Program Verification Prof. Dr. H. Peter Gumm
Summer 2023

Exercise sheet 10

Deadline: July 01, 8:00 p.m.

Please submit only a Dafny-file **ex10_*your_name*.dfy**.

Problem 1 (3+3 points)**.** The following two programs should have the same semantics, although they differ as to where variable x is declared.
For both programs, first calculate all verification conditions, then simplify them, stating in each case, why the simplication is justified. Notice that missing *requires* or *ensures* clauses are equivalent to *requires true* and *ensures true*.

a):

```
method square0(n:nat) returns (sqn : nat)
  ensures sqn == n*n
{
  sqn := 0;
  var i := 0;
  var x ;
  while i < n
    invariant i ≤ n && sqn == i*i
  {
    x := 2*i+1 ;
    sqn,i := sqn + x, i+1;
  }
}
```

and b):

```
method square1(n:nat) returns (sqn : nat)
  ensures sqn == n*n
{
  sqn := 0;
  var i := 0;
  while i < n
    invariant i ≤ n && sqn == i*i
  {
    var x  := 2*i+1 ;
    sqn, i := sqn + x, i+1;
  }
}
```

Problem 2 (2+4 pts)**.** As you can see below, Dafny claims that after executing the following method `strange()` we will have that 1=2;

```
method q(x:nat,y:nat) returns (z:nat)
  requires y - x > 2
  ensures x < z*z < y

method strange()
  ensures 1==2
{
  var x := 4;
  var c:nat := q(x,2*x);
}
```

(a). Do you have an explanation for this behaviour?

(b). Calculate by hand, using the Hoare rules, and what you know about method calls, that indeed

$\{true\}$ `var x:nat := 4; var c := q(x,2*x);` $\{\ 1 = 2\ \}$

is correctly derived.

Problem 3 (4 pts)**.** Use what you know about the weakest preconditions/strongest postconditions to explain why the following code verifies:

```
method test0(){
    var x:int := *;
    assume  x*x < 100;
    assert x <= 9;
}
```