Windmills of the Minds: An Algorithm for Fermat's Two Squares Theorem

Hing Lun Chan
Australian National University
Canberra, Australia
joseph.chan@anu.edu.au

Abstract

The two squares theorem of Fermat is a gem in number theory, with a spectacular one-sentence "proof from the Book". Here is a formalisation of this proof, with an interpretation using windmill patterns. The theory behind involves involutions on a finite set, especially the parity of the number of fixed points in the involutions. Starting as an existence proof that is non-constructive, there is an ingenious way to turn it into a constructive one. This gives an algorithm to compute the two squares by iterating the two involutions alternatively from a known fixed point.

CCS Concepts: • **Theory of computation → Automated reasoning**;

Keywords: Number Theory, Algorithm, Iteractive Theorem Proving.

ACM Reference Format:
Hing Lun Chan. 2022. Windmills of the Minds: An Algorithm for Fermat's Two Squares Theorem. In *Proceedings of the 11th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '22), January 17–18, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3497775.3503673

1 Introduction

Fermat's two squares theorem, dated back to 1640, states that a prime n that is one more than a multiple of 4 can be uniquely expressed as a sum of odd and even squares (Section 2, Theorem 2.1). Of the many proofs of this classical number theory result, this one-sentence proof by Zagier [22] caused a sensation in 1990:

> *The involution on the finite set*
>
> $$S = \{(x, y, z) \in \mathbb{N}^3 \mid n = x^2 + 4yz\}$$
>
> *defined by*
>
> $$(x,y,z) \longmapsto \begin{cases} (x + 2z, z, y - z - x) & \text{if } x < y - z \\ (2y - x, y, x + z - y) & \text{if } y - z < x < 2y \\ (x - 2y, x + z - y, y) & \text{if } x > 2y \end{cases} \quad (1)$$
>
> *has exactly one fixed point, so $|S|$ is odd, and the involution defined by $(x,y,z) \longmapsto (x,z,y)$ also has a fixed point.*

Those who are perplexed by this multi-line sentence are not alone. Even knowing involution, a self-inverse function, and fixed points, those values unchanged by a function, the proof is not obvious at a glance!

Listed as number 20 in Formalizing 100 Theorems [20], there are many formal proofs of this theorem. Some are based on textbook proofs, others follow the ideas in Zagier's proof. All show the existence of the two squares, only a few (Coq [17] and Lean [10]) include the uniqueness part. Therefore, a formalisation of this one-sentence proof, in a constructive way, is an interesting exercise in theorem-proving. As a bonus, the exercise is a path of discovery due to recent progress in understanding this proof.

As Don Zagier remarked after the one sentence, his proof was a condensed version of a 1984 proof by Roger Heath-Brown [9], who in turn acknowledged prior work in number theory taken up by Joseph Liouville [21]. This one-sentence proof invokes two involutions: the second one is obvious, but the first one in Equation (1) has been called "black magic" [18]. The algebraic formulation of this involution has been given a geometric interpretation by Alexander Spivak [16] in 2007. These are the windmills (Section 2.1). They explain why the magic works, and suggest an interplay of the involutions to identify fixed points of each other. Moreover, this provides an algorithm to find the two squares in Fermat's theorem. Thus the one-sentence proof can be made constructive, as elucidated by Zagier [11] in 2013.

CPP '22, January 17–18, 2022, Philadelphia, PA, USA

ACM ISBN 978-1-4503-9182-5/22/01…$15.00
https://doi.org/10.1145/3497775.3503673

1.1 Contribution

This paper gives the first formal proof of an algorithm to compute the two squares in Fermat's two squares theorem, by following a constructive version of Zagier's proof in HOL4.

As noted before, Zagier's proof has been formalised, in HOL Light [8], in NASA PVS [12] and in Coq [6], although not in this constructive form.

All the ideas used in this paper can be found in Shiu [15] and Zagier [11]. The novel feature of this work is an elegant and pictorial approach for our formalisation. The emphasis is in providing formal definitions and developing appropriate theories, not only for the present work, but also for supporting further work.

1.2 Overview

Major features in this formalisation are:

- the groundwork for Zagier's proof in Section 2,
- the two involutions for windmills in Section 3,
- the existence and uniqueness of two squares in Section 4,
- an algorithm to compute the two squares in Section 5,
- theories of involutions and iterations in Section 6, and
- a correctness proof of our algorithm in Section 7.

After a review of the work done, we conclude in Section 8.

1.3 Notation

Statements starting with a turnstile ($\vdash$) are HOL4 theorems, automatically pretty-printed to LaTeX from the relevant theory in the HOL4 development. Generally, our notation allows an appealing combination of quantifiers ($\forall, \exists, \exists!$), logical connectives ($\wedge$ for "and", $\vee$ for "or", $\neg$ for "not", also $\Rightarrow$ for "implies" and $\iff$ for "if and only if"), set theory ($\in$ for "element of", $\times$ for Cartesian product, and comprehensions such as $\{ x \mid x < 6 \}$), and functional programming (λ for abstraction, and juxtaposition for application). Repeated application of a function f is indicated by exponents, *e.g.*, $f\ (f\ (f\ x)) = f^3(x)$.

For a function f from set S to set T, we write $f : S \leftrightarrow T$ to mean a bijection. The empty set is denoted by $\emptyset$, and a finite set, denoted by $\mathsf{finite}\ S$, has cardinality $|S|$.

The set of natural numbers is denoted by $\mathbb{N}$, counting from 0, and $\mathsf{count}\ n \stackrel{\text{def}}{=} \{ x \mid x < n \}$, where $\stackrel{\text{def}}{=}$ means 'equality by definition'. For a natural number $n \in \mathbb{N}$, $\mathsf{square}\ n$ means it is a square: $\exists k.\ n = k^2$, $\mathsf{prime}\ n$ means it is a prime, and $\mathsf{even}\ n$ or $\mathsf{odd}\ n$ denotes its parity. The integer quotient and remainder of m divided by n are written as $m\ \mathsf{div}\ n$ and $m\ \mathsf{mod}\ n$, respectively. We write $n \mid m$ when n divides m, which is equivalent to $m \equiv 0 \pmod{n}$ when $n \neq 0$.

These are basic notations. Others will be introduced as they first appear.

HOL4 Sources. Proof scripts are located in a repository at https://bitbucket.org/jhlchan/project/src/master/fermat/twosq/. The scripts are compiled using HOL4, version `6dcb52a09341`.

In this paper, each theorem has *[script]*, which is hyperlinked to the appropriate line of the corresponding proof script in repository.

2 Sum of Two Squares

The only even prime is $2 = 1^2 + 1^2$, a sum of two squares. An odd prime, upon division by 4, leaves a remainder of either 1 or 3. Only an odd prime of the first type can be expressed as a sum of two squares, as supported by numerical evidence from Table 1.

Table 1. Examples of odd primes that can be expressed as a sum of two squares.

5	=	$4(1)+1$	=	1^2+2^2
13	=	$4(3)+1$	=	3^2+2^2
17	=	$4(4)+1$	=	1^2+4^2
29	=	$4(7)+1$	=	5^2+2^2
37	=	$4(9)+1$	=	1^2+6^2
41	=	$4(10)+1$	=	5^2+4^2
53	=	$4(13)+1$	=	7^2+2^2
61	=	$4(15)+1$	=	5^2+6^2

Pierre de Fermat, in a letter to Marin Mersenne on Christmas day 1640, claimed that he had an "irrefutable" proof of this:

Theorem 2.1 (**Two Squares Theorem**). *[script] A prime n can be expressed uniquely as a sum of odd and even squares if and only if $n = 4k + 1$ for some k.*

$$\begin{aligned}\vdash\ &\mathsf{prime}\ n \Rightarrow \\ &(n \equiv 1 \pmod 4) \iff \\ &\exists! u\ v.\ \mathsf{odd}\ u \wedge \mathsf{even}\ v \wedge n = u^2 + v^2)\end{aligned}$$

This paper concentrates on formalising an elementary proof of this result by Roger Heath-Brown, later simplified by Don Zagier. As shown in his one-sentence proof in Section 1, the idea is this: look at the representations of n not by squares, but in another form. Consider the following set S_n of triples (x, y, z):

$$S_n = \{(x, y, z) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N} \mid n = x^2 + 4yz\}. \qquad (2)$$

For a prime n of the form $4k + 1$, we have $(1, 1, k) \in S_n$. Thus the set S_n is non-empty, and there are only finitely many triples in S_n. A triple with $y = z$ will give $n = x^2 + 4y^2$, *i.e.*, a sum of two squares. If we can show that S_n has only one such triple, we have a proof of Fermat's Theorem 2.1, with both existence and uniqueness.

Meanwhile, some general theories will be developed as an exercise in formal proofs, so that they can be applied to similar problems. In addition, we extend the theories to establish not only an algorithm, but also a proof of its correctness, to compute the two unique squares for primes of the form $4k + 1$.

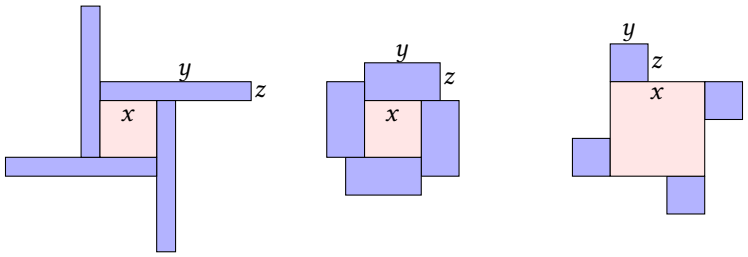


Figure 1. Typical windmills, where windmill $x\ y\ z = x^2 + 4yz$. The rightmost one has $y = z$.

2.1 Windmills

The following expression will be our main focus:

Definition 2.2. A windmill consists of a central square with four identical rectangular arms.

$$\text{windmill}\ x\ y\ z \stackrel{\text{def}}{=} x^2 + 4yz$$

Some typical windmills are shown in Figure 1. The first term x^2 is given by a central square of side x, and the second term $4yz$ is given by four arms, each a rectangle of width y and height z, arranged clockwise around the square.

Therefore each triple in the set S_n of Equation (2) can be represented by a windmill, that is, each triple (x, y, z) satisfies $n = \text{windmill}\ x\ y\ z$. Given a prime $n = 4k + 1$, we shall look for a windmill with four square arms (the one on the far right in Figure 1), *i.e.*, $y = z$, so that $n = x^2 + 4yy = x^2 + (2y)^2$. First, we collect all triples (x, y, z) which are solutions of $n = x^2 + 4yz$:

Definition 2.3. The mills of a number is its set of windmills.

$$\text{mills}\ n \stackrel{\text{def}}{=} \{\ (x,y,z)\ \mid\ n = \text{windmill}\ x\ y\ z\ \}$$

This is the formal definition of the set S_n of Equation (2). The conditions for a proper windmill, with all lengths nonzero, are:

$$\vdash \neg\text{square}\ n \wedge n \not\equiv 0\ (\text{mod}\ 4) \Rightarrow \forall x\ y\ z.\ (x,y,z) \in \text{mills}\ n \Rightarrow x \neq 0 \wedge y \neq 0 \wedge z \neq 0 \tag{3}$$

When n is a square, $n = x^2 + 4y(0)$ for any value of y. This would make mills n infinite. Otherwise:

Theorem 2.4. *[script] The number of windmills for a number n is finite if and only if n is not a square.*

$$\vdash \text{finite}\ (\text{mills}\ n) \iff \neg\text{square}\ n$$

Given an odd n that is not a square, we can determine all its windmill triples (x, y, z) by noting that, since $4yz$ is even, x must be odd, and y and z form the product $yz = (n - x^2)\ \text{div}\ 4$. In Table 2 this is worked out for $n = 29$, using successive odd x and factors for the product yz. The corresponding windmills are shown in Figure 2.

When a number n has the form $4k + 1$,

$$n = 1^2 + 4(1)k = \text{windmill}\ 1\ 1\ k,$$

showing that its mills $n \neq \emptyset$:

$$\vdash n \equiv 1\ (\text{mod}\ 4) \Rightarrow (1,1,n\ \text{div}\ 4) \in \text{mills}\ n$$

Moreover, when this form corresponds to a prime, this is the only triple (x, y, z) with $x = y$:

Theorem 2.5. *[script] For a prime of the form $4k + 1$, the only windmill with the first and second parameters equal is* windmill 1 1 k.

$$\vdash \text{prime}\ n \wedge n \equiv 1\ (\text{mod}\ 4) \Rightarrow \forall x\ z.\ n = \text{windmill}\ x\ x\ z \iff x = 1 \wedge z = n\ \text{div}\ 4$$

Proof. Note that $k = n\ \text{div}\ 4$ for prime $n = 4k + 1$. Consider $(x,y,z) \in \text{mills}\ n$ with $x = y$. This implies,

$$n = \text{windmill}\ x\ x\ z = x^2 + 4xz = x(x + 4z).$$

Therefore $x \mid n$. As prime n is not a square, $x < n$. Hence $x = 1$, so $y = 1$, and $z = k$. □

2.2 Involution

We are going to study involutions on mills n, the set of windmills for n. A function f is an involution on a set S, denoted by f involute S, when it is its own inverse:

$$f\ \text{involute}\ S \stackrel{\text{def}}{=} \forall x.\ x \in S \Rightarrow f\ x \in S \wedge f\ (f\ x) = x$$

That is, f is a bijection $f : S \leftrightarrow S$, pairing up x and $f\ x$, both in S. When $x = f\ x$, the element x is fixed by the involution f. We define the following sets:

Definition 2.6. The pairs and fixes of an involution f on a set S.

$$\text{pairs}\ f\ S \stackrel{\text{def}}{=} \{\ x\ \mid\ x \in S \wedge f\ x \neq x\ \}$$
$$\text{fixes}\ f\ S \stackrel{\text{def}}{=} \{\ x\ \mid\ x \in S \wedge f\ x = x\ \}$$

Clearly they are disjoint. The subset pairs $f\ S$ consists of distinct involute pairs, so its cardinality is even:

$$\vdash \text{finite}\ S \wedge f\ \text{involute}\ S \Rightarrow \text{even}\ |\text{pairs}\ f\ S|$$

So both $|S|$ and $|\text{fixes}\ f\ S|$ have the same parity. This leads to:

Theorem 2.7. *[script] If two involutions act on the same finite set S, their fixes have the same parity.*

$$\vdash \text{finite}\ S \wedge f\ \text{involute}\ S \wedge g\ \text{involute}\ S \Rightarrow (\text{odd}\ |\text{fixes}\ f\ S| \iff \text{odd}\ |\text{fixes}\ g\ S|)$$

Table 2. Determine all the windmill triples of $n = 29$, by odd x and factors of yz.

odd x	$n - x^2$	=	$4yz$	triple (x, y, z)	comment
1	$29 - 1^2$	=	$28 = 4(7)$	$(1, 1, 7), (1, 7, 1)$	factors of 7 are 1, 7.
3	$29 - 3^2$	=	$20 = 4(5)$	$(3, 1, 5), (3, 5, 1)$	factors of 5 are 1, 5.
5	$29 - 5^2$	=	$4 = 4(1)$	$(5, 1, 1)$	factor of 1 is 1.

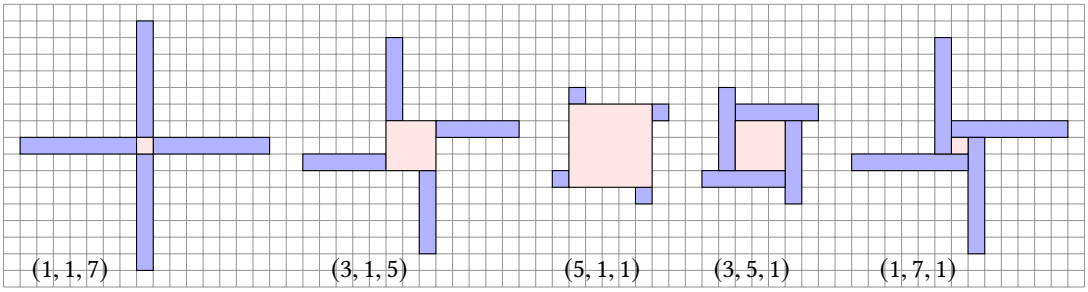


Figure 2. All the windmills of $n = 29$, determined from Table 2.

We shall meet the two involutions on mills n, a set which is finite for non-square n (by Theorem 2.4).

3 Windmill Involutions

Zagier's one-sentence proof is the interplay of two involutions on the set of windmills (mills n) for a prime $n = 4k + 1$.

3.1 Flip Map

The first involution just swaps the y and z in the triple (x, y, z):

Definition 3.1. The flip map for a triple.

$$\text{flip } (x,y,z) \stackrel{\text{def}}{=} (x,z,y)$$

The set $S = \text{mills } n$ of windmill triples of a number n can be partitioned by y, z into:

$$S_{y<z} = \{(x, y, z) \in S \mid y < z\}$$
$$S_{y=z} = \{(x, y, z) \in S \mid y = z\}$$
$$S_{y>z} = \{(x, y, z) \in S, \mid y > z\}$$

An example for $n = 29$ is shown in Figure 3. Clearly there is a bijection: flip: $S_{y<z} \leftrightarrow S_{y>z}$, and $S_{y=z}$ = fixes flip S. Thus the inverse of flip is itself:

$$\vdash \text{flip } (\text{flip } (x,y,z)) = (x,y,z)$$

showing that:

Theorem 3.2. *[script] The flip map is an involution on the set of windmills.*

$$\vdash \text{flip involute mills } n$$

3.2 Zagier Map

The other involution is the one devised by Don Zagier, as shown in Equation (1):

Definition 3.3. The Zagier map for a triple.

$$\begin{aligned}&\text{zagier } (x,y,z) \stackrel{\text{def}}{=}\\ &\quad \textbf{if } x < y - z \textbf{ then } (x + 2z,z,y - z - x)\\ &\quad \textbf{else if } x < 2y \textbf{ then } (2y - x,y,x + z - y)\\ &\quad \textbf{else } (x - 2y,x + z - y,y)\end{aligned}$$

Algebraically, this is indeed an involution, as HOL4 can verify without a blink:

$$\vdash x \neq 0 \wedge z \neq 0 \Rightarrow \text{zagier } (\text{zagier } (x,y,z)) = (x,y,z) \tag{4}$$

That HOL4 can verify this directly from definition is a showcase of its excellent algebraic simplifier, especially for natural numbers. However, we would like to see the magic behind, in terms of the geometry of windmills. Note that this definition differs slightly from Equation (1) since the else-parts include boundary cases. They actually correspond to improper windmills, and they are irrelevant for the values of n satisfying Equation (3).

3.3 Mind of a Windmill

The main purpose of introducing windmills is to read their minds.

Referring to Figure 4, a windmill has a mind (marked in dashes at middle), which is the maximum central square, with side x', that can be fitted with the four arms. When $x \leq x'$, the original square x^2 can grow to the mind x'^2, forming another windmill but keeping the overall shape (on the right). Conversely, going from right to left, we can use the mind as a reference to shrink the square term from x'^2 to x^2 by trimming four sides, thereby restoring the arms to original. Transforming a windmill's square term through the mind is the geometric interpretation of Equation (1).

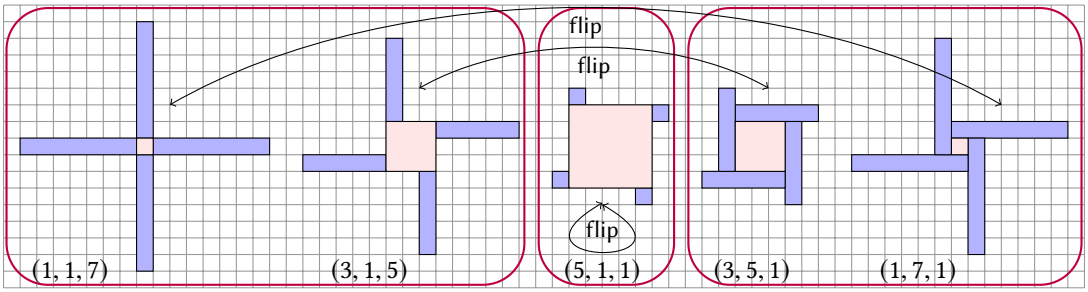


Figure 3. Partition of windmills of $n = 29$ for flip: those with $y < z$, $y = z$, and $y > z$. Note left and right pairing.

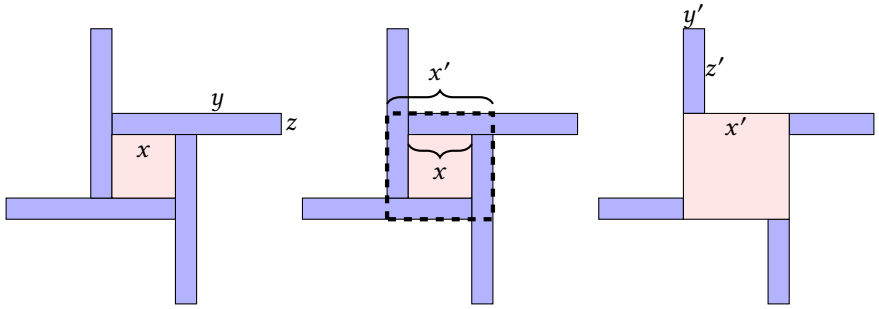


Figure 4. A typical windmill $x\ y\ z = x^2 + 4yz$, with a mind (in dashes) and transforms to another windmill.

Table 3. The five cases of Zagier map, transforming a triple (x, y, z) to (x', y', z').

Case	Type	condition	Mind	Picture	x'	y'	z'	condition	Type
1	$x < y$	$x < y - z$	$x + 2z$	Figure 5 (a)	$x + 2z$	z	$y - x - z$	$2y' < x'$	$y' < x'$
2		$y - z < x$	$2y - x$	Figure 5 (b)	$2y - x$	y	$x + z - y$	$x' < 2y'$	
3	$x = y$		x	Figure 5 (c)	x	y	z		$x' = y'$
4	$y < x$	$x < 2y$	x	Figure 5 (d)	$2y - x$	y	$x + z - y$	$y' - z' < x'$	$x' < y'$
5		$2y < x$	x	Figure 5 (e)	$x - 2y$	$x + z - y$	y	$x' < y' - z'$	

The Zagier map transforms (x, y, z) to (x', y', z') via the mind of the windmill, keeping its overall shape. There are three types, depending on whether $x < y$, $x = y$, or $y < x$. Both the first and last types are divided into two cases, as the geometry for the mind is different. Altogether there are five cases, as analysed in Table 3, and illustrated in Figure 5.[1]

Although five cases of Zagier map have been identified, note that the transformation rule:

$$(x', y', z') = (2y - x, y, x + z - y)$$

happens to be the same for case 2 and case 4. The same rule actually applies to case 3, which has $x = y$. Thus the Zagier map can be succinctly expressed as in Definition 3.3 with only three branches.

[1]Dubach and Muehlboeck [6] also identified five types for windmills.

Moreover, we can define the mind of a windmill triple as (see Table 3):

$$\begin{aligned}&\text{mind } (x,y,z) \stackrel{\text{def}}{=}\\ &\quad \textbf{if } x < y - z \textbf{ then } x + 2z\\ &\quad \textbf{else if } x < y \textbf{ then } 2y - x\\ &\quad \textbf{else } x\end{aligned}$$

and verify that the mind is an invariant under the Zagier map for any triple:

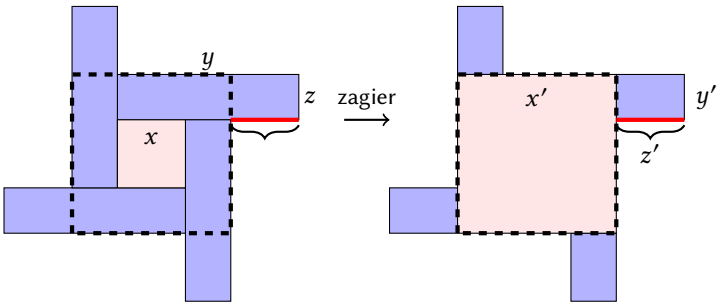
$$\vdash \text{mind } (\text{zagier } (x,y,z)) = \text{mind } (x,y,z)$$

Referring again to Table 3, the windmills in $S = \text{mills } n$ can be partitioned into three triple types:

$S_{x<y} = \{(x, y, z) \in S \mid x < y\}$ covering cases 1 and 2
$S_{x=y} = \{(x, y, z) \in S \mid x = y\}$ covering case 3
$S_{x>y} = \{(x, y, z) \in S \mid x > y\}$ covering cases 4 and 5

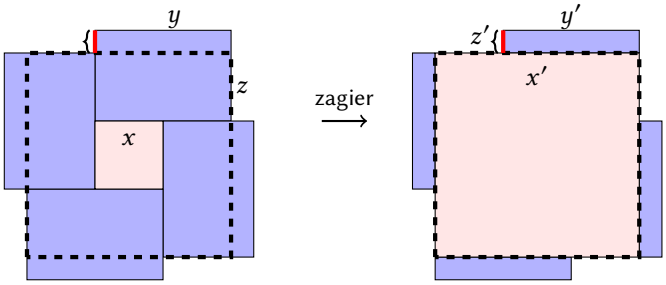
Such a partition for $n = 29$ is shown in Figure 6. Table 3 also shows that, for triples with proper windmills:

(a) Case 1: $x < y - z$.



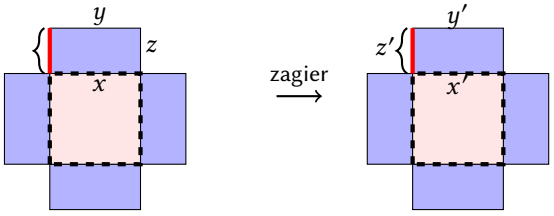
$$x' = x + 2z$$
$$y' = z$$
$$z' = y - x - z$$

(b) Case 2: $y - z < x$ and $x < y$, so $x < 2y$.



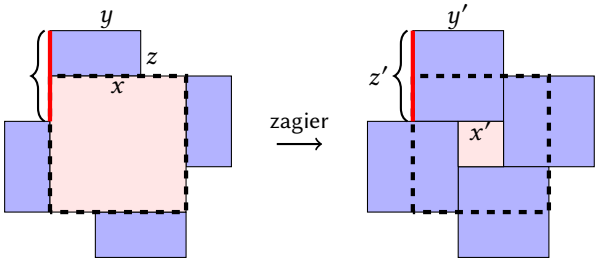
$$x' = 2y - x$$
$$y' = y$$
$$z' = x + z - y$$

(c) Case 3: $y = x$, so $y - z < x$ and $x < 2y$.



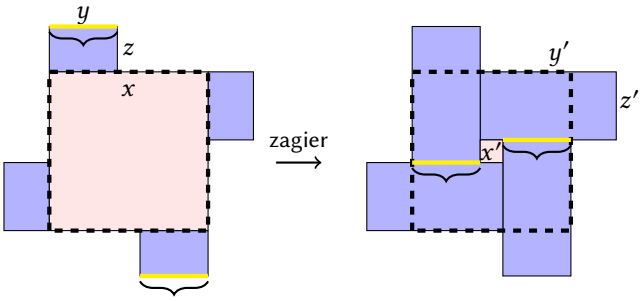
$$x' = 2y - x = x$$
$$y' = y$$
$$z' = x + z - y = z$$

(d) Case 4: $y < x$, but $x < 2y$.



$$x' = 2y - x$$
$$y' = y$$
$$z' = x + z - y$$

(e) Case 5: $2y < x$, so $y < x$.



$$x' = x - 2y$$
$$y' = x + z - y$$
$$z' = y$$

Figure 5. All five cases of the Zagier map, from (x, y, z) to (x', y', z') through the mind of a windmill.

- a triple of case 1 maps to case 5 and vice versa,
- a triple of case 2 maps to case 4 and vice versa, and
- a triple of case 3 maps to itself.

Therefore the Zagier map is its own inverse for proper triples. Combining Equation (4) and Equation (3) for the windmills of a prime, we have:

Theorem 3.4. *[script] The Zagier map is an involution on* mills *n for a prime n.*

$$\vdash \text{prime } n \Rightarrow \text{zagier involute mills } n$$

4 Two Squares Theorem

Now we have enough tools to formalise Fermat's two squares theorem.

4.1 Existence of Two Squares

For the Zagier map, it is straightforward to verify, as indicated in Table 3, that only a triple of case 3 can map to itself:

$$\vdash x \neq 0 \Rightarrow (\text{zagier } (x,y,z) = (x,y,z) \iff x = y)$$

Hence $S_{x=y} =$ fixes zagier (mills n). Applying Theorem 2.5 which characterises such triples, for certain primes $S_{x=y}$ is a singleton:

Theorem 4.1. *[script] A prime of the form $4k+1$ has only $(1,1,k)$ fixed by the Zagier map.*

$$\vdash \text{prime } n \wedge n \equiv 1 \pmod 4 \Rightarrow \text{fixes zagier } (\text{mills } n) = \{ (1,1,n \text{ div } 4) \}$$

The fixed points of two involutions play crucial roles in the existence of two squares for Theorem 2.1:

Theorem 4.2 (Two Squares Existence). *[script] A prime of the form $4k+1$ is a sum of two squares of different parity.*

$$\vdash \text{prime } n \wedge n \equiv 1 \pmod 4 \Rightarrow \exists u\ v.\ \text{odd } u \wedge \text{even } v \wedge n = u^2 + v^2$$

Proof. A prime is not a square, so mills n is finite by Theorem 2.4. , and both Zagier and flip maps are involutions on mills n, by Theorem 3.4 and Theorem 3.2. Note that Zagier map has a single fixed point by Theorem 4.1. Thus $|\text{fixes zagier } (\text{mills } n)| = 1$, so $|\text{fixes flip } (\text{mills } n)|$ is odd by Theorem 2.7. Hence fixes flip (mills n) $\neq \emptyset$, containing a triple (x,y,y). Thus $n =$ windmill $x\ y\ y = x^2 + 4y^2$. Take $u = x$, and $v = 2y$, then $n = u^2 + v^2$. Evidently v is even, and u is odd since n is odd. □

Current formalisations of Zagier's proof (HOL Light [8], NASA PVS [12] and Coq [6]), or its close relative Heath-Brown's proof (Mizar [14] and ProofPower [1]), stop at just showing the existence of two squares for the primes in Fermat's Theorem 2.1, most likely because this already meets the Formalizing 100 Theorems challenge [20]. See also related work in Section 7.5.

4.2 Uniqueness of Two Squares

The uniqueness of the two squares in Fermat's Theorem 2.1 is a consequence of the following property of a prime:

Theorem 4.3 (Two Squares Uniquenss). *[script] If a prime n can be expressed as a sum of two squares, the expression is unique up to commutativity.*

$$\vdash \text{prime } n \wedge n = a^2 + b^2 \wedge n = c^2 + d^2 \Rightarrow \{ a;\ b \} = \{ c;\ d \}$$

The proof is purely number-theoretic, which has also been formalised by Laurent Théry in Coq [17]. Moreover, we have:

Theorem 4.4. *[script] A number of the form $4k+3$ cannot be expressed as a sum of two squares.*

$$\vdash n \equiv 3 \pmod 4 \Rightarrow \forall u\ v.\ n \neq u^2 + v^2$$

This is an elementary result from possible remainders after division by 4: while a number, such as u or v, may have a remainer 0, 1, 2, or 3, a square, such as u^2 or v^2, can only have a remainder 0 or 1. Thus the sum of such remainders can never be 3.

Now we can complete the proof of Fermat's two squares Theorem 2.1:

$$\vdash \text{prime } n \Rightarrow (n \equiv 1 \pmod 4 \iff \exists! u\ v.\ \text{odd } u \wedge \text{even } v \wedge n = u^2 + v^2)$$

Proof. For the if part ($\Rightarrow$), existence is given by Theorem 4.2, and uniqueness is provided by Theorem 4.3. For the only-if part ($\Leftarrow$), an odd prime with $n \not\equiv 1 \pmod 4$ cannot be a sum of two squares by Theorem 4.4. □

5 Two Squares Algorithm

To make Zagier's proof constructive, we need to compute that single triple fixed by flip map.

Let n be a prime of the form $4k+1$. By Theorem 4.1, the only Zagier fixed point is $u = (1,1,k)$, meaning zagier $u = u$. To change the triple u, applying flip is the obvious choice. To keep changing the triple, zagier should be applied. Thus by applying the composition zagier ∘ flip repeatedly from the known Zagier fixed point, there is hope that the chain will lead to the only flip fixed point. Figure 7 shows that this is indeed the case for $n = 29$.

In terms of windmills, the flip map keeps the central square, but flips the arms of rectangles from y-by-z to z-by-y. This generally changes the mind of the windmill. The Zagier map keeps the mind, but changes the central square. Similar to the mind being an invariant of the Zagier map, the absolute difference $|y - z|$ is an invariant of the flip map. If the Zagier map can reduce this difference, the successive iterations of zagier ∘ flip will be able to locate the flip fixed point.

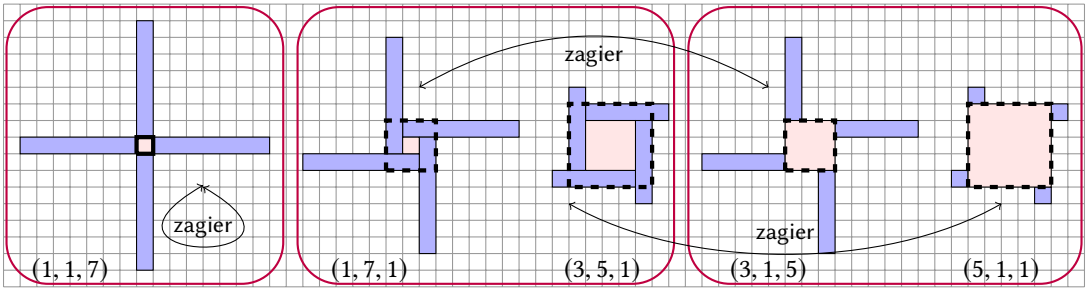


Figure 6. Partition of windmills of $n = 29$ for zagier: those with $x = y, x < y$, and $x > y$. Note pairing by minds.

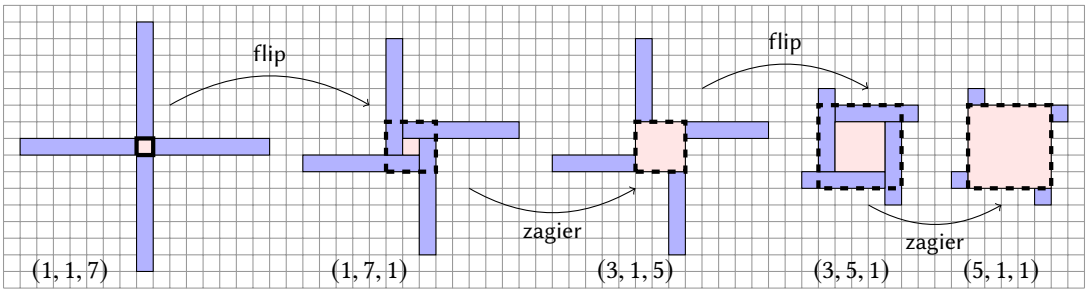


Figure 7. The iteration chain of $n = 29$ by the composition zagier ∘ flip, from Zagier fix to flip fix.

5.1 Flip Fix Search

To find the fixed point of the flip map, we can experiment with this pseudo-code:

- *Input*: a number $n = 4k + 1$.
- *Output*: a triple fixed by the flip map.
- *Method*:
- start with $u = (1,1,k)$, the Zagier fix.
- while (u is not a flip fix) :
- $\quad u \leftarrow (\text{zagier} \circ \text{flip})\ u$
- end while.

In an HOL4 interactive session, this pseudo-code can be implemented directly as:[2]

Definition 5.1. Computing the flip fixed point of $n = 4k + 1$ using a WHILE loop.

$$\begin{aligned}&\text{two\_sq } n \stackrel{\text{def}}{=}\\ &\quad \text{WHILE } ((\neg) \circ \text{found})\ (\text{zagier} \circ \text{flip})\ (1,1,n \text{ div } 4),\\ &\quad \text{where } \text{ found } (x,y,z) \stackrel{\text{def}}{=} y = z\end{aligned}$$

This simple while-loop may or may not terminate. We shall take up this issue in Section 7.3. For primes of the form $4k + 1$, it terminates and seems to work. To prove its correctness, we shall develop a theory of permutation iteration, then apply the theory to this algorithm.

[2]This pseudo-code can be implemented directly in any programming language that supports while-loops and tuples.

6 Permutation Orbits

In general, the composition of two involutions is no longer an involution, but just a permutation. Let $\varphi : S \longrightarrow S$ be a permutation, a bijection on the set S, denoted by φ permutes S. For an element $x \in S$ the iteration sequence $\varphi(x)$, $\varphi^2(x)$, $\varphi^3(x)$, *etc.*, form its *orbit*. The smallest positive index n such that $\varphi^n(x) = x$ is called the *period* of x under φ. If such a positive index does not exist, the period is defined to be 0. In HOL4, the definition makes use of OLEAST, the optional LEAST operator:

Definition 6.1. The period of function iteration of an element is the least nonzero index for the element iterate to wrap around, otherwise zero.

$$\begin{aligned}&\text{period } \varphi\ x \stackrel{\text{def}}{=}\\ &\quad \textbf{case } \text{OLEAST } k.\ 0 < k \wedge \varphi^k(x) = x \textbf{ of}\\ &\quad \mid \text{none} \triangleright 0\\ &\quad \mid \text{some } k \triangleright k\end{aligned}$$

When the set S is finite, the iterates cannot be always distinct. Thus the permutation orbit of any $x \in S$ is finite, with a nonzero period, denoted by $p = \text{period } \varphi\ x$:

$$\begin{aligned}\vdash \text{finite } S \wedge \varphi \text{ permutes } S \wedge x \in S \Rightarrow\\ \exists p.\ 0 < p \wedge p = \text{period } \varphi\ x\end{aligned}$$

and by definition the period is minimal, which means that there is no wrap around for element iterates when the index is less than the period:

$$\vdash 0 < j \wedge j < \text{period } \varphi\ x \Rightarrow \varphi^j(x) \neq x$$

This implies a criterion for an exponent index to be divisible by period:

Theorem 6.2. *[script] For a nonzero period p of x, x is fixed by the k-th iterate of φ if and only if k is a multiple of period p.*

$$\vdash 0 < p \wedge p = \text{period } \varphi\ x \Rightarrow (\varphi^k(x) = x \iff k \equiv 0 \pmod{p})$$

Moreover, the period is the same for all iterates in the same orbit:

$$\vdash \text{finite } S \wedge \varphi \text{ permutes } S \wedge x \in S \wedge y = \varphi^j(x) \Rightarrow \text{period } \varphi\ y = \text{period } \varphi\ x$$

6.1 Involution Composition

When the permutation $\varphi = f \circ g$, a composition of two involutions f and g, we shall investigate whether their fixed points are connected by a chain of composition iterations. Note the following pattern of function application:

$$f \circ (g \circ f) \circ (g \circ f) \circ (g \circ f) = (f \circ g) \circ (f \circ g) \circ (f \circ g) \circ f$$

by associativity. Also, $(f \circ g)^{-1} = g^{-1} \circ f^{-1} = g \circ f$ for involutions, so inverse is just reversal of application order in this case. Let $p = \text{period } (f \circ g)\ x$ for $x \in S$. With these notations, we can establish some basic results:

Theorem 6.3. *[script] When f fixes x, the period for x is 1 if and only if g also fixes x.*

$$\vdash f \text{ involute } S \wedge g \text{ involute } S \wedge x \in \text{fixes } f\ S \wedge p = \text{period } (f \circ g)\ x \Rightarrow (p = 1 \iff x \in \text{fixes } g\ S)$$

Pick an element x in the set S. For involutions, an iterate of $(f \circ g)$ can be equal to another iterate of $(g \circ f)$:

Theorem 6.4. *[script] The i-th iterate of $(f \circ g)$ equals the j-th iterate of $(g \circ f)$ if and only if $(i + j)$ is a multiple of period p.*

$$\vdash \text{finite } S \wedge f \text{ involute } S \wedge g \text{ involute } S \wedge x \in S \wedge p = \text{period } (f \circ g)\ x \Rightarrow ((f \circ g)^i(x) = (g \circ f)^j(x) \iff i + j \equiv 0 \pmod{p})$$

When f fixes point x, the iterates $(f \circ g)^i(x)$ and $(f \circ g)^j(x)$ are related when the sum $(i + j)$ is special:

Theorem 6.5. *[script] When f fixes x, the i-th and j-th iterate of $(f \circ g)$ differ by one f application if and only if $(i + j)$ is a multiple of period p.*

$$\vdash \text{finite } S \wedge f \text{ involute } S \wedge g \text{ involute } S \wedge x \in \text{fixes } f\ S \wedge p = \text{period } (f \circ g)\ x \Rightarrow ((f \circ g)^i(x) = f\ ((f \circ g)^j(x)) \iff i + j \equiv 0 \pmod{p})$$

There is a related result, with a similar proof:

Theorem 6.6. *[script] When f fixes x, the i-th and j-th iterate of $(f \circ g)$ differ by one g application if and only if $(i + j + 1)$ is a multiple of period p.*

$$\vdash \text{finite } S \wedge f \text{ involute } S \wedge g \text{ involute } S \wedge x \in \text{fixes } f\ S \wedge p = \text{period } (f \circ g)\ x \Rightarrow ((f \circ g)^i(x) = g\ ((f \circ g)^j(x)) \iff i + j + 1 \equiv 0 \pmod{p})$$

These theorems are useful in the study of iteration orbits starting from fixed points.

6.2 Period Parity

Given a finite set S, and an element $x \in S$, the iterates $(f \circ g)^j(x)$ form an orbit, with length equal to the period $p = \text{period } (f \circ g)\ x$. Figure 8 shows two orbits, one with an even period, the other with an odd period.

In the figure, black dots indicate iterates of $\varphi = f \circ g$, in dashes, and white dots indicate the intermediates, with g first, then f, through the arcs. Since f and g are involutions, the arcs can go both ways: forward or backward.

Let α denote a fixed point of f, and β denote a fixed point of g, *i.e.*, $f\ \alpha = \alpha$, and $g\ \beta = \beta$. We shall look at how these fixed points are related, which is crucial in the correctness proof of our algorithm (see Definition 5.1).

6.3 Fixed Point Period Even

Consider an orbit with even period starting with α, a fixed point of f. Figure 9 shows one on the left, and its real picture on the right.

This orbit is formed by taking the left diagram of Figure 8, but identifying the black dot on α (the leftmost one) with its preceding white dot from f, since $f\ \alpha = \alpha$, giving the left f-loop. This node α is now preceded by two g-arcs, one from a black dot and one from a white dot. However, g is an involution, which is injective, so the two dots are identical. The same reasoning shows that all the dots linked by double lines are identical, so that the orbit on the left can be simplified to the one on the right, taking only black dots.

Moreover, the rightmost black dot and a preceding white dot from f must be the same, due to g-arcs from identical dots. This means the half-period iterate, the rightmost black dot, is another fixed point of f, say α'. Note that $\alpha' \neq \alpha$, for otherwise the period will be affected. This example motivates the following:

Theorem 6.7. *[script] When f fixes x, and $(f \circ g)$ has an even period p for x, then f also fixes $(f \circ g)^{p \text{ div } 2}(x)$, which is not x itself.*

$$\vdash \text{finite } S \wedge f \text{ involute } S \wedge g \text{ involute } S \wedge x \in \text{fixes } f\ S \wedge p = \text{period } (f \circ g)\ x \wedge y = (f \circ g)^{p \text{ div } 2}(x) \wedge \text{even } p \Rightarrow y \in \text{fixes } f\ S \wedge y \neq x$$

Proof. First we show that f fixes y. Let $h = p \text{ div } 2$. Since period p is even, $p = 2h = h + h$. This implies that $h + h \equiv$

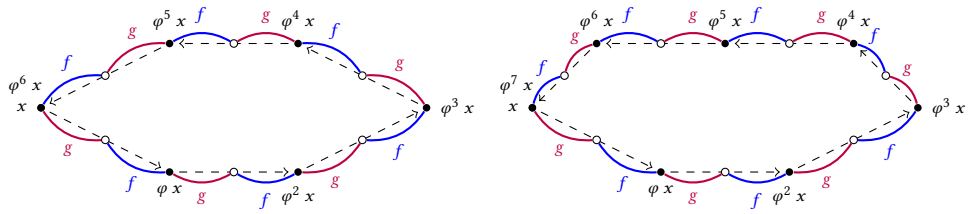


Figure 8. Orbits of $\varphi = f \circ g$ for point x. Left one has even period 6, right one has odd period 7.

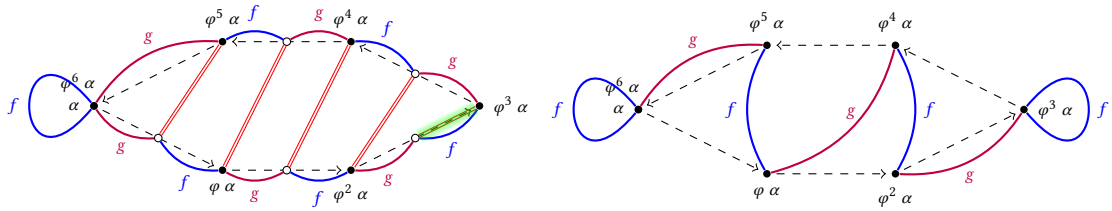


Figure 9. Orbit from an f fixed point α with even period 6. Identical points on the left (marked by two parallel lines) are merged on the right (move white dot to black dot). In particular, on the left the two vertices of the shaded line are the same, forming a fixed point of f on the right.

0 (mod p), so $y = f\ y$ by Theorem 6.5. Since $(f \circ g)$ is a permutation, $y \in S$, so $y \in \text{fixes}\ f\ S$. Next we show that $y \neq x$. Suppose $y = x$. Since for finite S the period $p \neq 0$, Theorem 6.2 shows that p divides $h = p \text{ div } 2$. Hence $p = 1$, which is not even. □

Therefore if a fixed point of f has an even period under $\varphi = f \circ g$, it is not alone. This leads directly to:

Corollary 6.8. *[script] If f fixes only a single x, then $f \circ g$ has an odd period for x.*

$$\begin{aligned}\vdash\ &\text{finite}\ S \wedge f\ \text{involute}\ S \wedge g\ \text{involute}\ S \wedge\\ &\text{fixes}\ f\ S = \{\ x\ \} \wedge p = \text{period}\ (f \circ g)\ x \Rightarrow\\ &\text{odd}\ p\end{aligned}$$

6.4 Fixed Point Period Odd

Now consider an orbit with odd period starting with α, a fixed point of f. Figure 10 shows one on the left, and its real picture on the right. This orbit is formed by taking the right diagram of Figure 8, but identifying the black dot on α (the leftmost one) with its preceding white dot from f, since $f\ \alpha = \alpha$, giving the left f-loop. The same reasoning as the even period orbit of Section 6.3 shows that all the dots linked by double lines are identical, so that the orbit on the left can be simplified to the one on the right, again taking only black dots.

Moreover, the rightmost black dot and a preceding white dot from g must be the same, due to f-arcs from identical dots. This means the half-period iterate, the rightmost black dot, must be a fixed point of g, say β. If $\beta = \alpha$, then period $p = 1$, in accordance with Theorem 6.3. This example motivates the following:

Theorem 6.9. *[script] When f fixes x, and $(f \circ g)$ has an odd period p for x, then g fixes $(f \circ g)^{p \text{ div } 2}(x)$, which is not x itself if and only if $p \neq 1$.*

$$\begin{aligned}\vdash\ &\text{finite}\ S \wedge f\ \text{involute}\ S \wedge g\ \text{involute}\ S \wedge\\ &x \in \text{fixes}\ f\ S \wedge p = \text{period}\ (f \circ g)\ x \wedge\\ &y = (f \circ g)^{p \text{ div } 2}(x) \wedge \text{odd}\ p \Rightarrow\\ &y \in \text{fixes}\ g\ S \wedge (y = x \iff p = 1)\end{aligned}$$

Proof. First we show that g fixes y. Let $h = p \text{ div } 2$. Since period p is odd, $p = 2h + 1 = h + h + 1$. Thus $h + h + 1 \equiv 0 \pmod{p}$, so $y = g\ y$ by Theorem 6.6. As $(f \circ g)$ is a permutation, $y \in S$, so $y \in \text{fixes}\ g\ S$. Theorem 6.3 ensures that: $y = x \iff p = 1$. □

6.5 Fixed Point Orbits

Let $\varphi = f \circ g$, and α, β be fixed points of f, g, respectively. Theorem 6.7 and Theorem 6.9 show that:

- if the period p of α is even, its orbit has another fixed point of f at the $h = p \text{ div } 2$ iterate: $\varphi^h(\alpha)$.
- if the period p of α is odd, its orbit has another fixed point of g at the $h = p \text{ div } 2$ iterate: $\beta = \varphi^h(\alpha)$.

Figure 9 and Figure 10 show that these orbits have no more fixed points. The only fixed point, of either f or g, occurs at halfway point of the orbit.

Thus, fixed point orbits lead directly from one fixed point to another. This is because, assuming one of the intermediate iterate is a fixed point, the iteration path will turn back, due to either f or g, both being involutions. This will produce an orbit with a shorter period, but period for an orbit is minimal.

Such considerations lead to the following stronger forms of Theorem 6.7 and Theorem 6.9:

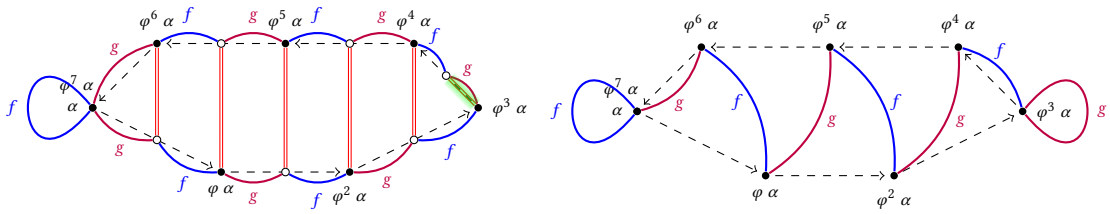


Figure 10. Orbit from an f fixed point α with odd period 7. Identical points on the left (marked by two parallel lines) are merged on the right (move white dot to black dot). In particular, on the left the two vertices of the shaded line are the same, forming a fixed point of g on the right.

Theorem 6.10. *[script] When f fixes x, the j-th iterate of $(f \circ g)$ from x is a fixed point of either f or g if and only if j is half of the period p.*

$$\begin{aligned}
&\vdash \text{finite } S \wedge f \text{ involute } S \wedge g \text{ involute } S \wedge \\
&\quad x \in \text{fixes } f\ S \wedge p = \text{period } (f \circ g)\ x \wedge \text{even } p \Rightarrow \\
&\quad\quad \forall j.\ 0 < j \wedge j < p \Rightarrow \\
&\quad\quad\quad ((f \circ g)^j(x) \in \text{fixes } f\ S \iff j = p \text{ div } 2) \\
&\vdash \text{finite } S \wedge f \text{ involute } S \wedge g \text{ involute } S \wedge \\
&\quad x \in \text{fixes } f\ S \wedge p = \text{period } (f \circ g)\ x \wedge \text{odd } p \Rightarrow \\
&\quad\quad \forall j.\ 0 < j \wedge j < p \Rightarrow \\
&\quad\quad\quad ((f \circ g)^j(x) \in \text{fixes } g\ S \iff j = p \text{ div } 2)
\end{aligned}$$

This completes our tour of the theory of permutation orbits and fixed points. The results provide the key to formally prove that our two-squares algorithm by iterations is correct.

7 Correctness of Algorithm

The algorithm to compute the flip fixed point from the known Zagier fixed point, given in Definition 5.1, makes use of a while-loop. A while-loop consists of a guard G and a body B, starting with an element x. The body is a function on x, producing iterates $B(x)$, $B^2(x)$, $B^3(a)$, *etc.*. The guard is a predicate on each iterate: the loop continues only if the test result by the guard stays true.

In HOL4, the WHILE loop with guard G and body B starting with x is defined as:

$$\text{WHILE } G\ B\ x \stackrel{\text{def}}{=} \textbf{if } G\ x \textbf{ then } \text{WHILE } G\ B\ (B\ x) \textbf{ else } x$$

from which one can easily show by induction that:

$$\begin{aligned}
&\vdash (\forall j.\ j < k \Rightarrow G\ (B^j(x))) \Rightarrow \\
&\quad \text{WHILE } G\ B\ x = \\
&\quad\quad \textbf{if } G\ (B^k(x)) \textbf{ then } \text{WHILE } G\ B\ (B^{k+1}(x)) \textbf{ else } B^k(x)
\end{aligned}$$

giving this expected result:

Theorem 7.1. *[script] The* WHILE *loop delivers the first body iterate that fails the guard test.*

$$\begin{aligned}
&\vdash (\forall j.\ j < k \Rightarrow G\ (B^j(x))) \wedge \neg G\ (B^k(x)) \Rightarrow \\
&\quad \text{WHILE } G\ B\ x = B^k(x)
\end{aligned}$$

7.1 Iterate with WHILE

From Section 6, we learn that for two involutions f and g, a fixed point α of f is paired up with a fixed point β of g whenever the period of α under the composition $\varphi = f \circ g$ is odd. In fact, β lies in the orbit of α at halfway point, the iterate at half period. Since a while-loop also gives an iterate, we have:

Theorem 7.2. *[script] For two involutions f and g, if f fixes x with an odd period, a WHILE loop with $f \circ g$ from x can reach a fixed point of g.*

$$\begin{aligned}
&\vdash \text{finite } S \wedge f \text{ involute } S \wedge g \text{ involute } S \wedge \\
&\quad x \in \text{fixes } f\ S \wedge p = \text{period } (f \circ g)\ x \wedge \text{odd } p \Rightarrow \\
&\quad\quad \text{WHILE } (\lambda t.\ g\ t \neq t)\ (f \circ g)\ x \in \text{fixes } g\ S
\end{aligned}$$

Proof. Let guard $G = (\lambda t.\ g\ t \neq t)$, and body $B = f \circ g$. If period $p = 1$, then $x \in \text{fixes } g\ S$ by Theorem 6.3. So $\neg G\ x$, and WHILE $G\ B\ x = x$ since the condition is not met at the start. Therefore WHILE $G\ B\ x \in \text{fixes } g\ s$.

If period $p \neq 1$, let $h = p \text{ div } 2$, and $z = B^h(x)$. Since $1 < p$, $0 < h < p$. Also f and g are involutions, so B permutes S. Hence $z \in \text{fixes } g\ S$ by Theorem 6.9. so $\neg G\ z$.

We claim $\forall j.\ j < h \Rightarrow G\ (B^j(x))$. To see this, let $y = B^j(x)$, which is an element of S. If $j = 0$, then $y = x$. Since period $p \neq 1$, $y \notin \text{fixes } g\ S$ by Theorem 6.3, so $G\ y$. If $j \neq 0$, then $0 < j < h < p$, and $j \neq h$. Hence $y \notin \text{fixes } g\ S$ by Theorem 6.10, so $G\ y$ again. The claim is proved.

By the claim and $\neg G\ z$, apply Theorem 7.1 to conclude WHILE $G\ B\ x = z \in \text{fixes } g\ S$. □

7.2 Two Squares by WHILE

We have developed the theory to show that the algorithm in Section 5 is correct:

Theorem 7.3. *[script] For a prime of the form $4k + 1$, the two squares algorithm of Definiton 5.1 gives a flip fixed point.*

$$\begin{aligned}
&\vdash \text{prime } n \wedge n \equiv 1\ (\text{mod } 4) \Rightarrow \\
&\quad \text{two_sq } n \in \text{fixes flip } (\text{mills } n)
\end{aligned}$$

Proof. Let $S = \text{mills } n$, $\varphi = \text{zagier} \circ \text{flip}$, $u = (1,1,n \text{ div } 4)$, and period $p = \text{period } \varphi\ u$. By Definition 5.1, and noting that $(\neg) \circ \text{found} = (\lambda t.\ \text{flip } t \neq t)$, this is to show: WHILE $(\lambda t.\ \text{flip } t \neq t)\ \varphi\ u \in \text{fixes flip } S$.

Since a prime is not a square, we have finite S. Now φ permutes S as Zagier map and flip map are both involutions, by Theorem 3.4 and Theorem 3.2, and fixes zagier $S = \{ u \}$ by Theorem 4.1. Thus $u \in$ fixes zagier S, and period p is odd by Corollary 6.8. So WHILE $(\lambda t.\ \text{flip}\ t \neq t)\ \varphi\ u \in$ fixes flip S by Theorem 7.2. □

It is almost trivial to convert two_sq n to following algorithm:

Definition 7.4. Compute the two squares for Fermat's two squares theorem.

$$\text{two\_squares}\ n \stackrel{\text{def}}{=} (\textbf{let}\ (x,y,z)\ =\ \text{two\_sq}\ n\ \textbf{in}\ (x,y+z))$$

giving the two squares in a pair, and its correctness is readily demonstrated:

Theorem 7.5. *[script] The algorithm by Definition 7.4 gives indeed Fermat's two squares.*

$$\vdash \text{prime}\ n \wedge n\ \equiv\ 1\ (\text{mod}\ 4) \Rightarrow (\textbf{let}\ (u,v)\ =\ \text{two\_squares}\ n\ \textbf{in}\ n = u^2 + v^2)$$

Table 4 shows a sample run in HOL4 session on a typical laptop, using EVAL for evaluation and prefix time to obtain timing statistics. Note that these EVAL executions are based on optimised symbolic rewriting in HOL4, thus orders of magnitude slower than running native code.

Other algorithms. A prime has a finite set of windmill triples, by Theorem 2.4. Fermat's two squares for a prime n with $n\ \equiv\ 1\ (\text{mod}\ 4)$, which must exist by Theorem 4.2, can be found by a brute-force search: subtract n by successive odd squares, and check whether the difference is a square. Although there are better ways to test a square than the square-root test, they are not simple to implement.

Don Zagier, after his one-sentence proof, referred to an effective algorithm by Wagon [19] to compute the two squares. The algorithm requires finding a quadratic non-residue of the given prime n.

The advantage of our algorithm in Definition 7.4 over such alternative methods is that only addition and subtraction are performed. The implementation is rather straightforward. The issue of termination is discussed next.

7.3 Terminating Condition

As mentioned in Section 5.1, for our algorithm the WHILE loop may or may not terminate. To gaurantee termination, convert the WHILE loop to a countdown loop, as follows. First, ensure that the input number n is not a square, so that mills n is finite (Theorem 2.4), and check $n\ \equiv\ 1\ (\text{mod}\ 4)$, so that $(1,1,n\ \text{div}\ 4) \in \text{mills}\ n$, *i.e.*, mills $n \neq \emptyset$.

Obviously for any triple $(x,y,z) \in \text{mills}\ n$, each x, y or y is less than n, hence $|\text{mills}\ n| < n^3$. Now, use a countdown loop from n^3 to 0, start with the triple $(1,1,n\ \text{div}\ 4)$ for the zagier ∘ flip iteration.

The iterations trace an orbit. At half-way point, the orbit hits either a flip fixed point, detected by $y = z$, when the period is odd (Theorem 6.9), or another Zagier fixed point, detected by $x = y$, when the period is even (Theorem 6.7). They provide actual exits from the countdown loop, much earlier than the count drops to zero.

7.4 Lessons Learnt

This formalisation work can be a self-contained project in a theorem-proving workshop. The ideas are simple, but formulating the theorems properly is not simple. For example, at first the author would like to prove:

$$\forall\ x\ y\ z.\ \text{zagier}\ (\text{zagier}\ (x,y,z)) = (x,y,z).$$

The interactive session produces several subgoals which he cannot resolve immediately. A comparison of Definition 3.3 with Equation (1) shows differences in boundary cases. Finally, some insight from windmills resolves why the boundaries are ignored, and provides the pre-condition $x \neq 0 \wedge z \neq 0$, see Equation (4). The result is Theorem 3.4.

Explaining the Zagier map is an involution through the mind of a windmill poses some challenges. Definition 3.3 of the Zagier map has 3 branches, so the initial effort is to treat just 3 cases. The first case is immediate, but the second case runs into a mess. It is only after drawing a lot of windmills that the author realises these finer points:

- there are 3 types: $x < y, x = y$, and $x > y$ for the windmill triple (x, y, z),
- the 3 types are further subdivided due to geometry of the mind, giving 5 cases in total,
- the 5 cases can be condensed into 3 branches, as the definition shows.

The result is Table 3 in Section 3.3.

For the permutation orbits in Section 6, the proofs about relations between iterates start as long-winded arguments treating if-part and only-if part separately. Putting them in this paper prompts the author to rethink the logic. The polished proofs simply employ a chain of logical equivalences.

Fixed point orbits have either even or odd period, as treated in Section 6.3 and Section 6.4. The drawing of the diagrams helps to refine the proofs to be short and sweet, making good use of theorems already proved.

About the correctness proof of the algorithm using a while-loop in Section 7, the author initially applied Hoare logic assertions to derive the desired iterate upon loop exit. This is awkward, as pointed out by Michael Norrish who knows the HOL4 theorem-prover inside out. The reason is that WHILE is *defined* as iteration of the body in HOL4. The section had since been rewritten.

Development Effort. The proofs have been streamlined after several revisions. Such refinements result in the script line counts for various theories developed, shown in Table 5.

Table 4. Running Fermat's two squares algorithm in a HOL4 session, with timing.

```
> time EVAL ``two_squares 97``;
runtime: 0.00770s, gctime: 0.00086s, systime: 0.00077s.
val it = |- two_squares 97 = (9,4): thm
> time EVAL ``two_squares 1999999913``;
runtime: 2m23s, gctime: 14.7s, systime: 11.3s.
val it = |- two_squares 1999999913 = (1093,44708): thm
> time EVAL ``two_squares 12345678949``;
runtime: 6m02s,    gctime: 37.5s,     systime: 26.0s.
val it = |- two_squares 12345678949 = (110415,12418): thm
> EVAL ``9 * 9 + 4 * 4``;
val it = |- 9 * 9 + 4 * 4 = 97: thm
> EVAL ``1093 * 1093 + 44708 * 44708``;
val it = |- 1093 * 1093 + 44708 * 44708 = 1999999913: thm
> EVAL ``110415 * 110415 + 12418 * 12418``;
val it = |- 110415 * 110415 + 12418 * 12418 = 12345678949: thm
```

Table 5. Statistics of various theories in this work.

HOL4 Theory	Description	#Lines
involute	basic involution	231
iteration	function iteration and period	917
iterateCompose	iteration of involute composition	1648
iterateCompute	iteration period computation	939
windmill	windmills and their involutions	1844
twoSquares	two-squares by windmills	1304

The scripts are fully documented, including the traditional proofs as comment before each theorem. Although comments almost double the script size, the line counts are still indicative of the effort to convert ideas into formal proofs.

7.5 Related Work

As noted in Section 1, Fermat's two squares theorem has been formalised. However, none of these formal proofs is constructive, in the sense that there is no formal proof of an algorithm to compute the two squares for a prime satisfying the theorem.

Fermat's two squares theorem has two parts: existence and uniqueness. All formal proofs include the existence part (see Theorem 4.2) , using classic and modern existence proofs: the method of infinite descent is used in one system, Gaussian integers are employed in three systems, both Heath-Brown's proof and Zagier's proof are treated in two systems.

Only two formal proofs include the uniqueness part (see Theorem 4.3): Théry [17] proved by algebraic identities and divisibility, and Hughes [10] proved by unique factorisation of Gaussian integers.

Recently, Dubach and Muehlboeck [6] formalised Zagier's proof using involutions in Coq's Mathematical Components Library. They illustrated their proof using the windmills as per this paper, and extended the use of involutions on the same set to formalise also an integer-partition proof of Fermat's two squares theorem by Christopher [5].

A summary of these formal proofs, in chronological order, is given in Table 6.

8 Conclusion

About Fermat's two squares theorem, G. H. Hardy wrote in his 1940 essay *A Mathematician's Apology* [7, Section 13]:

> *This is Fermat's theorem, which is ranked, very justly, as one of the finest of arithmetic. Unfortunately, there is no proof within the comprehension of anybody but a fairly expert mathematician.*

This work has been a rewarding exercise in formalisation, delivering a proof of Fermat's Theorem 2.1 using only natural numbers, involutions, and counting. There is a certain sense of mathematical beauty when a non-trivial result can be shown by elementary means, borrowing elegant ideas by Zagier and Spivak. Moreover, by developing a theory of involution iteration, an algorithm to compute the two squares of the theorem can be formally shown to be correct.

Future Work. The theory in Section 6, about orbits and fixed points, can be developed using group actions, since the iteration indices form an addition cyclic group under mod p, where p is the orbit period. One can exploit the symmetry in permutation orbits, especially for permutations arising from two involutions, to improve the algorithm, as shown in the analysis by Shiu [15]. In HOL4, this direction can start

Table 6. Chronology of formalisation of Fermat's two squares theorem.

Year	Author(s)[reference]	Theorem Prover	Comment
2004	Laurent Théry [17]	Coq	Gaussian integers, with uniqueness
2007	Roelof Oosterhuis [13]	Isabelle	Euler's proof with infinite descent
2009	Marco Riccardi [14]	Mizar	Heath-Brown's proof with involutions
2010	John Harrison [8]	HOL Light	Zagier's proof with involutions
2012	Anthony Narkawicz [12]	NASA PVS	Zagier's proof with involutions
2015	Mario Carneiro [2]	MetaMath	Gaussian integers
2016	Rob Arthan [1]	ProofPower	Heath-Brown's proof with involutions
2019	Chris Hughes [10]	Lean	Principal Ideal Ring of Gaussian integers, with uniqueness
2021	Dubach and Muehlboeck [6]	Coq	Zagier's and Christopher's proofs with involutions

from the algebra of group theory in Chan and Norrish [4]. A formal analysis of the performance of the algorithm for two squares described in Definition 5.1 can be modelled using an approach in Chan [3].

Acknowledgements

Many thanks to Michael Norrish for his careful review of the draft, providing useful advice and helpful recommendations to improve this paper. The author is also grateful to the anonymous reviewers who pointed out typographical errors and suggested clarifications. This paper has been revised to incorporate their comments.

References

[1] Rob Arthan. 2016. Mathematical Case Studies: Some Number Theory. Supplementary ProofPower Examples. Available from http://www.lemma-one.com/ProofPower/examples/wrk074.pdf, accessed 22 August, 2021.

[2] Mario Carneiro. 2015. Theorem 2sq. Metamath 100 proof. Available from http://us.metamath.org/mpeuni/2sq.html, accessed 22 August, 2021.

[3] Hing Lun Chan. 2019. *Primality Testing is Polynomial-time: a Mechanised Verification of the AKS Algorithm.* PhD. Australian National University, Canberra, Australia. https://doi.org/10.25911/5F58B06CA124E

[4] Hing Lun Chan and Michael Norrish. 2012. A String of Pearls: Proofs of Fermat's Little Theorem. In *Proceedings of Certified Programs and Proofs (LNCS, 7679)*, Chris Hawblitzel and Dale Miller (Eds.). Springer, 188–207. https://doi.org/10.1007/978-3-642-35308-6_16

[5] A. David Christopher. 2016. A partition-theoretic proof of Fermat's Two Squares Theorem. *Discrete Mathematics* 339, 4 (06 April 2016), 1410–1411. https://doi.org/10.1016/j.disc.2015.12.002

[6] Guillaume Dubach and Fabian Muehlboeck. 2021. Formal verification of Zagier's one-sentence proof. (24 April 2021). In arXiv: https://arxiv.org/abs/2103.11389, accessed 22 August, 2021.

[7] Godfrey H. Hardy. 1940. *A Mathematician's Apology.* Cambridge University Press. https://doi.org/10.1017/CBO9781139644112

[8] John Harrison. 2010. Representation of primes ≡ 1 (mod 4) as sum of 2 squares. Formalizing 100 Theorems in HOL Light. Available from https://github.com/jrh13/hol-light/blob/master/100/two_squares.ml, accessed 22 August, 2021.

[9] Roger Heath-Brown. 1984. Fermat's two squares theorem. *Invariant* 11 (1984), 3–5. Available from https://core.ac.uk/reader/97080, accessed 22 August, 2021.

[10] Chris Hughes. 2019. Sums of two squares. Lean Community, 100 theorems. Available from https://leanprover-community.github.io/100.html, accessed 22 August, 2021.

[11] Aimeric Malter, Dierk Schleicher, and Don Zagier. 2013. New Looks at Old Number Theory. *The American Mathematical Monthly* 120, 3 (2013), 243–264. https://doi.org/10.4169/amer.math.monthly.120.03.243

[12] Anthony Narkawicz. 2012. primes_sum_squares: THEORY. NASA PVS Library 6.0.9. In https://github.com/nasa/pvslib/blob/master/numbers/primes_sum_squares.pvs, accessed 22 August, 2021.

[13] Roelof Oosterhuis. 2007. Theory TwoSquares. Isabelle, Archive of Formal Proofs. Available from https://www.isa-afp.org/entries/SumSquares.html, accessed 22 August, 2021.

[14] Marco Riccardi. 2009. theorem :: NAT_5:23. Formalizing 100 Theorems in Mizar. Available from http://www.mizar.org/100/index.html, accessed 22 August, 2021.

[15] Peter Shiu. 1996. Involutions associated with Sums of Two Squares. *Publications de l'Institut Mathématique* 59 (1996), 18–39.

[16] Alexander Spivak. 2007. Winged Squares [in Russian]. *Popular lectures in mathematics, 2006-2007 academic year, Lecture 15 (165)* (10 March 2007). Available from http://mmmf.msu.ru/lect/spivak/zagir_!.pdf, accessed 22 August, 2021.

[17] Laurent Théry. 2004. Numbers Equal to the Sum of Two Square Numbers. Formalizing 100 theorems in Coq. Available from https://github.com/coq-contribs/sum-of-two-square, accessed 22 August, 2021.

[18] Todd Trimble and Vishal Lama. 2008. A proof from The Book? Todd and Vishal's blog. Available from http://topologicalmusings.wordpress.com/2008/05/04/a-proof-from-the-book/, accessed 22 August, 2021.

[19] Stan Wagon. 1990. Editor's Corner: The Euclidean Algorithm Strikes Again. *The American Mathematical Monthly* 97, 2 (1990), 125–129. https://doi.org/10.2307/2323912

[20] Freek Wiedijk. last update: 2020. Formalizing 100 Theorems. Available from https://www.cs.ru.nl/~freek/100/, accessed 22 August, 2021.

[21] Kenneth S. Williams. 2010. *Number Theory in the Spirit of Liouville.* Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9780511751684 ISBN: 9780511751684.

[22] Don Zagier. 1990. A One-Sentence Proof That Every Prime $p \equiv 1 \pmod 4$ Is a Sum of Two Squares. *The American Mathematical Monthly* 97, 2 (1990), 144. https://doi.org/10.2307/2323918