Report about the term project, dIFP 2013

S. Hannibal Krabbe-Keblovszki 20102295

1. januar 2014

1 Background and introduction

The goal of this term project is to illustrate what we have learned in dIFP 2013-2014. The topic of this course is functional programming and proving in Coq. Coq is a proof assistant that allows us to define function and to state theorems. What we have learned about Coq in dIFP is that Coq can make constructive proofs and induction proofs. This is because Coq works with inductive constructions.

It is delightful to have the functional programming in Coq because of the pattern matching concept for definitions and fixpoints. Furthermore, functional programming in Coq gives no side effects, which means that it is easier to program mathematical things in Coq.

To prove a theorem mathematically and to prove it with Coq is two completely different things. In Coq we need to take all the steps in a proof but when it is proved mathematically it is legal to take the steps one can grasp over an equal sign.

The main project is about some properties of the Fibonacci numbers. In this file I will prove d'Occagnes identity, Cassinis identity for even numbers and Cassinis identity for odd numbers. For one of the proofs I use a helping lemma to make it all easier.

The supplementary project is about 2 times 2 matrices. I will go through matrix addition, matrix multiplication, the exponential of a matrix and how to transpose a matrix. Afterwards I will show some properties about the matrices which have to deal with the Fibonacci numbers and furthermore some properties about the transpose of a matrix.

2 Main project

In this project it has been proved that there is only one Fibonacci function. It is proved by 'induction with k base cases' where $n+1$ it the k'th base case. This is needed because I was in a situation where I should use a case for fib (S n) and this is achieved by this form for induction.

For the next proposition, d'Occagnes identity, a theorem was created to make it a lot easier to prove the identity. This theorem is:

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

and this was showed by the normal induction. The way d'Occagnes is proved is essentially as a corollary of the other theorem. This mean that we use the theorem to rewrite a statement in the proof for d'Occagnes.

The final thing to prove in the main project was the Cassini identity. Firstly Cassini for the even numbers and then for the odd numbers but overall the same tactic is used for both of these guys.

I tried just to use Coq to prove these two. This worked absolutely fine for the base case in both of the proofs, but the output from Coq was to confusing in the induction case. This forced me to do it by hand an then implement it later. What I did here was to touch the right hand side (RHS) only and do the calculation there. As an alternative I could have done this the other way around and used the left hand side (LHS) instead.

Moreover, it is natural to try to prove Cassini's identity for even and odd numbers at the same time, but when these two identities are mixed together it will become:

$$F_{n+2} \cdot F_n - F_{n+1}^2 = (-1)^{n+1}.$$

Because this is alternating it is not manageable to set this up in Coq. If it is in any way possible to write the exponent on to an even number, so you can remove the minus, then it would be possible to write this in Coq by moving the squared Fibonacci numbers to the other side of the equal sign.

3 Supplementary project

My supplementary project is about 2 times 2 matrices and some properties about these.

In Coq we allowed to use the case technique. This means that it is allowed to perform a case analysis without losing information. As an example is a lot of properties about matrices proved with this technique which means the one could say; in case the matrix is a, b, c and d then this statement is correct.

In the start of this project a definition of adding matrices is stated. Afterwards it is proved that there is only one of these functions and this proof doesn't uses induction but the case technique. This technique is used a lot in this project.

Furthermore the plus definition isn't used to anything because it was just created as a warmup for the rest of the supplementary project.

Moreover the same was created for multiplication and with the same technique but here a lemma is addede which says that matrixmultiplication is associative. In this proof it is used a lot that multiplying numbers is associative because the way you multiply matrices is very similar to the way you multiply numbers - you just add some pluses for each row and column. Actually if this was proved for adding matrices too so this would have reminded a lot about that adding numbers are associative.

Before the Fibonacci numbers is defined in this project we defined what it would say to do the exponential of a matrix. Here it is stated that the identity matrix is the result of lifting a matrix to the zeroth which is a fine definition because if we look at $x^0 = 1$ this gives the neutral element for multiplting numbers. This is the same with matrices that if one multiply a matrix with the

identity matrix you got the same matrix which means that the identity matrix is the neutral element for multiplying matrices. For instance:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 \\ 3 & 0 \end{pmatrix} = \begin{pmatrix} 2 \cdot 1 + 3 \cdot 0 & 1 \cdot 1 + 0 \cdot 0 \\ 0 \cdot 2 + 1 \cdot 3 & 0 \cdot 1 + 1 \cdot 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 3 & 0 \end{pmatrix}$$

In the section about the exponential of a matrix it is proved that there is only one of these functions and this time with induction because we are lifting to a n and it is possible to do induction in n.

The Fibonacci numbers is now stated as in the other project so no further comment to this. After this statement some test for the following matrix performed:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

which says that:

$$A^2 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, A^3 = \begin{pmatrix} 3 & 2 \\ 2 & 1 \end{pmatrix}, A^4 = \begin{pmatrix} 5 & 3 \\ 3 & 2 \end{pmatrix}$$

So in the start, to be honest, I didn't realize that the two number at the buttom of the matrix was Fibonacci numbers too. This gave me some problems but when it was realized that the whole matrix has to deal with the Fibonacci numbers we could stated:

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

but in Coq is the natural numbers like $n-1$ only defined if n is strictly positive which isn't the case if n for instance is 0. That means it is needed to rewrite the statement to:

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

and this is proved by induction in n of course. The funny part here is that this matrixequation has much in common with the main project. If one takes the determinant on both sides it look like this:

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

and if the determinant is calculated it is:

$$(1 \cdot 0 - 1 \cdot 1)^{n+1} = F_{n+2} \cdot F_n - F_{n+1}^2 \Rightarrow F_{n+2} \cdot F_n - F_{n+1}^2 = (-1)^{n+1} \quad \forall n \in \mathbb{N}$$

which is Cassini's identity for all natural numbers n. Small world.

Normally the complexity of finding the n'th Fibonacci number with the most naive algorithm is exponential. With this exponential of the matrix A it is possible to get the complexity down to $O(n)$. To multiply a matrix with another takes $O(1)$ and the multiplication is done $n-1$ times, which means the complexity is $O(n-1) = O(n)$. It is possible to improve this to $O(\log n)$, if the theorem, we used to prove d'Occagne, is used. If we rewrite the theorem by letting $q = p$ and $q = p+1$ we have way to obtain Fibonacci numbers. What is obtained is that:

$$\text{odd: } F_{2p+1} = F_{p+1}^2 + F_p^2 \quad \text{even: } F_{2p} = F_{p+2} \cdot F_{p+1} + F_{p+1} \cdot F_p$$

This means that one can find the $2p+1$'th and the $2p$'th Fibonacci numbers only with the use of the p'th and the $p+1$'th Fibonacci numbers. By using such an algorithm the time complexity of finding the n'th Fibonacci number is $O(\log n)$. This is a huge improvement when comparing this with the naive algorithm.

After this two small and cute theorems are proved by induction in n and the theorems say:

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

and

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

As a last thing in this project it is defined how to transpose a matrix. About the transpose of a matrix it is proved that the tranpose function is involutive which is pretty easy. So maybe it could be fun to prove that:

$$\left(M^\top\right)^n = \left(M^n\right)^\top$$

This is proved by induction but in the induction case some lemmas is needed to finish the proof.

The two lemmas look like this:

$$M \cdot M^n = M^n \cdot M \quad \text{and} \quad \left(M_1 \cdot M_2\right)^\top = M_1^\top \cdot M_2^\top$$

forall $n \in \mathbb{N}$. With those two lemmas it all worked just fine.

4 Summary and conclusion

In these two project a lot of induction and case technique is used. But the more you practice with these techniques the easier it is to prove something with them in Coq.

In the way Coq is used it has been amazing to see all the steps been taken in a proof which is not always the case if you make a proof by pen and paper. You simply write it all out - every little detail.

Furthermore, it is to grasp the various propositions when you've finally figured out how to make them. If one have to go back for using a lemma or proposition it is manageable to see what a proposition or a lemma say.

On the other hand, Coq can be very confusing in its output, which among other things meant that you had to write some things out by hand in order to get an overview but when this is done you just have to do in Coq what you did on paper.

It has overall been a pleasure to work with Coq.