

# Secure & Dependable Systems

## Spring 2017

### Assignment 05

Humza Abid

April 6, 2017

#### **Problem 5.1** *Verification: Class Invariants*

The given class invariant holds true since in the context of the program and input variables for each method, the precondition holds true, the process terminates and then the post condition holds true (the invariant in question, in this case). This can be seen easily when the object *size*, a private member, is always constant throughout all the methods, ie. an increase or decrease of it is realized in every method of it. We can see further that the class invariant is *strong* since the invariant would hold in its own class each time a new instance of the class is called.

#### **Problem 5.2** *Verification: Class Invariants*

Choice of class: *Date*.

For the chosen class, the invariant is:

```
1 <= day && day <= 31, 1 <= month && month <= 12
```

Note, that this invariant is only a *weak invariant* since it does not hold true for all class instances (esp. in the case of February).

```
fun yesterday() {  
  if (month % 2 == 0) {  
    if (day == 1) {  
      return day = 31, month - 1;  
    } else {  
      return day - 1, month;  
    }  
  } elseif (month % 2 != 0) {  
    if (month == 3 && day == 1) {  
      return day = 28, month - 1;  
    } elseif (day == 1) {  
      return day = 30, month - 1;  
    } else {  
      return day - 1, month;  
    }  
  }  
}
```

```

fun tomorrow() {
  if (month % 2 == 0) {
    if (month == 2 && day == 28) {
      return day = 1, month + 1;
    } else if (day != 30) {
      return day + 1, month;
    } else {
      return day = 1, month + 1;
    } else if (month % != 0) {
      if (day != 31) {
        return day + 1, month;
      } else {
        return day + 1, month + 1;
      }
    }
  }
}

```

### Problem 5.3 *Verification: Pure Functions*

**Base Case:**  $zero + m == m + zero$

Applying *zero\_left* and *zero\_right*  $\implies m == m$ .

**Inductive hypothesis:**  $n + m == m + n$

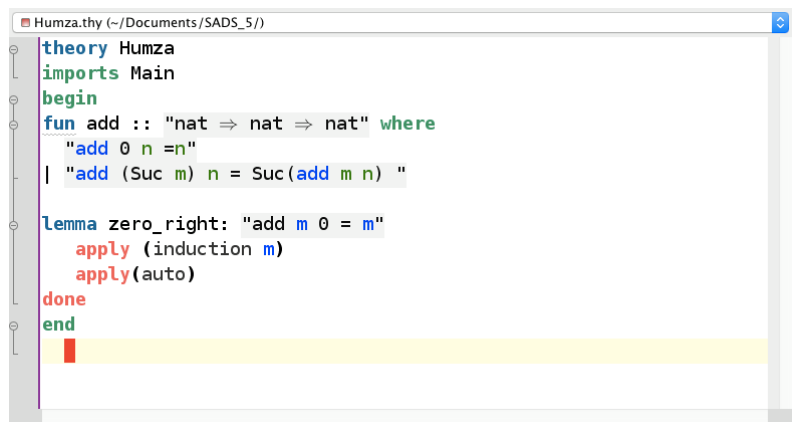
**Proving for *succ*(n):**  $succ(n) + m == m + succ(n) \implies succ(n + m) == m + succ(n)$

Using the Inductive Hypothesis:  $succ(m + n) == m + succ(n) \implies m + succ(n) == m + succ(n)$

**QED.**

### Problem 5.4 *Proof Assistants: Practice*

I installed Isabelle, and ran a simple function (addition for natural numbers) detailed in the notes. The screen shots details the function, and corresponding outputs:



```

theory Humza
  imports Main
begin
  fun add :: "nat ⇒ nat ⇒ nat" where
    "add 0 n = n"
  | "add (Suc m) n = Suc (add m n)"

  lemma zero_right: "add m 0 = m"
    apply (induction m)
    apply (auto)
  done
end

```



```

consts
  add :: "nat ⇒ nat ⇒ nat"
  Found termination order: "(λp. size (fst p)) <*> {}"

```

```
proof (prove)
goal (1 subgoal):
1. add m 0 = m
```

```
proof (prove)
goal (2 subgoals):
1. add 0 0 = 0
2.  $\wedge m. \text{add } m \ 0 = m \implies \text{add } (\text{Suc } m) \ 0 = \text{Suc } m$ 
```

```
theorem zero_right: add ?m 0 = ?m
```