CS 476 Homework #12 Due 10:45am on 12/5

Note: Answers to the exercises listed below should be handed to the instructor *in hardcopy* and in *typewritten form* (latex formatting preferred) by the deadline mentioned above. You should also email to `hildenb2@illinois.edu` the Maude code for the second exercise.

1. Answer the following questions:

   (a) Let $AP = \{p, q\}$. Explain — by giving in each case a counterexample trace $\tau \in [\mathbb{N} \to \mathcal{P}(AP)]$ such that, for ϕ the chosen wrong formalization, either $\tau \models p \rightsquigarrow q$ and $\tau \not\models \phi$ or $\tau \not\models p \rightsquigarrow q$ and $\tau \models \phi$ — why each of the following LTL formalizations ϕ of the "leads to" property $p \rightsquigarrow q$ is wrong:
   - $p \to q$
   - $\Box(p \to q)$

   (b) Write an LTL formula stating that for any path from the given initial state at only a finite number (possibly zero) of positions (i.e., steps) in such a path the property φ holds.

   (c) Write an LTL formula stating that for any path from the given initial state whenever φ holds at some step n, then it will also hold at all steps $n + 2k$ for any $k \in \mathbb{N}$.

   (d) Write an LTL formula stating that for any path from the given initial state, whenever φ holds at some step n, property ϕ will then hold at a strictly later step after that.

   (e) Write an LTL formula stating that for any path from the given initial state, whenever φ holds at some step n then ϕ does not hold at step n but will hold at a strictly later step after that.

   (f) Let $AP = \{p, q\}$. Define traces $\tau_1, \tau_2 \in [\mathbb{N} \to \mathcal{P}(AP)]$ such that, instantiating φ to p, and ϕ to q, τ_1 satisfies property (d) but not property (e), and τ_2 satisfies properties (d) and (e).

2. The problem below is based on using the Maude Reachability Logic Prover. For this problem, we have already included a template file located in the tool distribution that you can access from the course website. If the root of the tool distribution is the path `~/rltool` the template file is located at `~/rltool/ex/rw/rw-prob.maude`. You are asked to edit the template file in the designated places. Note: the template will take care of loading the tool for you; you can just specify the commands as we discussed in class. Finally, the distribution contains the two examples used in Lecture 24 in folders `~/rltool/ex/choice` and `~/rltool/2token`.

   In this problem, you are asked to verify that the module `R&W` below satisfies a specific invariant from an initial state. The initial state we will consider is `< 0, 0 >`, i.e. there are no readers or writers. The invariant we will consider is the conjunction of the `mutex` and (at most) `one-writer` predicates defined in Lecture 18. This invariant can be specified by the set of two patterns: `< N:Nat , 0 >` and `< 0 , s(0) >`. To carry out the proof, you should use the **Corollary** for proving invariants discussed in Lectures 23 and 24.

```
mod R&W is
  sorts Nat Config .
  op 0 : -> Nat [ctor] .
  op s : Nat -> Nat [ctor] .
  op <_,_> : Nat Nat -> Config [ctor] . --- readers/writers
  op [_,_] : Nat Nat -> Config [ctor] . --- terminating states
  op done  : -> Config .
  vars R W : Nat .
  rl [write+] : < 0    , 0    > => < 0    , s(0) > .
  rl [write-] : < R    , s(W) > => < R    , W    > .
  rl [read+ ] : < R    , 0    > => < s(R) , 0    > .
  rl [read- ] : < s(R) , W    > => < R    , W    > .
endm
```