Term project in Introduction to functional programming

Fie Hebsgaard-Pedersen, 20103511

1. januar 2014

1 Introduction

The goal with these projects is to show, what we have learned in the dIFP course 2013. As the name states, the course has been about functional programming, but also proving mathematical theorems in Coq. Functional programming is, as I understand it, a way of making programs that treats the computation as the evaluation of kinds of mathematical functions and doesn't have side effects or states. Where programming just is to formulate a problem to an executable program.

Coq is a proof assistant, where we get a formal language to write all kinds of mathematical definitions, theorems and proofs. Of course there are limitations, as in most(all) programs, but it is possible to extend the program, if you have the ability.

To prove a mathematical theorem and to prove the same theorem in Coq, is two quite different things. In Coq it is not allowed to say "Oh, this is obvious"or "It is intuitively right that...", Coq has to see the arguments for every step you take, when you are proving something. This means that Coq doesn't take any step itself, but see if every step you take is legal.

In the main prject I have proved d'Occagne's identity and Cassini's identity for both odd and even numbers, which is some properties of the Fibonacci numbers. d'Occagne's identity is proved with a helping lemma, while Cassini's identity is proved by induction. I've made a small addition to the fil, to use both of Cassini's identities.

The supplementary project is about 2 times 2 (2x2) matrices, where I have introduced multiplication of 2x2 matrices and the exponent of a 2x2 matrix. I show some properties about these definitons, and find a connection between the 2x2 matices and the main project. All the properties are from the dProgSprog course.

2 The main project

In this project I have proved d'Occagne's identity and Cassini's identity. The specification of the Fibonacci numbers was given, and the uniqueness of the function we have seen before, so I chose to focus on the identities instead.

To prove d'Occagne's identity, I used a theorem we had proved earlier, using induction in p. The propersition says that

$$F_{p+q+1} = F_{p+1} \cdot F_{q+1} + F_p \cdot F_q \quad \forall p, q \in \mathbb{N}$$

Where F_n is the n'th Fibonacci number

In Coq, you can destruct a specification to hypotheses, and then it gets a lot more clear, where and when you can use these. Now we can look at d'Occagne's identity as a collary of this theorem, which is why it is easier to prove, and you didn't have to use a lot of techniques to get there.

Cassini's identity for even numbers, was somewhat more difficult to get started on, because this was not near as close to intuitive, as the other proof. In Coq it can sometimes become unmanageable to see what it is, you have to rewrite to make it possible to use the induction hypothesis eg. The base case is almost never the problem.

There is added a unfold lemma for the square definition, this allow us to rewrite a square statement in a proof.

By destructing the specification as hypotheses of the Fibonacci numbers, the base case was proved using these and some rewriting of zeros and ones.

The induction case on the other hand, was not that easy. I tried to make all the rewriting on the left hand side of the equal sign, to make the proof more clear and manageable.

There had to be done a lot of rewriting of the multiplication and addition. Fortunately Coq has a lot of these techniques, as associativity and commutativity among others. Also the recursive definition af fibonacci is used, which is here: $F_{n+2} = F_{n+1} + F_n$. All this had to be done before I could use the induction hypothesis. And after the induction hypothesis used, some more rewriting was necessary, to see the two sides of the equal sign, was the same.

Now Cassini's identity for even numbers was proved, and I started to prove Cassini's identity for odd numbers, where all the rewriting is done on the right hand side of the equal sign. I wanted to make a link or association with the two identities, and I have prove a little propersition where both idetities are used. At first what I really wanted to do was to unify the two proofs, I found out that

$$F_{n-1} \cdot F_{n+1} - F_n^2 = (-1)^n \quad \forall p, q \in \mathbb{N}$$

but I couldn't find a way to express this equality, without the alternation of the (-1), which I would have to, before I could make a proof about it in Coq.

3 The supplementary project

As said before in this project I chose to prove some properties about 2x2 matrices, that we have seen in the dProgSprog course. First it is inductivly defined what a matrix is, by a create function that should decribe that a matrix is made. Then the definition of what it would say to multiply two matrices. This you do by making a definition, like you would on paper. In Coq is also used a definition and not a fixpoint, because the definition isn't recursive.

It is now stated and proved that multiplication for matrices are assosiative, not using induction, but the fact that addition and multiplication of natural numbers is assosiative.

The next definition is the identity 2x2 matrix I, and show that it is the neutral element on the left and right side, in multiplication af two matrices.

The final definition says what it means to do the exponential of a 2x2 matrix. This is done by a fixpoint in Coq, because the definiton here is recursive. With this definition I show that a small propersition about a specific matrix, where we see:

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^n = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix} \quad \forall p, q \in \mathbb{N}$$

Then I tested what it would do to take the exponent of the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^3 = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

and so on, and what I came to see was that

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n = \begin{pmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{pmatrix} \quad \forall p, q \in \mathbb{N}$$

I prefer not to write minus' in Coq and that is why I rewrote it to

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^{n+1} = \begin{pmatrix} F_n & F_{n+1} \\ F_{n+1} & F_{n+2} \end{pmatrix} \quad \forall p, q \in \mathbb{N}$$

So I added the specification of the Fibonacci numbers from the main project and proved this Theorem by induction in n. After a bit more thinking I saw a link between this project and the main project. I didn't prove this, but the connection is sweet, first we must have that

$$\det\left[\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^{n+1}\right] = \det\left[\begin{pmatrix} F_n & F_{n+1} \\ F_{n+1} & F_{n+2} \end{pmatrix}\right] \quad \forall p, q \in \mathbb{N}$$

Then after some rewriting we get that

$$F_{n-1} \cdot F_{n+1} - F_{n^2} = \det\left[\left(\begin{smallmatrix} F_n & F_{n+1} \\ F_{n+1} & F_{n+2} \end{smallmatrix}\right)^{\mathrm{T}}\right] = \det\left[\left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)^{n+1}\right] = \det\left[\left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)\right]^{n+1} = (-1)^{n+1}$$

Where T *denotes the transposed matrix.* I have not defined and proved some properties about the determinant or how the transponent of a matrix look, but the connection is there!

4 Conclusion

There has been used a lot of induction and case techniques in the projects. "The more you practice the luckier you get"you can say about using the right techniques in the right way in Coq. It is wondeful that you have to take every step, and always has to explain what you do, and explicitly tell Coq, what is has to do. The proofs are much easier to understand (when you first have made them). It is just the unmanageble side of the expressions that can be an overwhelming experince - not in a good way. But I have improved my "proving-skills"much, by not taking any step in a proof for granted.