Formalizing Fibonacci Squares

Muhammad Harun Ali Khan[1]

[1]Imperial College London

Abstract

The only squares in the Fibonacci sequence are 0, 1, and 144. We implemented a proof of this theorem in the Lean Theorem Prover. In this paper, we will discuss our methods as well as some implementation problems we faced and their solutions.

1. Introduction

Formalization is the process of implementing mathematical statements and their proofs in a theorem prover such as Coq, Lean, Isabelle, etc. These systems verify that every step of a proof is justified by checking a chain of implications down to the axioms. In this paper, we will discuss our novel formalization of the statement that 0,1, and 144 are the only Fibonacci squares in the Lean Theorem Prover.

Let $\mathbb{N}_0$ be the set of natural numbers including 0. We defined the Fibonacci and Lucas sequences as

$$\begin{aligned} &F : \mathbb{N}_0 \to \mathbb{N}_0 && L : \mathbb{N}_0 \to \mathbb{N}_0 \\ &F_0 = 0 && L_0 = 2 \\ &F_1 = 1 && L_1 = 1 \\ &F_{n+2} = F_{n+1} + F_n, && L_{n+2} = L_{n+1} + L_n. \end{aligned}$$

We formalized the following main theorem.

Theorem 3. Let $k, n \in \mathbb{N}_0$ such that $F_n = k^2$. Then $n = 0, 1, 2, 12$.

This was an old conjecture answering a natural question: the characterization of squares in one of the most extensively studied sequences. We implemented a proof due to Cohn found in [1] which uses advanced elementary number theory topics such as quadratic residues and also characterizes squares in the Lucas sequence. Our project can be accessed at https://github.com/mhk119/fibonacci_squares. The proof from [1] can be broken down into three parts: preliminaries, Lucas squares and Fibonacci squares. We have dedicated a section to each part after presenting an outline of the proof in [1].

FMM 2021 – Fifth Workshop on Formal Mathematics for Mathematicians at CICM 2021
muhammad.khan19@imperial.ac.uk (M. H. A. Khan)
0000-0003-3379-5631 (M. H. A. Khan)

2. Outline of proof in [1]

Cohn [1] considers the extension of the Fibonacci and Lucas sequences to negative numbers as well. Formally, $F_0 = 0,\ F_1 = 1, L_0 = 2,\ L_1 = 1$, and define for all integers m,

$$L_m = L_{m-1} + L_{m-2}, \qquad\qquad F_m = F_{m-1} + F_{m-2}.$$

Before proving the only Fibonacci squares are 0, 1, and 144, [1] proves two theorems.

Theorem 1. The only Lucas squares are 1 and 4.

Theorem 2. The only numbers in the Lucas sequence that are 2 times a square are 2 and 18.

To prove Theorem 1, we note the following three key lemmas that hold for all integers m.

(a) $L_{2m} = L_m^2 + (-1)^{m-1} \cdot 2$.
(b) $L_{m+2k} \equiv -L_m \pmod{L_k}$ for any even integer k not divisible by 3.
(c) $L_k \equiv 3 \pmod 4$ for any even integer k not divisible by 3.

For odd Lucas terms notice that $L_1 = 1$ is a square. Then consider $m \equiv 1 \pmod 4,\ m \neq 1$. Write $m = 1 + 2 \cdot 3^r \cdot k$ where k is an even integer not divisible by 3. By (b), $L_m \equiv -L_1 \pmod{L_k}$. And by (c), $L_k \equiv 3 \pmod 4$. So if L_m is a square, -1 is a quadratic residue modulo a number congruent to $3 \pmod 4$ which is a contradiction. Let us call this argument above the *reduction argument*. The case for when $m \equiv 3 \pmod 4$ is analogous. Lastly, by (a), even Lucas terms cannot be squares.

The proof for Theorem 2 uses essentially the same idea. We look at the the residue class modulo 8 of the index of a Lucas square. If there are no such indices in some residue class, then we can prove that half of the Lucas terms under consideration are indeed non-residues by observing patterns of Lucas numbers modulo 8 (Preliminary 13 of [1]). If there is such an index in some residue class, then we repeat the reduction argument to obtain a contradiction from the fact that -1 is a non-residue modulo a number $3 \pmod 4$.

Finally, we can prove that the only Fibonacci squares are 0, 1, and 144. Suppose $F_n = x^2$ for some integers n and x. If $n \equiv 1 \pmod 4$ notice that F_1 is a square so use the reduction argument to prove impossibility of other squares. If $n \equiv 3 \pmod 4$, we are already done since $F_n = F_{-n}$ thus reducing it to $-n \equiv 1 \pmod 4$. If n is even, one can prove $F_n = L_{n/2} F_{n/2}$ and also that $\gcd(F_{n/2}, L_{n/2}) \leq 2$. If the product of two numbers is a square then each of them is the $\gcd$ of the two numbers times a square. So we resolve two cases based on the possible values of the $\gcd$, and use Theorems 1 and 2 to obtain the result.

3. Preliminaries

These are the 13 preliminary lemmas in [1] that we first formalized. Most of our lemmas required us to prove pre-requisite results. For example, to formalize our identities (Preliminaries 1, 2, 3 (see [1])), we first formalized $\forall m, n \in \mathbb{N}_0$,

$$L_{n+1} = F_{n+2} + F_n,$$
$$L_{m+n+1} = F_{m+1}L_{n+1} + F_m L_n.$$

To prove preliminaries 4, 5, 6, and 7 (see [1]), we formalized $\forall m \in \mathbb{N}_0$,

$F_{m+1} > 0$ and $L_m > 0$,
$\gcd(F_{m+1}, F_{m+2}) = 1$,
Fibonacci residue patterns modulo 2,
Lucas residue patterns moduli 2, 3, and 4.

Additionally we proved some lemmas that shortened our proofs. For example, formalizing

$$\forall m, n \in \mathbb{N}_0,\ m \leq n \text{ or } \exists (k \in \mathbb{N}_0),\ m = k + n + 1$$

helped verify base cases for induction efficiently. Now we shall discuss two problems we faced.

3.1. Fibonacci and Lucas Closed Forms

Perhaps, we could have proved some of our identities using the closed form for the Fibonacci or Lucas sequences. However, this was problematic as performing algebraic manipulations in Lean with $\sqrt{5}$ would not have been easy or efficient. Tactics such as `simp` and `ring` are far more effective in manipulating integers.

Hence, we decided to prove all of our identities with induction. This was indeed much easier to implement. For example, proving $2L_{m+n} = 5F_mF_n + L_mL_n$, for natural numbers, took merely 7 lines with two-step induction.

3.2. Integers

Lean has a very large library of lemmas and theorems proven for natural numbers. To begin with, we defined our sequences (both Fibonacci and Lucas) from $\mathbb{N}_0 \to \mathbb{N}_0$, although [1] uses the extensions of Fibonacci and Lucas to integers. Otherwise, it would have been difficult to alternate back and forth from our integer sequences to theorems in our natural number libraries. The problem we faced in using only natural numbers is that the proof of the case $n \equiv 3 \pmod 4$ in Theorem 3 relies on the fact that $F_n = F_{-n}$ which implies $-n \equiv 1 \pmod 4$. So we easily reduced to the $1 \pmod 4$ case which implies $n = 1$ by the reduction argument presented earlier.

However, rather than defining our sequences over integers, we insisted on their natural number definitions and modified the proof of Theorem 3 slightly. Preliminaries 1, 2, 11 and 12 (from [1]) state that $\forall m, n, k \in \mathbb{Z},\ 2 \mid k,\ 3 \nmid k$,

Preliminary 1. $2F_{m+n} = F_mL_n + L_mF_n$.
Preliminary 2. $2L_{m+n} = 5F_mF_n + L_mL_n$.
Preliminary 11. $L_{m+2k} \equiv -L_m \pmod{L_k}$.

Preliminary 12. $F_{m+2k} \equiv -F_m \pmod{L_k}$.

We proved the following facts instead $\forall m, n, k \in \mathbb{N}_0,\ 2 \mid k,\ 3 \nmid k$,

(1) $2F_{m+n} = F_m L_n + L_m F_n$.
(2) $2F_m = (-1)^n F_{m+n} L_n + (-1)^{n+1} L_{m+n} F_n$.
(3) $2L_{m+n} = 5F_m F_n + L_m L_n$.
(4) $2L_m = (-1)^{n+1} \cdot 5F_{m+n} F_n + (-1)^n L_{m+n} L_n$.
(5) $L_k \mid L_{m+2k} + L_m$.
(6) $2k \geq m \implies L_k \mid L_{2k-m} + (-1)^m L_m$.
(7) $L_k \mid F_{m+2k} + F_m$.
(8) $2k \geq m \implies L_k \mid F_{2k-m} + (-1)^{m+1} F_m$.

That is, for preliminaries 1, 2, 11 and 12, we formalized an addition version and a subtraction version separately. Also, we did not need to formalize preliminaries 8 and 9 of [1] which state $F_n = (-1)^{n-1} F_{-n}$ and $L_n = (-1)^n L_{-n}$. Now we present our slightly different proof for the case when $n \equiv 3 \pmod 4$ in Theorem 3 with these modifications.

Let $n = 4 \cdot 3^r \cdot u - 1$ for $r, u \in \mathbb{N}_0$ and $\gcd(u, 3) = 1$. Then,

$$L_{2u} \mid F_{n+4u} + F_n, \tag{9}$$

$$L_{2u} \mid F_{n+4u} + F_{4u-1}, \tag{10}$$

$$L_{2u} \mid F_{4u-1} + F_1. \tag{11}$$

Fact (9) is a direct consequence of (7). Moreover, we generalized (9) by induction to obtain (10). Fact (11) is a direct consequence of (8). Perform (9) − (10) + (11) to obtain

$$L_{2u} \mid F_n + F_1. \tag{12}$$

From here, $F_1 = 1$ so $F_n \equiv -1 \pmod{L_{2u}}$. But $L_{2u} \equiv 3 \pmod 4$ so -1 is a nonresidue implying F_n is not a square.

Alternatively, we could have generalized (8) by induction to obtain (12) directly. However, we found that formalizing (12) as we did above was easier since

- fact (7) was already generalized to prove Theorem 1;
- formalizing a generalization of (8) would have been harder as it was a statement in Lean over integers. So we would have to carefully manipulate the coercion maps from natural numbers to integers.

We formalized the $n \equiv 2 \pmod 8$ case of Theorem 2 in a similar manner.

4. Lucas Squares

In this section, we will discuss our formalization of Theorems 1 and 2. Before proving Theorem 1, we proved $\forall m, n \in \mathbb{N}_0$,

(A) $m^2 + 2 \neq n^2$,
(B) $n > 0 \implies \exists r, u \in \mathbb{N}_0,\ n = 3^r \cdot u,\ 3 \nmid u$,
(C) -1 is a quadratic residue modulo $n \implies n \not\equiv 3 \pmod 4$.

To prove (A), we used quadratic residues modulo 4. Alternatively, we could have proved this using the fact that the difference between consecutive squares is greater than 2 if $m \neq 0$. However, this would yield a longer formalization due to the inequalities involved. With the latter, the lemma reduces to computing some analogous cases which Lean can easily do with the `simp` tactic.

We formalized (B) in order to decompose an index for the reduction argument. We proceeded by contradiction. If no such r exists then either, $3^r \mid n$ and $3^{r+1} \mid n$, or $3^r \nmid n$. However, the latter cannot hold, since $3^0 \mid n$ and induction yields $3^r \mid n,\ \forall r \in \mathbb{N}_0$. We showed this leads to a contradiction since $n < 3^n$ but $3^n \mid n$.

For (C), it was already proved in the `number_theory.quadratic_reciprocity` library that

$$-1 \text{ is a residue modulo } p \iff p \not\equiv 3 \pmod 4. \tag{13}$$

Our strategy was to use strong induction. We took two cases depending on whether n was prime or not. If n was not prime, then from `data.nat.prime` library, n has some prime divisor p. Then we know from (13), $p \not\equiv 3 \pmod 4$. And for n/p, we use our induction hypothesis. The problem reduces to checking that the product of two numbers not congruent to $3 \pmod 4$ is also not congruent to $3 \pmod 4$.

Perhaps, navigating between different representations of remainders was slightly inconvenient. Fact (13) uses `data.zmod.basic` for integers modulo n and field structures over modulo a prime number. On the other hand, some lemmas we used reside in `data.nat.modeq`. Finally, our lemma (C) needed to be proved as `n%4≠3`, a third representation of remainders in Lean.

Proceeding from the step in Theorem 1 after

$$L_n \equiv -4 \pmod{L_k},$$

initially seemed challenging due to division mod L_k, inverses and coercions. We decided to divide both sides by 4 and then showed that $L_k/4 \equiv (\sqrt{L_k}/2)^2$. Using lemmas in `zmod.basic` allowed us to work with inverses modulo L_k. Then Lean tactics `ring` and `nat.cast` lemmas rewrote any expressions with maps between numbers with type `zmod` L_k and natural numbers.

5. Fibonacci Squares

In this section, we will discuss our formalization of Theorem 3, the main theorem. Before we could begin formalizing Theorem 3, we needed to prove for $a, b, n \in \mathbb{N}$,

$$ab = n^2 \implies \exists r, s \in \mathbb{N},\ a = \gcd(a,b) r^2,\ b = \gcd(a,b) s^2. \tag{14}$$

We did this in two steps; by first proving

$$ab = n^2,\ \gcd(a, b) = 1 \implies \exists r \in \mathbb{N},\ a = r^2. \tag{15}$$

Once again, we proved this by strong induction. Take a prime divisor p of a. Hence $p \mid ab = n^2$. Then, by a lemma in `nat.prime` library, $p \mid n$. So $p^2 \mid n^2 = ab$. We showed by contradiction $p \nmid b$ (otherwise a, b are not coprime). Then, we have that $\gcd(p^2, b) = 1$ directly from a lemma in `nat.prime`. Lastly, from `nat.coprime` library we get that $p^2 \mid a$. So, by the induction hypothesis n/p^2 is a square so n is a square.

To prove (14), we simply considered $\dfrac{a}{\gcd(a, b)}$ and $\dfrac{b}{\gcd(a, b)}$ which are coprime from an existing lemma in the library. We showed their product is $(n/\gcd(a, b))^2$ and used (15) twice to complete the proof. Finally, Theorem 3 could be proved easily by inserting the correct lemma at every step of the proof.

6. Conclusion

Our formalisation of this theorem shows that interactive theorem provers are capable of verifying non-trivial mathematics. On one hand, powerful tactics make proving routine identities seamless. On the other hand, higher-order arguments (such as quadratic residues) are handled efficiently by simply inserting powerful lemmas into relevant parts of a proof. For humans, a complex theorem is harder to understand and hence verify. For a computer, the verification is no different to that of an easier fact. Perhaps, someday Lean can verify the proof that the only Fibonacci perfect powers are 0, 1, 8, 144 [2] - a proof that uses ingenious tools from the proof of Fermat's Last Theorem.

Acknowledgements

I thank Professor Kevin Buzzard for his guidance and supervision throughout this project.

References

[1] J. H. Cohn, Square fibonacci numbers, etc, Fibonacci Quarterly 2 (1964) 109–113. URL: https://math.la.asu.edu/~checkman/SquareFibonacci.html.

[2] Y. Bugeaud, M. Mignotte, S. Siksek, Classical and modular approaches to exponential diophantine equations i. fibonacci and lucas perfect powers, Annals of Mathematics 163 (2006) 969–1018. doi:10.4007/annals.2006.163.969.