Proving Properties of Functions

Exercise: Tail recursion  
Sophia Drossopoulou and Rustan Leino

# In these questions we will prove properties of functions, and will also practise the development of useful auxiliary lemmas. The answers to these exercises are shorter than in the previous exercises, but you will need to put some more thought into the proofs, or the discovery of the auxiliary lemmas.

# 1. Question – provided by Rustan Leino

Consider the definition of functions F and G:

type T;

function f(t: T): T

function F(x: T, n: nat): T

{

if n == 0 then x else F(f(x), n-1)

}

function G(x: T, n: nat): T

{

if n == 0 then x else f(G(x, n-1))

}

ensures F(x, n) == G(x, n);

Obviously, functions F and G calculate the same value. Prove that.

ghost method Lemma(x: T, n: nat)

ensures F(x, n) == G(x, n);

**Hint**: You can prove the lemma by applying strong induction and by applying the inductive hypothesis more than once, or you will need to find and prove appropriate auxiliary lemma(s).

# 2. Question – tail recursion fibonacci

Consider the definition of the function fib, and the tail recursive version with functions F:

function fib(n: nat): nat

{

if n<2 then n

else fib(n-1) + fib(n-2)

}

And now, consider the tail recursive version with functions F:

function F(n:nat, k:nat, m1:nat, m2:nat) :nat

requires n>=k;

decreases n-k;

{

if n==k then m1

else F(n, k+1, m1+m2, m1)

}

Obviously, function F was developed to calculate fib.. Prove that.

function G(n: nat): nat

{

if n<2 then n

else F(n,2,1,0)

}

ghost method Lemma(n:nat)

ensures fib(n) == G(n);

# 3. Question – more on tail recursion

Consider the definition of the function f::

function f(n: nat): nat

{

if n==0 then 9

else if n==1 then 7

else if n==2 then 5

else f(n-1) + f(n-3)

}

and its tail-recursive versionG

function G(n:nat, k:nat, m1:nat, m2:nat, m3 :nat) :nat

{

if n==0 then 9

else if n==1 then 7

else if n==2 then 5

else F(n,2,5,7,9)

}

Obviously, function F was develo0ped to calculate f.. Prove that.

**Sample answers** in Repeat\_n\_times.dfy, TailRecursionFib.dfy, and TailRecursion.dfy