CSC410

Assignment 1

Due on Friday September 25, 2020 at 11:59pm

Assignment Format and Guidelines on Submission

This assignment is worth 10% of the total course mark. Submit on Markus and follow these rules:

- This is a group assignment, designed for groups of 4 students. The volume of work is designed so that it would be reasonable for a group of 4. You may choose to work in a group of 3. Groups of fewer than 3 students will be not be accepted. You cannot submit individually. You can use Piazza or Quercus to connect with your future group members.
- Each group should submit four files named `gcd1.dfy`, `gcd2a.dfy`, `gcd2b.dfy` for their answers to questions 1, 2(a), and 2(b). Note that Markus has been setup to accept exactly those 3 files. All the required lemmas, helper functions, etc should be included in each file for each problem. That means, if you use a particular lemma to solve two different problems, you repeat it in the corresponding files. Your solution to 2(a), for example, should be repeated in 2(b) if you end up using 2(a) to solve 2(b).
- You should not give a new definition to the `gcd` function. You will add `include "gcd.dfy"` to the beginning of your Dafny files which will import the definition of `gcd` as defined in the provided file (and studied in class). Your Dafny files will only include your proofs and your helper lemmas. Note that this means that modifying the `gcd` function to get your proofs done will result in your proofs not working during the automated grading when the original `gcd` definition will be used.
- Assume statements and lemmas without bodies will result in zero marks for that question. Both provide a way of admitting facts without proofs into your reasoning and are disallowed for that reason.
- Method signatures for each method should remain exactly as specified in the handout. Changing the method signature to something incompatible with the original will result in zero marks for that question.

Note that your assignment will be automatically graded. Your function will be called from another function. If you mess with the signature, the call will fail and the autograder will give you a 0 mark.

The submission system will remain open for 12 hours after the deadline, but there is a penalty deduction formula set in Markus that deducts 4% for every hour of late submission up to 12 hours.

Word of Advice

Before you sit behind a Dafny terminal, make sure you have a detailed proof worked out on paper. If you don't have such a proof, you cannot hack your way through a Dafny proof. If you have such a proof, and are certain about its correctness, but cannot get it through to Dafny, then it most likely means that you are making a leap in reasoning somewhere that seems trivial to you, but not to the prover. We can assure you that this has nothing to do with a *feature* or a *command* that you do not know and have to dig up from a manual/tutorial. Everything you need to know to solve these has already been covered in class.

Greatest Common Divisor

All the references to `gcd` in this hand out are related to Euclid's algorithm as discussed in class with the relevant material already posted on the course webpage. In math mode, we use the standard notation

$$(a, b)$$

(from number theory) to refer to the greatest common divisor of a and b.

Problem 1 (10 points)

Prove that the Euclid's algorithm for computing the gcd satisfies the following:

$$\forall a, b, m \in \mathbb{N} : \ (a, b) = (mb - a, b) \qquad \text{if } mb > a$$
$$\forall a, b, m \in \mathbb{N} : \ (a, b) = (a - mb, b) \qquad \text{if } mb < a$$

by giving a proof for the following lemma in Dafny:

```
lemma SubCancelation(a: nat, b: nat, m: nat)
   requires a > 0 && b > 0
   ensures m * b > a ==> gcd(m * b - a, b) == gcd(a,b)
   ensures m * b < a ==> gcd(a - m * b, b) == gcd(a,b)
```

You may not change the signature of the above lemma. Only provide a body for it which proves it correct.

Problem 2 (45 points)

Consider the predicate `divides` as defined below:

```
predicate divides(a: nat, b:nat)
   requires a > 0
{
  exists k: nat :: b == k * a
}
```

It formalizes the standard mathematical concept of $a|b$.

(a) (25 points) Encode the following property of *gcd* in Dafny and prove it:

$$\forall k, a, b \in \mathbb{N} : \ k|a \wedge k|b \implies k|(a, b)$$

Note that we want to prove that Euclid's GCD algorithm satisfies the above (and not the mathematical definition of (a, b)).

(b) (30 points) Prove that *gcd* is associative, that is:

$$\forall a, b, c \in \mathbb{N} : \ (a, (b, c)) = ((a, b), c)$$

Part (a) may help with one way of arguing for this, but any other proof unrelated to part (a) will be equally fine.