# HW4 Problem Set

ECS 261

Due: Friday, May 30, 2025

1. In class, we saw that there is a difference between *truth* and *provability*. A formula is *true* over the natural numbers if for all variable assignments, the formula evaluates to true. A formula is *provable* if it can be deduced from the finitely many allowed rules of first-order logic, together with any allowed axioms.

   We don't know if this true for sure, but one statement that *may* be true but impossible to prove is the Collatz conjecture (there is a nice video by Veritasium on YouTube[1] if you would like a more detailed introduction). It says that if you take any positive integer $n$ and apply the following rules, the result should eventually end up at 1:

   - If $n$ is even, replace $n$ with $n/2$.
   - If $n$ is odd, replace $n$ with $3 * n + 1$.

   (a) Write a method in Dafny together with a pre and postcondition such that the method is correct (terminates on all inputs in a state satisfying the postcondition) if and only if the Collatz conjecture is true. Please use the following signature: `method Collatz(n: nat) returns (b: bool)`.

   (b) Does verification pass? Explain what might happen if you try to add more assertions and invariants to prove the statement.

   (c) Add the following annotation to your method: `decreases *` and to all while loops inside it. This tells Dafny that the method may not terminate. Show that you can get this modified code to pass the Dafny verifier. Explain why this does not prove the Collatz conjecture and why it does not contradict the result from class that some statements are true but not provable.

2. This problem is about strongest postconditions and weakest preconditions. For a program $P$, state which of the following is possible. If it is possible, give an example; if it is impossible, give a short proof. For all parts, preconditions and postconditions are considered the same if they are logically equivalent as formulas (i.e., each one implies the other).

   (a) Two different preconditions have the same strongest postcondition.

   (b) Two different postconditions have the same weakest precondition.

   (c) For some precondition $\varphi$ and postcondition $\psi$, $\psi$ is the strongest postcondition for $\varphi$ but $\varphi$ is not the weakest precondition for $\psi$.

[1] https://www.youtube.com/watch?v=094y1Z2wpJg

(d) For some precondition $\varphi$ and postcondition $\psi$, $\varphi$ is the weakest precondition for $\psi$ but $\psi$ is not the strongest postcondition for $\varphi$.

(e) The weakest precondition of `true` is `false`.

(f) The weakest precondition of `false` is `true`.

3.

4. According to the rules of Hoare logic,

5. Here are two different implementations of the `IsPrime` function. Use Dafny to prove that the two functions are equivalent. You should do so by filling in an appropriate spec for both functions, and then filling in the proof with associated invariants and lemma(s) as needed such that the following test passes the Dafny verifier. You should not change the implementation of either function.

```
method IsPrime1(n: nat) returns (result: bool) {
    if n <= 1 {
        return false;
    }
    for d := 2 to n {
        if (n % d == 0) {
            return false;
        }
    }
    return true;
}

method IsPrime2(n: nat) returns (result: bool) {
    if n <= 1 {
        return false;
    }
    var d := 2;
    while d * d <= n {
        if n % d == 0 {
            return false;
        }
        d := d + 1;
    }

    return true;
}

method IsPrimeEquiv(n: nat) {
    var y1 := IsPrime1(n);
    var y2 := IsPrime2(n);
    assert y1 == y2;
}
```

Attach your code with your submission. Your code should pass the verifier (`dafny verify hw4.dfy`) without any holes within a short time limit (under 10 seconds), and it should not have any unproven axioms or assume statements (`dafny audit hw4.dfy`). We recommend running these commands to check whether your

**Note:** The proof for the above is difficult. The official solution uses about 100-200 lines of lemmas to ensure the postcondition for the second function; they are various facts about integers or divisibility; for example, if $a * b = n$, then $n$ modulo $a$ and $n$ modulo $b$ return zero. These can be difficult to show, and sometimes require induction to show correctly. To use induction a lemma should itself on smaller arguments; the Dafny tutorial[2] may be helpful. Please come to office hours if you get stuck. You will receive partial credit if you can prove the code correct using some unproven lemmas (`lemma:axiom`) about integers, multiplication (`*`) or modulo (`%`) that are true statements.

**Submission instructions:**

- Upload your solutions (as a PDF) and your code (in Dafny) in Gradescope.
- If you use this LaTeX template to create your solutions, please remove the problem statements and include only your solutions.
- Your code should be a file `hw4.py`, together with any necessary helper files. It should include your solutions for parts 1, 3, and 5.
- Please include all of the function names and signatures as listed in the document above, and do not modify any function signatures. (You are welcome to add additional functions and tests.)

[2] https://dafny.org/dafny/OnlineTutorial/Lemmas