Sums of two and four squares

Roelof Oosterhuis
University of Groningen

June 11, 2019

Abstract

This document gives the formal proofs of the following results about the sums of two and four squares:

1. Any prime number \( p \equiv 1 \mod 4 \) can be written as the sum of two squares.
2. (Lagrange) Any natural number can be written as the sum of four squares.

The proofs are largely based on chapters II and III of the book by Weil [Wei83].

The results have been formalised before in the proof assistant HOL Light [Har]. A more complete study of the sum of two squares, including the first result, has been formalised in Coq [The04]. The results can also be found as numbers 20 and 19 on the list of ‘top 100 mathematical theorems’ [Wie].

This research is part of an M.Sc. thesis under supervision of Jaap Top and Wim H. Hesselink (RU Groningen). For more information see [Oos07].
Theory TwoSquares

imports
  HOL-Number-Theory.Number-Theory

begin

context

  fixes sum2sq-nat :: nat ⇒ nat ⇒ nat
  defines sum2sq-nat a b ≡ a^2 + b^2

  fixes is-sum2sq-nat :: nat ⇒ bool
  defines is-sum2sq-nat n ≡ (∃ a b. n = sum2sq-nat a b)

begin

private lemma best-division-abs: (n::int) > 0 ⇒ ∃ k. 2 * |a - k*n| ≤ n
⟨proof⟩

definition sum2sq-int :: int × int ⇒ int where
sum2sq-int = (λ(a,b). a^2 + b^2)

private definition is-sum2sq-int :: int ⇒ bool where
is-sum2sq-int n ←→ (∃ a b. n = sum2sq-int(a,b))

private lemma sum2sq-int-nat-eq: sum2sq-nat a b = sum2sq-int(a,b)
⟨proof⟩

lemma is-sum2sq-int-nat-eq: is-sum2sq-nat n =⇒ is-sum2sq-int(int n)
⟨proof⟩

lemma product-two-squares-aux: sum2sq-int(a, b) * sum2sq-int(c, d) = sum2sq-int(a*c - b*d, a*d + b*c)
⟨proof⟩

lemma product-two-squares-int: is-sum2sq-int m =⇒ is-sum2sq-int(n) =⇒ is-sum2sq-int(m*n)
⟨proof⟩

lemma product-two-squares-nat: is-sum2sq-nat m =⇒ is-sum2sq-nat n =⇒ is-sum2sq-nat(m*n)
⟨proof⟩

lemma sots1-aux: assumes prime (4*k+3) assumes odd (multiplicity (4*k+3) n) shows ¬ is-sum2sq-nat n
⟨proof⟩

lemma sots1: assumes is-sum2sq-nat n shows ∃ k. prime (4*k+3) =⇒ even (multiplicity (4*k+3) n)
⟨proof⟩

lemma aux-lemma: assumes [(a::nat) = b] (mod c) b < c shows ∃ k. a = c*k + b
⟨proof⟩

lemma Legendre-1mod4: prime (4*k+1::nat) =⇒ (Legendre (-1) (4*k+1)) = 1
⟨proof⟩

lemma qf1-prime-exists: prime (4*k+1) =⇒ is-sum2sq-nat (4*k+1)
⟨proof⟩

lemma fermat-two-squares: assumes prime p (¬ [p = 3] (mod 4)) shows is-sum2sq-nat p
⟨proof⟩

lemma sots2: assumes k. prime (4*k+3) =⇒ even (multiplicity (4*k+3)

Contents

1 Lagrange's four-square theorem 3
n) shows is-sum2sq-nat n {proof}

theorem sum-of-two-squares:
    is-sum2sq-nat n ℎ→ (∀ k. prime (4∗k+3) ℎ→ even (multiplicity (4∗k+3) n))
⟨proof⟩ lemma k-mod-eq: (∀ p::nat. prime p ∧ [p = 3] (mod 4) ℎ→ P p) = (∀ k. prime (4∗k+3) ℎ→ P (4∗k+3))
⟨proof⟩

theorem sum-of-two-squares':
    is-sum2sq-nat n ℎ→ (∀ p. prime p ∧ [p = 3] (mod 4) ℎ→ even (multiplicity p n))
⟨proof⟩

theorem sum-of-two-squares-prime: assumes prime p
    shows is-sum2sq-nat p = [p≠3] (mod 4)
⟨proof⟩

end

1 Lagrange’s four-square theorem

theory FourSquares
    imports HOL-Number-Theory.Number-Theory
begin

context
    fixes sum4sq-nat :: nat ⇒ nat ⇒ nat ⇒ nat ⇒ nat
    defines sum4sq-nat a b c d ≡ a^2+b^2+c^2+d^2

    fixes is-sum4sq-nat :: nat ⇒ bool
    defines is-sum4sq-nat n ≡ (∃ a b c d. n = sum4sq-nat a b c d)

begin

private lemma best-division-abs: (n::int) > 0 ⇒ ∃ k. 2 * |a - k*n| ≤ n
⟨proof⟩

Shows that all nonnegative integers can be written as the sum of four squares.

The proof consists of the following steps:

• For every prime p = 2n + 1 the two sets of residue classes
  \{x^2 \mod p \mid 0 \leq x \leq n\} and \{-1 - y^2 \mod p \mid 0 \leq y \leq n\}
  both contain n + 1 different elements and therefore they must have at least
  one element in common.

• Hence there exist x, y such that x^2 + y^2 + 1^2 + 0^2 is a multiple of p.
The next step is to show, by an infinite descent, that \( p \) itself can be written as the sum of four squares.

Finally, using the multiplicity of this form, the same holds for all positive numbers.

**private definition**

\[ \text{sum4sq-int :: } \text{int} \times \text{int} \times \text{int} \times \text{int} \Rightarrow \text{int} \]

\[ \text{sum4sq-int} = (\lambda (a,b,c,d). \ a^2+b^2+c^2+d^2) \]

**private definition**

\[ \text{is-sum4sq-int :: } \Rightarrow \text{bool} \]

\[ \text{is-sum4sq-int} n \leftrightarrow (\exists a b c d. n = \text{sum4sq-int}(a,b,c,d)) \]

**private lemma** mult-sum4sq-int: \( \text{sum4sq-int}(a,b,c,d) \ast \text{sum4sq-int}(p,q,r,s) = \text{sum4sq-int}(a*p+b*q+c*r+d*s, a*q-b*p-c*s+d*r, a*r+b*s-c*p-d*q, a*s-b*r+c*q-d*p) \)

\[ \langle \text{proof} \rangle \]

**lemma** sum4sq-int-nat-eq: \( \text{sum4sq-nat a b c d} = \text{sum4sq-int}(a,b,c,d) \)

\[ \langle \text{proof} \rangle \]

**lemma** is-sum4sq-int-nat-eq: \( \text{is-sum4sq-nat n} \Rightarrow \text{is-sum4sq-int} \text{int n} \)

\[ \langle \text{proof} \rangle \]

**lemma** is-mult-sum4sq-int: \( \text{is-sum4sq-int x} \Rightarrow \text{is-sum4sq-int y} \Rightarrow \text{is-sum4sq-int} (x \ast y) \)

\[ \langle \text{proof} \rangle \]

**lemma** is-mult-sum4sq-nat: \( \text{is-sum4sq-nat x} \Rightarrow \text{is-sum4sq-nat y} \Rightarrow \text{is-sum4sq-nat} (x \ast y) \)

\[ \langle \text{proof} \rangle \]

**lemma** mult-oddprime-is-sum4sq: \( \exists t. 0 < t \land t < p \land \text{is-sum4sq-int} (p \ast t) \)

\[ \langle \text{proof} \rangle \]

**lemma** zprime-is-sum4sq: \( \text{prime (nat p)} \Rightarrow \text{is-sum4sq-int p} \)

\[ \langle \text{proof} \rangle \]

**lemma** prime-is-sum4sq: \( \text{prime p} \Rightarrow \text{is-sum4sq-nat p} \)

\[ \langle \text{proof} \rangle \]

**theorem** sum-of-four-squares: \( \text{is-sum4sq-nat n} \)

\[ \langle \text{proof} \rangle \]

end

References


