Verification of Cyber-physical Systems : Exercise Sheet 3

Deadline : Monday 16th October 2017, 11 :55 pm

Exercise 1

Consider the following *Promela* model of MutEx. From it, create a new correct model of mutual exclusion (for two processes as well), with the following properties :

- There should be only one generic Promela process defined.
- Replace the assertion checks with a never claim.
- You can replace the x and y variables with an array of 2 elements.

```
#define true 1
#define false 0
#define Aturn false
#define Bturn true

bool x,y,t;
byte count;

proctype A(){
        startA:
        x = true;
        t = Bturn;
        (y == false || t == Aturn);
        count++;
        assert(count <= 1);      /*critical section*/
        count--;
        x = false;
        goto startA;
}

proctype B(){
        startB:
        y = true;
        t = Aturn;
        (x == false || t == Bturn);
        count++;
        assert(count <= 1);      /*critical section*/
        count--;
        y = false;
        goto startB;
}

init{ atomic{ run A(); run B(); } }
```

Exercise 2

With your model from Exercise 1, remove the never claim and check your model with an *LTL* formula. To do that, translate your *LTL* formula into a never claim (using *Spin* in command line) and copy the result in your model.

Exercise 3

Suppose that we consider a model that contains the following never claim, where p is some atomic proposition.

```
never{
        T0_init:
        if
                :: p -> goto accept_S1;
                :: true -> goto T0_init;
        fi;
        accept_S1:
        if
                :: p -> goto accept_S1;
                :: true -> goto T0_init;
        fi;
}
```

Describe what we want that the model verify.

Submit your commented `.pml` files on *Ilias* and also paste it in your PDF submission.