not true at all states in the path (e.g. $S_1$).

f) FG extended
Satisfied because after we reach $S_3$, 'extended' is true globally.

g) !extended U malfunction
Not satisfied because 'extended' is not continuously false until 'malfunction' is true – since 'extended' is true in $S_2$.

LTL formulae and system Spring
Note: *by definition, a path $\pi$ for LTL must be infinite.*

a) F extended
Satisfied because from $S_1$ we must move to a state were extended is true in all paths.

b) G(!extended $\Rightarrow$ X extended)
Satisfied because !extended is true only at $S_1$, and the next state after $S_1$ in all paths must be $S_2$, at which extended holds.

c) FG extended
Not satisfied because in path $S_1 \rightarrow S_2 \rightarrow S_1 \rightarrow S_2$ … is it never the case that G extended is true.

d) !FG extended
Not satisfied because in path $S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_3 \rightarrow S_3$ … G extended is satisfied.

e) G(extended $\Rightarrow$ X !extended)
Not satisfied, look at path $S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_3 \rightarrow S_3$ …

3. LTL equivalences
**Solutions:**
a) XFp, FXp
Equivalent

b) (FGp) $\wedge$ (FGq), F(Gp $\wedge$ Gq)
Equivalent

c) (p U q) U q, p U q
Equivalent

d) (p U q) $\wedge$ (q U r), (p U r)
Not equivalent. This is so because the first formula requires that q hold at some point in the path, while the second formula does not. Thus, a counterexample structure is:

Let p be true in state $S_1$, r true in state $S_2$, and q true in state $S_3$
Then $S_1 \rightarrow S_1 \rightarrow S_1 \rightarrow S_2$ satisfies the second formula but not the first.

e) Find a CTL* path formula
$A[G(p \rightarrow X \neg p)\ \ G(\neg p \rightarrow Xp)]$

4. Monotone functions and fixed points
**Solutions:**
a) Which are monotone?
H1 is monotone

Its is clear to see that if $X_1 \subseteq X_2$, then $H_1(X_1) \subseteq H_1(X_2)$ because $H_1$ will have removed the same elements from *both* $X_1$ and $X_2$, thereby maintaining their relationship.

H2 is not monotone

Counterexample:
Let $X_1 = \{2\}$, $X_2 = \{2,5\}$, then $X_1 \subseteq X_2$, but
$H_2(X_1) = \{5,9\}$, $H_2(X_2) = \{9\}$, so $H_2(X_1)$ $!\subseteq$ $H_2(X_2)$

H3 is monotone

A union with a larger set cat either make the intersection larger, or make no change in size – it will never reduce the size. Thus the relation between $X_1$ and $X_2$ is maintained.

b) Greatest and least fixed points of $H_3$

Least fixed point: {2,4}
Greatest fixed point: {1,2,3,4,5}

c) Fixed points of $H_2$?

No fixed points because if the input contains any of {2,5,9}, those elements will be removed, resulting in a different output. Likewise, if the input does not contain any of {2,5,9}, then the output will be {2,5,9} which will be different than the input.

5. Relational mu-calculus
**Solution:**
We need to prove p |= vZ.Z so here f = Z

We use induction on m
Base case m =0
$v_0Z.f = 1$ (by definition)

By mu-calculus grammar definition, p |= 1 so the base case holds.

Suppose this holds for m = p.

$p \models V_pZ.Z$

Then we compute $V_{p+1}Z.Z$

$V_{p+1}Z.Z = VpZ.Z$ (replace Z with $V_{p+1-1}Z.Z$ according to vZ.f definition.)

But we know from the induction hypothesis that p |= VpZ.Z. So p |= $V_{p+1}Z.Z$
And by induction for all m >= 0, p |= vZ.Z