CS477:Fall 2019: Homework #2

Due Feb 29, 2019

Question 1. [5+10+10+10+15+10=60pts]

Consider the following program.

```
int pm(int x, int y)
@requires x >= 0 & y >= 0;
@ensures res = x * y
{
  a := x;
  b := y;
  res := 0;
  while (a > 0) {
     if ((a mod 2) = 1) then res := res + b;
     a := a div 2;
     @assert a >= 0;
     b := b * 2;
   }
return res;
}
```

The above code implements the algorithm for *Peasant's multiplication* (aka Egyptian multiplication (an ancient Egyptian algorithm that shows how you can do multiplication by just multiplying by 2 and doing addition). In the above, *div* denotes integer division— i.e., $a \; div \; b = \lfloor a/b \rfloor$.

- Write down an inductive loop invariant for the while loop that will prove the program correct.
- Write down the 6 basic blocks for the above program, using your loop invariant annotation. (Note that there is an assertion and a postcondition in the program above.)
- For each basic block, write down the verification conditions using weakest pre-conditions.

- For each basic block, write down the verification conditions using strongest post-conditions.
- Formulate the validity of each of the above VCs (all 8 of them) as satisfiability problems in Z3, and prove them valid.
- Finally, write the above program in Dafny with your loop invariant and prove it correct.

Question 2. [20pts]

There is a large urn with infinitely many balls, each of them that has a natural number on it (there can be many balls with the same number).

A robot first picks a ball. In every round, as long as the robot has any balls, it does the following:

- It chooses any ball it has, say with number n
- It drops the ball into the urn and picks up an arbitrary number of balls, but such that the number on each ball is less than n. (Of course, if $n = 0$, it does not pick up any balls).

It halts when it has no balls left.

Prove that the robot always terminates, no matter what the first ball is.

Prove the above formally by (a) establishing a ranking function that maps the state of the robot to a domain that has a well-founded order, and (b) showing that each move of the robot decreases the rank.