Tutorial 03 — Correctness of Algorithms I

mercredi, mai 18

1

Write a reasonably short proof that the function `f(n)=n` is a bound function for this recursive algorithm.

A bound function satisfies the following properties:

- It is an integer-valued function
- Whenever the algorithm is applied recursively, the value of the function has been *decreased* by *at least one*
- If the function's value is $<= 0$ when the algorithm is applied then the algorithm does not call itself recursively during this execution.

The $fibPair()$ algorithm takes in an integer n as the input parameter. Therefore this function is a well-defined total function for this algorithm's inputs. Secondly, in the $fibPair()$ algorithm the function is only called recursively once on *line 5*, with the argument of $n-1$, with n decremented each time. This satisfies the second condition above for the bound function. Lastly, on *lines 2-4* of this algorithm, we can see that the function does not call itself recursively when the input received is $n = 0$ or $n < 0$. The algorithm will simply return a value and will terminate when this value of n is reached.

2

Write a proof that the algorithm `fibPair()` correctly solves the “Fibonacci Pair Computation” problem.

We will prove this using the standard form of *mathematical induction* on n.

BASIS STEP

Given a non-negative integer $n = 0$ as input, the algorithm returns an array F with two elements, such that $F[0] = F_n = 0$ and $F[1] = F_{n+1} = 1$, before terminating. This is due to the test on *line 2* which passes, as n is 0, which then results in *lines 3-4* being executed. The base case is established.

INDUCTIVE STEP

Let k be an integer such that $k >= 0$.

Inductive hypothesis

Given an input $n = k$, the algorithm $fibPair(k)$ will eventually terminate, and return an array F with two elements, with $F[0]$ being the k^{th} element of the Fibonacci sequence and $F[1]$ being the $(k+1)^{th}$ element of the Fibonacci sequence, like so:

$$F_k = F_k + F_{k+1}$$

Inductive claim

Given a non-negative integer input $k+1$, the algorithm $fibPair(k+1)$ will eventually terminate and return an array F with $F[0]$ being the $k+1^{th}$ term of the Fibonacci sequence, and $F[1]$ being the $k+2^{th}$ term of the Fibonacci sequence, like so:

$$F_{k+1} = F_{k+1} + F_{k+2}$$

Proof

Consider an execution of the $fibPair$ algorithm with $n = (k+1)$ given as input. Since $k >= 0$ then $n = (k+1) > 1$. Therefore the execution *lines 5-8* of the algorithm will be executed.

On *line 5* of the algorithm, the function is called recursively with the argument decrement by one, as such: $fibPair((k+1)-1)$. This can be simplified down to $fibPair(k)$ which we have defined under our *inductive hypothesis.* By assumption then, this algorithm eventually terminates, returning an array F with with $F[0]$being the k th element of the Fibonacci sequence and $F[1]$ being the $(k+1)$ th element of the Fibonacci sequence.

Then on *line 6*, the first element of the array F, $F[0]$ is set to $F[0]$ of the old F from the previous iteration, which is F_{k+1}.

Now, since $n+1 = ((k+1)+1)$, we have $k+2 >= 2$. Therefore $oldF[1]$ is set to be F_{n+1} = F_k + F_{k+1}. And by definition of the Fibonacci sequence, F_k + F_{k+1} is equal to F_{k+2}. Therefore, the second element of the current F will be set to F_{k+2}, returning an array:

$$F_{k+1} = F_{k+1} + F_{k+2}$$

As required.