Tutorial 8: old 2/3rd term Nov. 24 2023

- You can solve the problems in any order. Solutions must be submitted to the automated judgement system Themis.
- The exam consist of 6 problems. You get 10 grade points for free. The problems 2, 3, and 4 are worth 10 grade point each . The remaining problems are worth 20 grade points each.
- Themis does not run tests. It only verifies correctness using the Dafny verifier. This means that a 'green' judgment means a pass, while a non-pass is actually a 'compilation error' (hence purple, not red). Clearly, this automatically implies that you cannot score partial points.
- Note that it is very easy to fake a correct submission in Themis (for example, make false a precondition or true a postcondition of a method), but of course that is not allowed. As a result, your grade in Themis is only an indication (actually, an upperbound) of your final grade. After the exam, all accepted submissions will be checked manually and points will be subtracted for 'cheating'/misleading Themis.
- The number of submissions to Themis is not limited. No points are subtracted for multiple submissions.
- Needless to say: you are not allowed to work together. If plagiarism is detected, both parties (supplier of the code and the person that sends in copied code) will be excluded from any further participation in the course.
- You are not allowed to use email, phones, tablets, calculators, etc. You are allowed to consult the Dafny lecture slides, which are available digitally (as a pdf) in Themis. You are allowed to access your own submissions previously made to Themis.

Problem 1 (20 points): MCQ (Multiple Choice Questions

For each of the following annotations determine which choice fits on the empty line (.....). The variables `x`, `y`, and `z` are of type `int`. Note that `A` and `B` (capital letters!) are specification constants (so not program variables).

```
1. // x == y + 7
   .....
   // x == 11

   (a) y := 4;
   (b) x := x - y + 4;
   (c) x := 11 - (x + y + 7);
```

2. // x == A && y == B
   .....
   // x == B - A && y == A + B

   (a) x := y - x; y := x + y;
   (b) y := x + y; x := x - 2*y;
   (c) y := x + y; x := y - 2*x;

3. // x == A && y == B
   .....
   // x - y == A - B

   (a) x := x + y; y := y + x;
   (b) x := x + 1; y := y + 1;
   (c) x := y; y := x;

4. .....
   y := x - y;
   x := x - y;
   // x == A - B && y == B

   (a) // x == A && y == A - B
   (b) // x == 2*A - 3*B && y == A - 2*B
   (c) // x == B && y == A

5. .....
   x := x + 3*y + 5;
   // 10 < x <= 16

   (a) // 10 < 2*x + 6*y < 20
   (b) // 0 < x + 3*y <= 6
   (c) // 10 <= x + 3*y + 5 < 16

6. .....
   z,x,y := x - y, x + y + z, z - y;
   // x == A + B && y == 3*B - A

   (a) // x + y + z == A + B && z - y == 3*B - A
   (b) // 3*x - 3*y == A + B && x - 2*y == 3*B - A
   (c) // x + 2*y == 2*A - 2*B && z == y - A + 3*B

Problem 2 (10 points)

From Themis you can download the file `problem2.dfy`, which contains:

```
// problem 2:
// name:      ....... (fill in your name)
// s-number: s....... (fill in your student number)
// table:     ....... (fill in your table number)

method problem2(n:int, t:int, ghost X:int, ghost Y:int)
  returns (r:int, s:int)
requires n == Y + 1 && t == X + 2*Y
ensures r == X && s == Y
{
  // X and Y are specification constants and are not allowed
  // to appear in the body of this method.
  //
  // implement yourself
  ..........
}
```

Fill in the dots such that the Dafny verifier accepts the method.

Problem 3 (10 points)

From Themis you can download the file `problem3.dfy`, which contains:

```
// problem 3:
// name:      ....... (fill in your name)
// s-number: s....... (fill in your student number)
// table:     ....... (fill in your table number)

method problem3(n:int, ghost X:int) returns (r:int)
requires X >= 0 && (n == 2*X + 5 || 2*n == 9 - X)
ensures r == X
{
  // X is a specification constant and is not allowed to appear in
  // the body of this method.
  //
  // implement yourself
  ..........
}
```

Fill in the dots such that the Dafny verifier accepts the method.

Problem 4 (10 points)

From Themis you can download the file `problem4.dfy`, which contains:

```
// problem 4:
// name:      ....... (fill in your name)
// s-number: s....... (fill in your student number)
// table:     ....... (fill in your table number)

method problem4(a: int, b:int, c:int) returns (i: int, j: int, k: int)
requires a <= b <= c
ensures i == j == k
{
  i,j,k:=a,b,c;
  while i<j || j < k
  invariant .......      // choose a suitable invariant
  decreases .......      // choose a suitable variant function
  {
    if i < j {
      i := i + 1;
    } else {
      if j < k {
        j := j + 1;
      } else {
        k := k + 1;
      }
    }
  }
}
```

Fill in the dots such that the Dafny verifier accepts the method.

Problem 5 (20 points)

From Themis you can download the file `problem5.dfy`, which contains:

```
// problem 5:
// name:      ....... (fill in your name)
// s-number: s....... (fill in your student number)
// table:     ....... (fill in your table number)

ghost function f(n: nat): int {
  // you are not allowed to remove 'ghost', so an asignment
  // like x := f(n) is not allowed.
  if n == 0 then 0 else 2*f(n/7) + n%7
}
```

```
method problem5(n:nat) returns (x: int)
ensures x == f(n)
{
  // implement yourself. You are required to give an invariant!
}
```

Fill in the dots such that the Dafny verifier accepts the method.

Problem 6 (20 points)

From Themis you can download the file `problem6.dfy`, which contains:

```
// problem 6:
// name:      ..... (fill in your name)
// s-number: s..... (fill in your student number)
// table:     ..... (fill in your table number)

ghost function f(n: nat): int {
  if n <= 1 then n else 2*f(n-1) + 3*f(n-2)
}

ghost function fSum(n: nat): int {
  // give the body of this function
  // it should return Sum(i: 0<=i<n: f(i))
  ..........
}

method problem6(n: nat) returns (a: int)
ensures a == fSum(n)
{
  var k, x, y;
  a, k,x,y := ..........;  // initialize yourself
  while k < n
    invariant 0<=k<=n && x==f(k) && y==f(k+1) && a == fSum(k)
  {
    k := k + 1;
    // complete the rest of this method
    a := ......;
    y := ......;
    x := ......;
  }
}
```

Fill in the dots such that the Dafny verifier accepts the method. The function `fSum(n)` should return $\Sigma(i:\ 0 \leq i < \mathtt{n}:\ \mathtt{f}(i))$.