CS 476 Homework #12 Due 10:45am on 4/16

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 `fanyang6@illinois.edu` the Maude code for both exercises.

1. Given a concurrent system specified by a rewrite theory $\mathcal{R}$, recall that an *invariant* is specified as a Boolean-valued predicate $I : State \to Bool$ with equations protecting `BOOL`, where *State* is the chosen sort of states in $\mathcal{R}$. This can be generalized to the notion of a *parametric invariant*, as a Boolean-valued function $I : State\ A_1 \ldots A_n \to Bool$, where $A_1 \ldots A_n$ are sorts in the signature of $\mathcal{R}$, and the equations defining I again protect `BOOL`. Then, in $I(S, x_1, \ldots, x_n)$ the $x_1 : A_1, \ldots, x_n : A_n$ are called the *data parameters* of the invariant.

   Let now $init(x_1 : A_1, \ldots, x_n : A_n)$ be a term of sort *State* whose only variables are the $x_1 : A_1, \ldots, x_n : A_n$. We then say that the parametric invariant $I(S, x_1, \ldots, x_n)$ *holds* in the initial reachability model $\mathcal{T}_{\mathcal{R}}$ for the parametric family of initial states $init(x_1 : A_1, \ldots, x_n : A_n)$ with data parameters $x_1 : A_1, \ldots, x_n : A_n$ if an only if for each ground substitution $\rho = \{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$ of the variables $x_1 : A_1, \ldots, x_n : A_n$ the (unparametric) invariant $I(S, u_1, \ldots, u_n)$ holds from the (ground) initial state $init(u_1, \ldots, u_n)$.

   Consider now the following unordered communication channel between a sender and a receiver:

```
mod COMM-CHANN is protecting NAT .  protecting QID .
  sorts Msg MsgMSet QidList State .
  subsort Msg <  MsgMSet .
  subsort Qid < QidList .

  op nil : -> QidList [ctor] .
  op _;_ : QidList QidList -> QidList [ctor assoc id: nil] .
  op [_,_] : Qid Nat -> Msg [ctor] .
  op null : -> MsgMSet [ctor] .
  op _ _ :  MsgMSet MsgMSet -> MsgMSet [ctor assoc comm id: null] .
  op [_:_|_|_:_] : QidList Nat MsgMSet Nat QidList -> State [ctor] .

  vars N M I J : Nat . var MS : MsgMSet . vars L L' : QidList .
  vars A B : Qid .

  rl [send] : [A ; L : I | MS | J : L'] => [L : s(I) | [A,I] MS | J : L'] .
  rl [receive] : [L : I | MS [A,J] | J : L'] => [L : I | MS | s(J) : L' ; A] .
endm
```

   where both the sender (on left) and the receiver (on right) hold buffers (lists) with lists of Qids to be sent (resp. already received), and the channel (in the middle) is a multiset of messages. We assume both counters will initially be at 0, and that the receiver's buffer will initially hold the `nil` list of Qids.

   Intuitively, the counters are used to ensure *in order reception*. That is, even though the channel is a multiset so that the messages *can get out of order*, we expect and desire that, *at any given time* during the sending and receiving process, the list of Qids in the receiver's buffer will be arranged *in the exact same order* in which they were initially in the sender's buffer, although, of course, only at the end of the sending and receiving process will the receiver hold the *entire list* sent by the sender.

   You are asked to do the following in a module `COMM-CHAN-PREDS` protecting `BOOL` and `COMM-CHAN`:

- Define the above property of in-order-reception as a parametric invariant of the form:

  ```
  op in-order-reception : State QidList -> Bool .
  ```

  Hint: the use of Maude's `owise` feature can make the definition easier.

- Specify the parametric family $init(L : QidList)$ of initial states for which `in-order-reception(L)` is an invariant for `init(L)` as an equationally-defined function:

  ```
  op init : QidList -> State .
  ```

- Show a screenshot of your results for verifying the parametric invariant when the parameters L is instantiated to the Qid lists:

  ```
  'a ; 'b ; 'c

  'a ; 'b ; 'c ; 'd

  'a ; 'b ; 'c ; 'd ; 'e
  ```

2. Using the same `COMM-CHANN` module of Exercise 1, define equationally in a module `COMM-CHAN-PREDS` protecting `BOOL` and `COMM-CHAN`:

   (a) a parameterized invariant

   ```
   op final : State QidList -> Bool .
   ```

   such that `final(S,L)` is true iff state `S` is such that the sender's buffer is empty, the channel is empty, the receiver's buffer contains the list `L`, and the sender and receiver counters both have the value `lenght(L)` (you will need to define `lenght` as an auxiliary function).

   (b) an unparameterized invariant

   ```
   op enabled : State -> Bool .
   ```

   such that `enabled(S)` is true iff state `S` can be rewritten by one of the rules in the `COMM-CHANN` module.

   Note that then `final(S,L) or enabled(S)` is also an invariant parametric on `L`. Show a screenshot of your results for verifying the parametric invariant `final(S,L) or enabled(S)` from initial state `init(L)` when the parameters `L` is instantiated to the Qid lists:

   ```
   'a ; 'b ; 'c

   'a ; 'b ; 'c ; 'd

   'a ; 'b ; 'c ; 'd ; 'e
   ```