Sads HW5

Atabak Hafeez

April 5, 2017

1 Verification: Class Invariants

We need to argue about the three functions.

getSize This does not change the value of any mutable variable and just returns size. Hence, the loop invariant holds here.

push If an element is pushed, the size is incremented by 1. Hence the length increases by 1 and the size increases by 1, which maintains the loop invariant.

pop If an element is popped, the size is decremented by 1. Hence, the length decreases by 1 and size decreases by 1, which maintains the loop invariant.

The loop invariant holds for a new instance as well: $length(NIL) = size = 0$

2 Verification: Class Invariants

3 Verification: Pure Functions

Using induction:

Base case: Using zero_left and zero_right rule from notes:

$$zero + n == m + zero \Rightarrow n = m$$

Step case: Inductive hypothesis: $n + m == m + n$

$$succ(n) + m == m + succ(n)$$

$$\Rightarrow succ(n + m) == m + succ(n)$$

By inductive hypothesis $\Rightarrow succ(m + n) == m + succ(n)$

$$\Rightarrow m + succ(n) == m + succ(n)$$

Hence, proven.

4 Proof Assistants

I installed coq. I used an example given with the installation files. The Factorial defined recursively under /theories/Arith was run using the ide. I have given screenshots for each step of the proof check.