

Assignment

Diego Garca-Olano, Rajagopal Hariharan, Roberto E. Vargas Caballero

November 15, 2013

- 7.7 The LTS definition of the problem is:

Listing 1: Field LTS definition

```
const True = 1
const False = 0
range Bool = False .. True
range Card = 1 .. 2
set BoolActions = { setTrue, setFalse, [False], [True] }
set CardActions = { set1, set2, [1],[2] }

BOOLVAR = VAL[False],
VAL[v:Bool] = ( setTrue -> VAL[True]
               | setFalse -> VAL[False]
               | [v] -> VAL[v]
               ).

||FLAGS = (flag1:BOOLVAR || flag2:BOOLVAR ).

CARDVAR = VAL[1],
VAL[i:Card] = ( set1 -> VAL[1]
               | set2 -> VAL[2]
               | [i] -> VAL[i]
               ).

NEIGHBOR1 = (flag1.setTrue->t.set2->TEST),
TEST = (flag2[raised:Bool] ->
        t[i:Card] ->
        if(raised && i == 2) then (wait->TEST )
        else (enter->exit->flag1.setFalse->NEIGHBOR1)
        )+{{flag1,flag2}.BoolActions,t.CardActions}.

NEIGHBOR2 = ( flag2.setTrue -> t.set1-> TEST),
TEST = ( flag1[raised:Bool] ->
        t[i:Card] ->
        if(raised && i == 1) then (wait->TEST)
        else (enter->exit->flag2.setFalse->NEIGHBOR2)
        )+{{flag1,flag2}.BoolActions,t.CardActions}.

progress ENTER1 = { alice.enter }
```

```

progress ENTER2 = { bob.enter }
property MUTEX = ( alice.enter -> alice.exit -> MUTEX
                  | bob.enter -> bob.exit -> MUTEX ).

||FIELD = ( alice:NEIGHBOR1 || bob:NEIGHBOR2 ||
            {alice,bob}::FLAGS
            || {alice,bob}::t:CARDVAR || MUTEX ).
||GREEDY_FIELD = FIELD << { alice.enter, bob.enter }.

```

- 7.8 The implementation of the stressed Field is:

Listing 2: Field class

```

public class Field {
    public static void main(String[] args) {
        Turn turn = new Turn();
        Flag AliceF = new Flag(Neighbor.Alice);
        Flag BobF = new Flag(Neighbor.Bob);
        Neighbor Bob = new Neighbor(Neighbor.Bob,
                                    Neighbor.Alice,
                                    BobF,
                                    AliceF,
                                    turn);
        Neighbor Alice = new Neighbor(Neighbor.Alice,
                                      Neighbor.Bob,
                                      AliceF,
                                      BobF,
                                      turn);

        Bob.start();
        Alice.start();
    }
}

```

Listing 3: Neighbor class

```

public class Neighbor extends Thread {
    int me, you;
    Flag[] flags;
    Turn turn;
    String name;
    public static int Alice = 0, Bob = 1;
    private static int inside = 0;

    public Neighbor(int me, int you, Flag myflags, Flag
yourflags, Turn turn) {
        this.turn = turn;
        this.me = me;
        this.you = you;
        flags = new Flag[2];
        flags[me] = myflags;
        flags[you] = yourflags;
        name = name(me);
    }
}

```

```

public static String name(int guy) {
    return (guy == Bob) ? "Bob" : "Alice";
}

public void run() {
    for (;;) {
        doit();
    }
}

private void doit() {
    System.out.println("Me=" + me + ",You=" +
you);
    flags[me].set(true);
    turn.set(you);

    while (flags[you].get() && turn.get() == you)
        ;

    enter();
    exit();
    flags[me].set(false);
}

private void enter() {
    if (inside != 0) {
        System.out.println("Critical region
violated");
        System.exit(0);
    }
    inside = 1;
    System.out.println(name + " enters in
garden");
}

private void exit() {
    inside = 0;
    System.out.println(name + " exits of garden");
}
}

```

Listing 4: Flag class

```

public class Flag {
    boolean value = false;
    int who;

    public Flag(int who) {
        this.who = who;
    }

    void set(boolean val) {
        value = val;
        System.out.println("Flag " +
Neighbor.name(who) + " " + value);
    }
}

```

```
        boolean get() {  
            return value;  
        }  
    }  
}
```

Listing 5: Turn class

```
public class Turn {  
    int value = 0;  
  
    void set(int val) {  
        System.out.println("Turn " +  
Neighbor.name(value));  
        value = val;  
    }  
  
    int get() {  
        return value;  
    }  
}
```