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 -
  assume a: n > 0
  define k where k = a div n
  hence h: a - k * n = a mod n by (simp add: mod-div-mult-eq algebra-simps)
  thus ?thesis
proof (cases 2 * (a mod n) ≤ n)
  case True
  hence 2 * |a - k*n| ≤ n using h pos-mod-sign a by auto
  thus ?thesis by blast
next
  case False
  hence 2 * (n - a mod n) ≤ n by auto
  have a - (k+1)*n = a mod n - n using h by (simp add: algebra-simps)
  hence 2 * |a - (k+1)*n| ≤ n using h pos-mod-bound[of n a] a False by fastforce
  thus ?thesis by blast
qed

private 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)
  unfolding sum2sq-nat-def sum2sq-int-def by simp

private lemma is-sum2sq-int-nat-eq: is-sum2sq-nat n = is-sum2sq-int (int n)
unfolding is-sum2sq-nat-def is-sum2sq-int-def

proof
  assume \( \exists a\ b\ n = \text{sum2sq-nat}\ a\ b \)
  thus \( \exists a\ b\ \text{int}\ n = \text{sum2sq-int}\ (a\ b) \) using sum2sq-int-nat-eq by force

next
  assume \( \exists a\ b\ \text{int}\ n = \text{sum2sq-int}\ (a\ b) \)
  then obtain \( a\ b\ \text{where}\ \text{int}\ n = \text{sum2sq-int}\ (a\ b) \) by blast
  hence \( \text{int}\ n = \text{sum2sq-int}\ (\text{int} |a||a|) = \text{sum2sq-int}\ (\text{int} |b||b|) \)
  unfolding sum2sq-int-def by simp
  hence \( \text{int}\ n = \text{int} (\text{sum2sq-nat} (\text{nat}|a||a|) (\text{nat}|b||b|)) \)
  using sum2sq-int-nat-eq by presburger
  thus \( \exists a\ b\ n = \text{sum2sq-nat}\ a\ b \) by auto

qed

private lemma product-two-squares-aux: \( \text{sum2sq-int}(a\ b) * \text{sum2sq-int}(c\ d) = \text{sum2sq-int}(a*c - b*d, a*d + b*c) \)
  unfolding power2-eq-square sum2sq-int-def by (simp add: algebra-simps)

private lemma product-two-squares-int: \( \text{is-sum2sq-int}\ m \Longrightarrow \text{is-sum2sq-int}\ n \)
  by (unfold is-sum2sq-int-def; auto simp only: product-two-squares-aux, blast)

private lemma product-two-squares-nat: \( \text{is-sum2sq-nat}\ m \Longrightarrow \text{is-sum2sq-nat}\ n \)
  using product-two-squares-int is-sum2sq-int-nat-eq by simp

private lemma sots1-aux:
  assumes prime \( \langle 4* k + 3 \rangle \)
  assumes odd \( \langle \text{multiplicity}\ (4* k + 3)\ n \rangle \)
  shows \( \sim \text{is-sum2sq-nat}\ n \)
proof
  assume \( \text{is-sum2sq-nat}\ n \)
  then obtain \( a\ b\ \text{where}\ h1: n = a'^2 + b'^2 \)
  unfolding is-sum2sq-nat-def sum2sq-nat-def
  by blast
  have \( ab-nz: a \neq 0 \vee b \neq 0 \) by (rule ccontr) (insert assms, auto simp: h1)
  let \( ?p = 4* k + 3 \)
  let \( ?g = \text{gcd}\ a\ b \)
  have \( h2: \ ?g \neq 0 \) using assms(2) h1 odd-pos by fastforce
  then obtain \( a'\ b'\ \text{where}\ h3: a = a' * \ ?g\ b = b' * \ ?g\ \text{coprime} a'\ b' \)
    using gcd-coprime-exists by blast
    have \( ?g' \neq 0 \) using dvd-dvd add h1 by auto
  then obtain \( m\ \text{where}\ h4: m = \ ?g' \ 
  using dvd-dvd-mult-self by blast
  also have \( \ldots = (a' \ ?g)^\ast 2 + (b' \ ?g)^\ast 2 \) unfolding h1 using h3 by presburger
  also have \( \ldots = ?g' \ast a'^\ast 2 + ?g' \ast b'^\ast 2 \) unfolding power2-eq-square by simp
  finally have \( ?g' \neq 0 \) by (simp add: distrib-left mult.commute)
  hence \( h5: m = a'^\ast 2 + b'^\ast 2 \) using h2 by auto
  let \( ?mul = \text{multiplicity}\ ?p \ ?g \)
  have \( \text{multiplicity}\ ?p\ (\ ?g'^2 \) = \ ?mul\ + \ ?mul \)
    unfolding power2-eq-square using h2 assms
    by (subst prime-elem-multiplicity-mult-distrib) simp-all
  hence \( \text{even} (\text{multiplicity}\ ?p\ (\ ?g'^2 \)) \) by simp
  moreover have \( m \neq 0 \) using assms(2) h4 odd-pos by fastforce
  ultimately have \( \text{odd} (\text{multiplicity}\ ?p\ m) \)
using assms ab-nz by (simp-all add: h4 [symmetric] prime-elem-multiplicity-mult-distrib)
hence \(? p \vdash m\) using not-ded-imp-multiplicity-0 by force
hence \(? p \vdash a^\prime 2 + b^\prime 2\) using h5 by auto
{
  assume \(? p \vdash a^\prime 2\)
  moreover hence \(? p \vdash b^\prime 2\) using h6 dvd-add-iff by blast
ultimately have \(? p \vdash a^\prime \) \(\vdash p \vdash b^\prime\) using assms(1) prime-dvd-power-nat by blast+
hence False
  using assms(1) h3(3) coprime-common-divisor-nat[of \(a^\prime\) \(b^\prime\) \(? p\)] not-prime-1 by linarith
}
hence \(? p \vdash a^\prime 2\) ..
hence h7: \(? p \vdash a^\prime\) using assms(1)
  by (simp add: power2-eq-square prime-imp-coprime)
hence coprime \(? p \vdash a^\prime\)
  using assms(1) by (simp add: prime-imp-coprime)
thm prime-imp-coprime-nat
moreover have \(a^\prime \neq 0\) using h7 dvd-0-right[of \(? p\)] by meson
ultimately obtain \(a^\prime\) aux where \(a^\prime * a^\prime = \vdash p * aux + 1\)
  using bezout-nat[of \(a^\prime\) \(? p\)]
  by (auto simp: ac-simps)
hence \([a^\prime * a^\prime = 1]\ mod \(? p\)\) using cong-to-1'-nat by auto
from cong-mult [OF this this] have h11: \([1 = (a^\prime 2 + a^\prime)2]\ mod \(? p\)\)
  unfolding power2-eq-square by (simp add: algebra-simps cong-sym)
let \(\vdash b^\prime diva = a^\prime * b^\prime\)
have \(a^\prime 2 + (a^\prime 2 + b^\prime 2) = 0\) \(\mod \(? p\)\)
  using h6 cong-dvd-modulus-nat cong-multiplicity-right by blast
from cong-add [OF h11 this] have \([1 + a^\prime 2 * b^\prime 2 = 0]\ \(\mod \(? p\)\)\)
  unfolding add-mult-distrib2 using cong-add-cancel-nat[of \(a^\prime 2 + a^\prime 2\) \(\vdash p\)\]
  by fastforce
hence h8: \(\vdash b^\prime diva 2 + 1 = 0\) \(\mod \(? p\)\) by (simp add: power-mult-distrib)
{
  assume \(? p \vdash \vdash b^\prime diva\)
  hence \(? p \vdash \vdash b^\prime diva 2\) by (simp add: assms(1) prime-dvd-power-nat-iff)
hence \(\vdash b^\prime diva 2 = 0\) \(\mod \(? p\)\) using cong-aldldef-nat by auto
hence \(\vdash b^\prime diva 2 + 1 = 1\) \(\mod \(? p\)\) using cong-add-cancel-0-nat by blast
from this h8 have \([0 = 1]\ \(\mod \(? p\)\)\)
  using cong-sym cong-trans by blast
hence \(? p \vdash 1\) using cong-0-1-nat by auto
hence False using assms(1) by simp
}
hence \(? p \vdash \vdash b^\prime diva\) ..
hence h9: \(\vdash b^\prime diva \vdash p - 1 = 1\) \(\mod \(? p\)\)
  using assms(1) fermat-theorem [of \(? p\) \(\vdash b^\prime diva\)] by simp
have h10: \(? p \geq 3\) by simp
have h11: \(\vdash b^\prime diva \vdash 4 * k + 2 = 1\) \(\mod \(? p\)\) using h9 by auto
have \([\vdash b^\prime diva 2 + 1] = 0\) \(\mod \(? p\)\) using h8 cong-pow [of \(\vdash b^\prime diva \vdash 2 + 1 0 \vdash p\) 2]
  by auto
moreover have \(\vdash b^\prime diva \vdash 2 * 4 = (\vdash b^\prime diva \vdash 2) \vdash 2\) by auto
hence \(\vdash b^\prime diva \vdash 2 + \vdash b^\prime diva \vdash 2 + 1\)
  by (auto simp: algebra-simps power2-2-eq-square)
ultimately have \(\vdash b^\prime diva \vdash 4 + \vdash b^\prime diva \vdash 2 + \vdash b^\prime diva \vdash 2 = 1\) \(\mod \(? p\)\) by simp
moreover have \(\vdash b^\prime diva \vdash 4 + \vdash b^\prime diva \vdash 2 + (\vdash b^\prime diva \vdash 2 + 1) = \vdash b^\prime diva \vdash 4 + \vdash b^\prime diva \vdash 2 + 0\)
(mod ?p)
  using h8 cong-add-rcancel-nat by blast
ultimately have [?bdiva^4 + ?bdiva^2 = 0] (mod ?p) by (simp add: cong-def)
hence [?bdiva^4 + ?bdiva^2 + 1 = 0 + 1] (mod ?p) using cong-add-rcancel-nat by blast
moreover have [?bdiva^4 + (?bdiva^2 + 1) = ?bdiva^4 + 0] (mod ?p)
  using h8 cong-add-rcancel-nat by blast
ultimately have [?bdiva^4 = 1] (mod ?p) by (simp add: cong-def)
hence [(?bdiva^4)^k = 1^k] (mod ?p) using cong-pow by blast
hence h12: [?bdiva^(4*k) = 1] (mod ?p) by (simp add: power-mult)
hence h13: [?bdiva^(4*k) + (?bdiva^2 + 1) = 1*(?bdiva^2 + 1)] (mod ?p)
  using cong-scalar-right by blast
have (?bdiva^4*(?bdiva^2 + 1) = ?bdiva^4*(4*k+2)+?bdiva^2*(4*k)
    unfolding add-mult-distrib2 power-add by simp
  using h13 unfolding nat-mult-1 by presburger
moreover have [(?bdiva^4*(4*k+2)+?bdiva^2*(4*k) = ?bdiva^2 + 1)] (mod ?p)
  using h11 h12 cong-add by blast
ultimately have [?bdiva^2 + 1 = 2] (mod ?p)
by (auto simp add: cong-def)
hence [?L = 2] (mod ?p) using h8 by (simp add: cong-def)
then have ?p dvd 2 by (auto dest: cong-dvd-iff)
then show False
  by (auto dest: dvd-imp-le)
qd

private lemma sots1: assumes is-sum2sq-nat n
  shows \ A k. prime (4*k+3) \rightarrow even (multiplicity (4*k+3) n)
using sots1-aux assms by blast

private lemma aux-lemma: assumes [(a::nat) = b] (mod c) b < c
  shows \ A k. a = c*k + b
proof -
  have a mod c = b using assms by (simp add: cong-def mod-if)
  hence b \le a using assms by auto
  thus \thesis using cong-le-nat assms(1) by auto
qd

private lemma Legendre-1mod4*: prime (4*k+1::nat) \rightarrow (Legendre (-1) (4*k+1)) = 1
proof -
  let ?p = 4*k+1
  let ?L = Legendre (-1) ?p
  assume p: prime ?p
from p have k \neq 0 by (intro notI) simp-all
hence p2: ?p > 2 by simp
with p have [?L = (-1)^((?p - 1) div 2)] (mod ?p)
  by (rule euler-criterion)
hence [?L = (-1)^((2 * nat k))] (mod ?p) by auto
hence [?L = 1] (mod ?p) unfolding power-mult by simp
moreover have ?L=1 \lor ?L=0 \lor ?L=-1 by (simp add: Legendre-def)
ultimately have \(?L = 1 \lor \not \exists \ p \ d \ V \ ?p \ d \ (2::int)\) by auto
moreover
{ assume \(?p \ d \ V \ ?p \ d \ (2::int)\)
  with \(p2\) have \(False\) by (auto simp add: zdvd-not-zless) }
ultimately show \(?\text{thesis}\) by auto
qed

private lemma qf1-prime-exists: \(\text{prime (}4\ast k+1) \Rightarrow \text{is-sum2sq-nat (}4\ast k+1)\)
proof –
  let \(?p = 4\ast k+1\)
  assume \(p: \text{prime} \ ?p\)
  hence \(\text{Legendre (-1) \ ?p = 1}\) by (rule \(\text{Legendre-1mod4}\))
moreover
{ assume \(\neg \text{QuadRes} \ ?p (-1)\)
  hence \(\text{Legendre (-1) \ ?p \neq 1}\) by (unfold \(\text{Legendre-def}\), auto) }
ultimately have \(\text{QuadRes} \ ?p (-1)\) by auto
then obtain \(s1\) where \(s1: \{s1\ast2 = -1\}\) (mod \(?p\)) by (auto simp add: QuadRes-def)
hence \(s1\ast2 + 1 = 0\) (mod \(?p\)) by (simp add: cong-altdef-int)
define \(s2\) where \(s2 = \text{nat} \ s1\)
hence \(\text{int} \ (s2\ast2 + 1) = s1\ast2 + 1\) by auto
with \(s1\) have \(\text{int} \ (s2\ast2 + 1) = 0\) (mod \(?p\)) by presburger
hence \(s2: \{s2\ast2 + 1 = 0\}\) (mod \(?p\))
  using \(\text{cong-int-iff}\) by fastforce
from \(p\) have \(p0: \ ?p > 0\) by simp
then obtain \(s\) where \(s0p: 0 \leq s \land s < \ ?p \land [s2 = s]\) (mod \(?p\))
  using \(\text{cong-less-unique-nat}[of \ ?p]\) by fastforce
then have \(s2 = s2\ast2\) (mod \(?p\))
  by (simp add: cong-sym cong-pow)
with \(s2\) have \(s: \{s\ast2 + 1 = 0\}\) (mod \(?p\))
  using \(\text{cong-trans cong-add-rzcancel-nat}\) by blast
hence \(?p \ d \ s\ast2 + 1\) using \(\text{cong-altdef-nat}\) by auto
then obtain \(t\) where \(t: \ s2\ast2 + 1 = \ ?p\ast t\) by (auto simp add: dvd-def)
hence \(?p\ast t = \text{sum2sq-nat} \ s I\) by (simp add: sum2sq-nat-def)
qf1pt: \(\text{is-sum2sq-nat} \ (?p\ast t)\) by (auto simp add: is-sum2sq-nat-def)
have \(t-l-p: \ t < ?p\)
proof (rule \(\text{ccontr}\))
  assume \(\neg \ t < \ ?p\)
  hence \(t > \ ?p - 1\) by simp
  with \(p0\) have \(\text{?p}*(\ ?p - 1) < \ ?p\ast t\) by (simp only: multi-less-mono2)
  also with \(t\) have \(\ldots = s\ast2 + 1\) by simp
  also have \(\ldots \leq \ ?p*(\ ?p - 1) - \ ?p + 2\)
proof –
  from \(s0p\) have \(s \leq \ ?p - 1\) by (auto simp add: less-le)
  with \(s0p\) have \(s\ast2 \leq (\ ?p - 1)\ast2\) by (simp only: power-mono)
  also have \(\ldots = \ ?p*(\ ?p - 1) - 1*(\ ?p - 1)\) by (simp only: power2-eq-square)
diff-mult-distrib
finally show \(?\text{thesis}\) by auto
qed
finally have \(?p < 2\) by simp
with \(p\) show \(False\) by (unfold \(\text{prime-def}\), auto)
qed
have \(tpos: \ t \geq 1\)
proof (rule ccontr)
assume \(- t \geq 1\)
hence \(t < 1\) by auto
moreover
\{ assume \(t = 0\) with \(t\) have \(s^2 + 1 = 0\) by simp \}
moreover
\{ assume \(t < 0\)
  with \(p0\) have \(?p*t < ?p*0\) by (simp only: zmult-less_mono2)
  with \(t\) have \(s^2 + 1 < 0\) by auto \}
moreover have \(s^2 \geq 0\) by (simp only: zero_le_power2)
ultimately show False by (auto simp add: less_le)
qed
moreover
\{ assume \(t1: t > 0\)
then obtain \(tn\) where \(tn: tn = t - 1\) by auto
have is_sum2sq_nat \((?p*(1 + 0))\) (is \(?Q 0\))
  — So, \(Q n\) there exist \(x, y\) such that \(x^2 + y^2 = (p \times (1 + int(n)))\)
proof (rule ccontr)
assume \(nQ1: \neg ?Q 0\)
have \((1 + tn) < ?p \implies \neg ?Q tn\)
proof (induct \(tn\) rule: infinite_descent0)
case 0
  from \(nQ1\) show \(1 + 0 < ?p \implies \neg ?Q 0\) by simp
next
case \((\text{smaller } n)\)
hence \(n0: n > 0\) and \(IH: 1 + n < ?p \land ?Q n\) by auto
then obtain \(x\ y\) where \(x^2 + y^2 = int \((?p*(1 + n))\)\)
  using is_sum2sq_int_nat_eq (unfold is_sum2sq_int_def sum2sq_int_def, auto)
hence \(xy: x^2 + y^2 = (int Pp)*(int \((1 + n))\)\) unfolding of_nat_mul by presburger
let \(n1 = int \((1 + n)\)
from \(n0\) have \(n1pos: n1 > 0\) by simp
then obtain \(r\ v\) where \(rv: v = x - r * n1 \land 2 * |v| \leq n1\)
  by (frule_tac n=\(n1\) in best_division_abs, auto)
from \(n1pos\) obtain \(s\ w\) where \(sw: w = y - s * n1 \land 2 * |w| \leq n1\)
  by (frule_tac n=\(n1\) in best_division_abs, auto)
let \(C = v^2 + w^2\)
have \(n1 dvd C\)
proof
  from \(rv\ sw\) have \(C = (x - r * n1)^2 + (y - s * n1)^2\) by simp
  also have \(\ldots = x^2 + y^2 - 2 * x * (r * n1) - 2 * y * (s * n1) + (r * n1)^2 + (s * n1)^2\)
  unfolding power2_eq_square by (simp add: algebra_simps)
  also with \(xy\) have \(\ldots = n1 * p - n1 * (2 * x * r) - n1 * (2 * y * s) + n1 * r^2 + s^2\)
  + \(n1^2 * 2^* 2\)
    by (simp only: ac_simps power_mult_distrib)
  finally show \(C = n1 * (\(p - 2 * x * r - 2 * y * s + n1 * (r^2 + s^2))\))
    by (simp only: power_mult_distrib distrib_left ac_simps
      left_diff_distrib right_diff_distrib power2_eq_square)
qed
then obtain \(m1\) where \(C = n1 * m1\) by (auto simp add: dvd_def)
have \(mm: m1 < n1\)
proof (rule ccontr)
assume \(- m_1 < \?n_1\) hence \(\?n_1 - m_1 \leq 0\) by simp

hence \(4 * \?n_1 - 4 * m_1 \leq 0\) by simp

with \(n_1\) have \(2 * \?n_1 - 4 * m_1 < 0\) by simp

with \(n_1\) have \(\?n_1 + (2 * \?n_1 - 4 * m_1) < \?n_1 * 0\) by (simp only: zmult_zless_mono2)

hence \(\?n_1 + (2 * \?n_1 - 4 * m_1) < 0\) by simp

have \(hlp: 2 * |v| \geq 0 \land 2 * |w| \geq 0\) by simp

from \(m_1\) have \(4 * \?n_1 \cdot m_1 = 4 * v^2 + 4 * w^2\) by arith

also have \(\ldots = (2 * |v|)^2 + (2 * |w|)^2\)

by (auto simp add: power_mult_distrib)

also from \(rv\) \(hlp\) have \(\ldots \leq \?n_1^2 + (2 * |w|)^2\)

using power_mono [of \(2 * |b|\) \(1 + \text{int n 2 for b}\) by auto]

also from \(sw\) \(hlp\) have \(\ldots \leq \?n_1^2 + ?n_1^2\)

using power_mono [of \(2 * |b|\) \(1 + \text{int n 2 for b}\) by auto]

finally have \(\?n_1 \cdot m_1 \cdot 4 \leq \?n_1 \cdot \?n_1 \cdot 2\) by (simp add: power2_eq-square ac_simps)

hence \(?n_1 \cdot (2 * \?n_1 - 4 * m_1) \geq 0\) by (simp only: right_diff_distrib ac_simps)

with \(\text{contr}\) show \(False\) by auto

qed

have \(\?p \cdot s \cdot m_1 = (r \cdot s \cdot v + s \cdot s \cdot w + m_1)^2 + (r \cdot w - s \cdot v)^2\)

proof –

from \(m_1\) \(xy\) have \((\?p \cdot s \cdot ?n_1) \cdot ?C = (x \cdot 2 + y \cdot 2) \cdot (v \cdot 2 + w \cdot 2)\) by simp

also have \(\ldots = (x \cdot s \cdot v + y \cdot s \cdot w)^2 + (x \cdot w - y \cdot v)^2\)

by (simp add: eval_nat_numeral field_simps)

also with \(rv\) \(sw\) \(hlp\) have \(\ldots = ((r * \?n_1 + v) \cdot s \cdot v + (s * ?n_1 + w) \cdot s \cdot v)^2 + ((r * \?n_1 + v) \cdot w - (s * \?n_1 + w) \cdot s \cdot v)^2\)

by simp

also have \(\ldots = ((?n_1 \cdot (r \cdot s \cdot v) + \?n_1 \cdot (s \cdot s \cdot w) + (v \cdot 2 + w \cdot 2)) \cdot 2 + (\?n_1 \cdot (r \cdot w) - \?n_1 \cdot (s \cdot s \cdot v))^2\)

by (simp add: eval_nat_numeral field_simps)

also from \(m_1\) \(hlp\) have \(\ldots = ((?n_1 \cdot (r \cdot s \cdot v) + \?n_1 \cdot (s \cdot s \cdot w) + \?n_1 \cdot m_1)^2 + (\?n_1 \cdot (r \cdot w) - \?n_1 \cdot (s \cdot s \cdot v))^2\)

by simp

finally have \((\?p \cdot ?n_1) \cdot ?C = \?n_1 \cdot 2 \cdot (r \cdot v + s \cdot s \cdot w + m_1)^2 + \?n_1 \cdot 2 \cdot (r \cdot w - s \cdot v)^2\)

by (simp add: eval_nat_numeral field_simps)

with \(m_1\) \(hlp\) have \(\?n_1 \cdot 2 \cdot (r \cdot s \cdot m_1) = \?n_1 \cdot 2 \cdot ((r \cdot v + s \cdot s \cdot w + m_1)^2 + (r \cdot w - s \cdot v)^2)\)

by (simp only: ac_simps power2_eq_square, simp add: distrib_left)

hence \(\?n_1 \cdot 2 \cdot (r \cdot v + s \cdot s \cdot w + m_1)^2 - (r \cdot w - s \cdot v)^2 = 0\)

by (auto simp add: distrib_left right_diff_distrib)

moreover from \(n_1\) \(hlp\) have \(\?n_1 \cdot 2 \neq 0\) by (simp add: power2_eq_square)

ultimately show \(\text{thesis}\) by simp

qed

hence \(gf1pm1: is\-sum2sq\-int\ ((\text{int } ?p) \cdot m_1)\)

by (unfold is_sum2sq_int_def sum2sq_int_def, auto)

have \(n1\) \(pm1: m_1 > 0\)

proof –

{ assume \(v \cdot 2 + w \cdot 2 = 0\)

hence \(v = 0 \land w = 0\) using sum-power2_eq_zero_iff by blast

with \(rv\) \(sw\) \(hlp\) have \(\?n_1 \cdot d \cdot x \land \?n_1 \cdot d \cdot y\) by (unfold dvd_def, auto)

hence \(\?n_1 \cdot 2 \cdot d \cdot x \cdot 2 \land \?n_1 \cdot 2 \cdot d \cdot y \cdot 2\) by simp

hence \(\?n_1 \cdot 2 \cdot d \cdot x \cdot 2 + y \cdot 2\) by (simp only: dvd_add)

with \(xy\) have \(\?n_1 \cdot \?n_1 \cdot d \cdot \?n_1 \cdot \?p\) by (simp only: power2_eq_square ac_simps)
moreover from \( n1pos \) have \(?n1 \neq 0 \) by simp
ultimately have \(?n1 dvd ?p \) by (rule zdvd-mult-cancel)
with \( n1pos \) have \(?n1 \geq 0 \land ?n1 dvd ?p \) by simp
with \( p \) have \(?n1 = 1 \lor ?n1 = ?p \) unfolding prime-nat-iff by presburger
with \( IH \) have \(?Q 0 \) by auto
with \( nQ1 \) have False by simp }
moreover
\{
  assume \( v^2 + 1 \cdot w^2 \neq 0 \)
moreover have \( v^2 + w^2 \geq 0 \) by simp
ultimately have \( vwpos : v^2 + w^2 > 0 \) by arith
with \( m1 \) have \( m1 \neq 0 \) by auto
moreover have \( m1 \geq 0 \)
proof (rule ccontr)
  assume \( \neg m1 \geq 0 \)
  hence \( m1 < 0 \) by simp
  with \( n1pos \) have \(?n1 \cdot m1 < ?n1 \cdot 0 \) by (simp only: zmult-zless-mono2)
  with \( m1 vwpos \) show False by simp
qed
ultimately have \( ?thesis \) by auto }
ultimately show \( ?thesis \) by auto
qed

hence \( 1 + \text{int}((\text{nat} m1) - 1) = m1 \) by arith
with \( qf1pm1 \) have \( Qm1 : ?Q ((\text{nat} m1) - 1) \)
  using is-sum2sq-int-nat-eq by (simp add: algebra-simps)
then obtain \( mm \) where \( \text{tmp}: mm = (\text{nat} m1) - 1 \land ?Q mm \) by auto
moreover have \( mm < n \) using \( \text{tmp} \) \( \text{mn} \) \( m1pos \) by arith
moreover with \( IH \) have \( 1 + \text{int} \ mm < ?p \) by auto
ultimately show \( \neg case \) by auto
qed

hence \( \neg \text{is-sum2sq-nat} (\?p \cdot t) \) using \( \text{tn} \) \( \text{tpos} \) \( \text{t-l-p} \) by auto
with \( qf1pt \) show False by simp
qed

hence \( ?thesis \) by auto }
ultimately show \( ?thesis \) by (auto simp add: less-le)
qed

private lemma \( \text{fermat-two-squares: assumes} \) prime \( p \) \( (\neg [p = 3] \mod 4) \)
shows is-sum2sq-nat \( p \)
proof (cases \( p = 2 \))
case True
  have \( (2::\text{nat})=1^2+1^2 \) using power2-eq-square by simp
  thus \( ?thesis \) unfolding is-sum2sq-nat-def sum2sq-nat-def using True by fast
next
case False
  hence \( p > 2 \) using \( \text{assms(1)} \) unfolding prime-nat-iff by auto
  hence \( h1: \text{odd} \ p \) using \( \text{assms(1)} \) \( \text{prime-odd-nat} \) by simp
  hence \( h2: \neg [p = 0] \mod 4) \) unfolding cong-def by fastforce
  have \( h3: \neg [p = 2] \mod 4) \) using \( h1 \) cong-ded-iff [of \( p \ 2 \)] cong-ded-modulus-nat by auto
  obtain \( x \) where \( h4: [p = x] \mod 4) \land x < 4 \) by (meson cong-less-unique-nat zero-less-numeral)
  from \( h1 \) \( h2 \) \( h3 \) \( h4 \) \( \text{assms} \) have \( x \neq 0 \land x \neq 2 \land x \neq 3 \land x < 4 \) by meson
  hence \( x = 1 \) by linarith
from this h4 have \( p = 1 \) (mod 4) by simp
then obtain \( k \) where \( p = 4\cdot k + 1 \) using aux-lemma by fastforce
thus \( \neg \)thesis using qf1-prime-exists assms by blast
qed

private lemma sots2: assumes \( \forall k. \text{prime} (4 \cdot k + 3) \rightarrow \text{even} \) (multiplicity \( (4\cdot k + 3) \))
shows is-sum2sq-nat \( n \) using assms
proof (induction \( n \) rule: nat-less-induct)
case (1 \( n \))
thus \( \neg \)thesis
proof (cases \( n > 1 \))
case False
thus \( \neg \)thesis
proof (cases \( n = 1 \))
case True
have \( (1::nat) = 0^2 + 1^2 \) by (simp add: power2-eq-square)
thus \( \neg \)thesis using True unfolding is-sum2sq-nat-def
sum2sq-nat-def by blast
next
case False
hence \( n = 0 \) using \( \neg \) by simp
moreover have \( (0::nat) = 0^2 + 0^2 \) by (simp add: power2-eq-square)
ultimately show \( \neg \)thesis unfolding is-sum2sq-nat-def
sum2sq-nat-def by blast
qed
next
case True
then obtain \( p \) \( m \) where h1: prime \( p \land n = p \cdot m \) using prime-divisor-exists[of \( n \)]
by (auto elim: dvdE)
with True have m-nz: \( m \neq 0 \) by (intro notI)
assume a1: \( [p = 3] \) (mod 4)
then obtain kp where \( p = 4\cdot kp + 3 \) using aux-lemma by fastforce
hence even (multiplicity \( p \) \( n \)) using 1.prems h1 by auto
moreover have multiplicity \( p \) \( n \) \( \neq 0 \) using h1 True m-nz
by (subst multiplicity-eq-zero-iff) (auto simp: prime-gt-0-nat)
ultimately have h3: multiplicity \( p \) \( n \) \( \geq 2 \) by presburger
have \( p \) dvd \( m \)
proof (rule ccontr)
assume a2: \( \neg p \) dvd \( m \)
hence multiplicity \( p \) \( m \) \( = 0 \) by (rule not-dvd-impl-multiplicity-0)
moreover from h1 have multiplicity \( p \) \( p = 1 \) by (intro multiplicity-prime) auto
moreover have \( m > 0 \) using h1 True by (cases \( m = 0 \)) simp-all
ultimately have multiplicity \( p \) \( n \) \( = 1 \) using h1
using prime-elem-multiplicity-mult-distrib [of \( p \) \( p \) \( m \)] m-nz prime-gt-0-nat
by auto
thus \( \neg \)thesis using h3 by simp
qed
then obtain \( m' \) where h4: \( m = p \cdot m' \) using dvdE by blast
with h1 have h5: \( n = p^2 \cdot m' \) by (simp add: power2-eq-square)
have h6: \( m' < n \)
using dual-order.strict-trans h1 h2 h4 nat-mult-less-cancel1 prime-gt-0-nat[of p]

by blast

have ∃k. prime (4*k + 3) ⇒ even (multiplicity (4*k + 3) m')

proof −

  fix kq::nat
  let ?q = 4*kq + 3
  assume a2: prime ?q

  { assume p: p=?q
    hence h7: multiplicity ?q (p^2) = 2 using h1
    by (auto intro!: multiplicity-prime-power)
  have even (multiplicity ?q n) using 1(2)[of kq] a2 by blast
  also note h5
  also from p h1 h4 m-nz
  have multiplicity (4 * kq + 3) (p^2 * m') =
    Suc (Suc (multiplicity (4 * kq + 3) m'))
    by (subst prime-elem-multiplicity-mult-distrib) auto
  finally have even (multiplicity ?q m') by simp

  moreover
  { assume p≠?q
    from a2 h4 m-nz have multiplicity ?q n =
      multiplicity (4 * kq + 3) (p^2) + multiplicity (4 * kq + 3) m'
    unfolding h5 by (subst prime-elem-multiplicity-mult-distrib) simp-all
    also from p ≠ ?q; a2 h1 have multiplicity ?q (p^2) = 0
    by (intro multiplicity-distinct-prime-power) simp-all
    finally have multiplicity ?q n = multiplicity ?q m' by simp
    moreover have even (multiplicity ?q n) using 1(2)[of kq] a2 by blast
    ultimately have even (multiplicity ?q m') by simp
  }
  ultimately show even (multiplicity ?q m') by blast

  hence is-sum2sq-nat m' by (simp add: 1 h6)
  moreover have p^2 = p^2 + 0^2 by simp
  hence is-sum2sq-nat (p^2) unfolding is-sum2sq-nat-def sum2sq-nat-def by blast
  ultimately have ?thesis using product-two-squares-nat h5 by blast

  moreover
  { assume a1: ¬ [p = 3] (mod 4)
    have ∃k. prime (4*kq+3) ⇒ even (multiplicity (4*kq+3) m)
    proof −
      fix kq
      let ?q = 4*(kq::nat) + 3
      assume a2: prime ?q
      { assume p = ?q
        then have False using a1 cong-add-rcancel-0-nat [of 4 * kq 3 4]
        by (auto simp add: cong-def)
      }
      hence p≠?q ..
    }

    have n = p * m using h1 by simp
    also from h1 a2 m-nz have multiplicity ?q .. =
multiplicity \((4 \cdot kq + 3)\) \(p\) + multiplicity \((4 \cdot kq + 3)\) \(m\)

by (substitution prime-element-multiplicity-mult-distrib) (simp-all add: prime-gl-0-nat)

also from \(p \neq q\) \(a2 \cdot h1\) have multiplicity \(q \cdot p = 0\)

by (intro prime-multiplicity-other) simp-all

finally have multiplicity \(q \cdot n\) = multiplicity \(q \cdot m\) by simp

ultimately show even (multiplicity \(q \cdot n\)) using \(1\) \(2\) \([\text{of } kq]\) \(a2\) by blast

qed

hence is-sum2sq-nat \(m\) by (simp add: \(1\) \(h2\))

moreover have is-sum2sq-nat \(p\) using \(\text{fermat-two-squares}\) \(a1\) \(h1\) by blast

ultimately have \(?\text{thesis}\) using \(\text{product-two-squares-nat}\) \(h1\) by blast

ultimately show \(?\text{thesis}\) by blast

qed

theorem sum-of-two-squares:

is-sum2sq-nat \(n\) \(\longleftrightarrow\) (\(\forall\ k.\ \text{prime} (4\cdot k+3)\ \longrightarrow\ \text{even} (\text{multiplicity} \ (4\cdot k+3)\ \cdot \ n)\))

using \(\text{sots1[of } n]\) \(\text{sots2[of } n]\) by blast

private lemma \(k\)-mod-eq: (\(\forall\ p::\text{nat.}\ \text{prime} p \land [p = 3] \pmod{4} \longrightarrow P \ p\) = (\(\forall\ k.\ \text{prime} (4 \cdot k+3) \longrightarrow P \ (4 \cdot k+3)\))

proof

assume \(a1::\forall\ p.\ \text{prime} p \land [p = 3] \pmod{4} \longrightarrow P \ p\)

{ fix \(k::\text{nat}\)

  assume prime \((4 \cdot k + 3)\)

  moreover hence \([4 \cdot k+3 = 3] \pmod{4}\)

  by (simp add: \(\text{cong-add-cancel-rcancel-0-nat}\) \(\text{cong-mult-self-left}\))

  ultimately have \(P \ (4 \cdot k + 3)\) using \(a1\) by blast

}

thus \(\forall\ k.\ \text{prime} \ (4 \cdot k + 3) \longrightarrow P \ (4 \cdot k + 3)\) by blast

next

assume \(a1::\forall\ k.\ \text{prime} \ (4 \cdot k + 3) \longrightarrow P \ (4 \cdot k + 3)\)

{ fix \(p::\text{nat}\)

  assume prime \(p\) \([p = 3] \pmod{4}\)

  moreover with aux-lemma obtain \(k\) where \(p = 4 \cdot k + 3\) by fastforce

  ultimately have \(P \ p\) using \(a1\) by blast

}

thus \(\forall\ p.\ \text{prime} \ p \land [p = 3] \pmod{4} \longrightarrow P \ p\) by blast

qed

theorem sum-of-two-squares':

is-sum2sq-nat \(n\) \(\longleftrightarrow\) (\(\forall\ p.\ \text{prime} p \land [p = 3] \pmod{4} \longrightarrow \text{even} (\text{multiplicity} \ p \ n)\))

using sum-of-two-squares \(\text{k-mod-eq}\) by presburger

theorem sum-of-two-squares-prime: assumes prime \(p\)

shows \(\text{is-sum2sq-nat}\) \(p\) \([p \neq 3] \pmod{4}\)

proof (cases \(p = 3\) \(\pmod{4}\))

case \(\mathbf{True}\)

have odd (multiplicity \(p\) \(p\)) using \(\text{assms}\) by simp
hence \(\neg (\text{is-sum}\text{-}\text{sq}-\text{nat } p)\) using assms True sum-of-two-squares' by blast
with True show \?thesis by simp
qed (simp add: fermat-two-squares assms)

\begin{document}

\section{Lagrange’s four-square theorem}

theory FourSquares
imports HOL\textendash{}Number-Theory.Number-Theory
begin

context

fixes sum4sq-nat :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat \Rightarrow nat
defines sum4sq-nat \(a \ b \ c \ d\) \(\equiv a^2 + b^2 + c^2 + d^2\)

fixes is-sum4sq-nat :: nat \Rightarrow bool
defines is-sum4sq-nat \(n\) \(\equiv (\exists \ a \ b \ c \ d. \ n = \text{sum4sq-nat } a \ b \ c \ d)\)

begin

private lemma best-division-abs:\(\forall n::\text{int}>0 \implies \exists k. 2*|a - k*n| \leq n\)
proof –
assume \(a; n>0\)
define \(k\) where \(k = a \text{ div } n\)
have \(h; a - k*n = a \text{ mod } n\) by (simp add: div-mult-mod-eq algebra-simps k-def)
thus \?thesis
proof (cases \(2* (a \text{ mod } n) \leq n\))
  case True
  hence \(2*|a - k*n| \leq n\) using \(h\) pos-mod-sign \(a\) by auto
  thus \?thesis by blast
next
  case False
  hence \(2* (n - a \text{ mod } n) \leq n\) by auto
  have \(a - (k+1)*n = a \text{ mod } n - n\) using \(h\) by (simp add: algebra-simps)
  hence \(2* |a - (k+1)*n| \leq n\) using \(h\) pos-mod-bound[of \(n\) \(a\)] a False by fastforce
  thus \?thesis by blast
qed

shows that all nonnegative integers can be written as the sum of four squares.
The proof consists of the following steps:

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

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

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

**private definition**

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

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

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

by (unfold \text{sum4sq-int-def}, simp add: eval-nat-numeral field-simps)

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

unfolding \text{sum4sq-int-nat-def} \text{sum4sq-int-def} by simp

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

unfolding \text{is-sum4sq-int-nat-def} \text{is-sum4sq-int-def}

proof

assume \(\exists \, a \, b \, c \, d. \, n = \text{sum4sq-nat} \, a \, b \, c \, d\)

thus \(\exists \, a \, b \, c \, d. \, \text{int} \, n = \text{sum4sq-int} \, (a, b, c, d)\) using \text{sum4sq-int-nat-eq} by force

next

assume \(\exists \, a \, b \, c \, d. \, \text{int} \, n = \text{sum4sq-int} \, (a, b, c, d)\)

then obtain \(a \, b \, c \, d\) where \(\text{int} \, n = \text{sum4sq-int} \, (a, b, c, d)\) by blast

hence \(\text{int} \, n = \text{sum4sq-int} \, (\text{int} \, (\text{nat} \, |a|), \text{int} \, (\text{nat} \, |b|), \text{int} \, (\text{nat} \, |c|), \text{int} \, (\text{nat} \, |d|))\)

unfolding \text{sum4sq-int-def} by simp

hence \(\text{int} \, n = \text{int} \, (\text{sum4sq-nat} \, (\text{nat} \, |a|) \, (\text{nat} \, |b|) \, (\text{nat} \, |c|) \, (\text{nat} \, |d|))\)

using \text{sum4sq-int-nat-eq} by presburger

thus \(\exists \, a \, b \, c \, d. \, \text{nat} \, n = \text{sum4sq-nat} \, a \, b \, c \, d\) by auto

qed

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

by (unfold \text{is-sum4sq-int-def}, auto simp only: \text{mult-sum4sq-int}, blast)

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

using \text{is-mult-sum4sq-int} \text{is-sum4sq-int-nat-eq} by simp

**private lemma** \(\text{mult-oddprime-is-sum4sq}::\text{[ prime} \, (\text{nat} \, p); \, \text{odd} \, p \, ] \Rightarrow \exists \, \text{t.} \, \text{0} < \text{t} \land \text{t} < \text{p} \land \text{is-sum4sq-int} \, (p*t)\)

proof

assume \(\text{p1}: \text{prime} \, (\text{nat} \, p)\)

then have \(\text{p2}: \, p > 1 \, \text{and} \, \text{prime} \, p\)
by (simp-all add: prime-int-nat-transfer prime-nat-iff)
assume \(p^2\) \emph{odd} \(p\)
then obtain \(n\) where \(n\): \(p = 2*n + 1\) using \emph{oddE} by blast
with \(p^1\) have \(n^0\): \(n > 0\) by (auto simp add: prime-nat-iff)
let \(?C\) = \{
\(0\ .. n\)
\}
let \(?D\) = \{
\(0\ .. n\)
\}
let \(?f\) = \(\%x. x^2 \mod p\)
let \(?g\) = \(\%x. (-1 - x^2) \mod p\)
let \(?A\) = \(?f\) \(?D\)
let \(?B\) = \(?g\) \(?D\)

have \(\text{finC: finite } \?C\) by simp
have \(\text{finD: finite } \?D\) by simp
from \(p^0\) have \(A\subseteq \?C\) and \(B\subseteq \?C\)
  by (auto simp add: pos-mod-conj)
with \(\text{finC}\) have \(\text{finA: finite } \?A\) and \(\text{finB: finite } \?B\)
  by (auto simp add: finite-subset)
from \(A\subseteq \?C\) \(B\subseteq \?C\) have \(\text{AunB}: \?A \cup \?B \subseteq \?C\) by (rule Un-least)
from \(p^0\) have \(\text{cardC: card } \?C = \text{nat } p\) using \emph{card-atLeastZeroLessThan-int} by blast

have \(\text{cardA: card } \?A = \text{card } \?D\)
proof
  have \(\text{inj-on } \?f\) \(?D\)
proof (unfold \emph{inj-on-def}, auto)
  fix \(x\) \(y\)
  assume \(x^0\): \(0 \leq x\) and \(x^1\): \(x \leq n\) and \(y^0\): \(0 \leq y\) and \(y^1\): \(y \leq n\)
  and \(yp\): \(x^2 \mod p = y^2 \mod p\)
  with \(p^0\) have \(\{\mod p\} = x^2 = y^2\) \emph{cong-def} by blast
  hence \(p \text{ dvd } x^2 - y^2\) \emph{cong-altdef-int} by blast
  hence \(p \text{ dvd } (x+y)^2 - (x-y)^2\) by (simp add: power2-eq-square algebra-simps)
  hence \(p \text{ dvd } x+y \lor p \text{ dvd } x-y\) using \(\text{prime } p\) \(p^0\)
  by (auto dest: \emph{prime-dvd-multD})
moreover
  \{ \assume \(p \text{ dvd } x+y\)
  moreover from \(x^n y^n\) have \(x+y < p\) by auto
  ultimately have \(\sim x+y > 0\) by (auto simp add: dvd-not-zless)
  with \(x^0 y^0\) have \(x = y\) by auto \}\ --- both are zero
moreover
  \{ \assume \(p \text{ dvd } x-y\)
  have \(x = y\)
  proof (rule \emph{constr}, \emph{case-tac} \(x-y \geq 0\))
  assume \(x-y \geq 0\) and \(x \neq y\) \emph{hence} \(x-y > 0\) by auto
  with \(\text{ass}\) have \(\sim x-y < p\) by (auto simp add: dvd-not-zless)
  with \(x^0 y^0 n p^0\) show \(\text{False}\) by auto
next
  assume \(0 \leq x-y\) \emph{hence} \(y-x \geq 0\) by auto
  moreover from \(x^0 y^n p^0\) have \(y-x < p\) by auto
  ultimately have \(\sim p \text{ dvd } y-x\) by (auto simp add: dvd-not-zless)
  moreover from \(\text{ass}\) have \(p \text{ dvd } -(x-y)\) by (simp only: \emph{dvd-minus-iff})
  ultimately show \(\text{False}\) by auto
  qed \}
ultimately show \(x=y\) by auto
qed
with finD show \( \text{thesis} \) by (simp only: inj-on-iff-eq-card)

qed

have cardB: card ?B = card ?D

proof (unfold inj-on-def, auto)

  fix \( x \) \( y \)

  assume \( x0 \): \( 0 \leq x \) \( x n \): \( x \leq n \) \( y0 \): \( 0 \leq y \) \( y n \): \( y \leq n \)

  and \( x y p \) have \((-1 - x^2)^2 = -1 - x^2 \) (mod \( p \)) by (simp only: cong-def)

  hence \( p \) dvd \((-1 - y^2)^2 - (1 - x^2)^2 \) by (simp only: cong-alldef-int)

  moreover have \(-1 - y^2 - (1 - x^2) = x^2 - y^2 \) by arith

  ultimately have \( p \) dvd \( x^2 - y^2 \) by simp

  hence \( p \) dvd \( x + y \) \( x - y \) by (simp add: power2-eq-square algebra-simps)

  with \( p1 \) have \( p \) dvd \( x + y \) \| \( p \) dvd \( x - y \) using \( \text{prime} p \) \( p0 \)

  by (auto dest: prime-dvd-multD)

  moreover

  \{ assume \( p \) dvd \( x + y \)

  moreover from \( x n \) \( y n \) have \( x + y < p \) by auto

  ultimately have \( \neg x + y > 0 \) by (auto simp add: zdvd-not-zless)

  with \( x0 \) \( y0 \) have \( x = y \) by auto \}

  \{ assume \( \text{ass} \): \( p \) dvd \( x - y \)

  have \( x = y \)

  proof (rule ccontr, case-tac \( x - y \geq 0 \))

    assume \( x - y \geq 0 \) \( x \neq y \) hence \( x - y > 0 \) by auto

    with \( \text{ass} \) have \( \neg x - y < p \) by (auto simp add: zdvd-not-zless)

    with \( x n \) \( y n \) \( p0 \) show \( False \) by auto

  next

    assume \( \neg 0 \leq x - y \) hence \( y - x > 0 \) by auto

    moreover from \( x0 \) \( y n \) \( p0 \) have \( y - x < p \) by auto

    ultimately have \( \neg p \) dvd \( y - x \) by (auto simp add: zdvd-not-zless)

    moreover from \( \text{ass} \) have \( p \) dvd \( -(x - y) \) by (simp only: dvd-minus-iff)

    ultimately show \( False \) by auto

  qed \}

  ultimately show \( x = y \) by auto

  with finD show \( \text{thesis} \) by (simp only: inj-on-iff-eq-card)

  qed

have \( ?A \cap ?B \neq {} \)

proof (rule ccontr, auto)

  assume \( \text{ABdisj} \): ?A \( \cap \) ?B = {}

  from cardA cardB cardD have \( 2 + 2*\text{(nat n)} = \text{card} ?A \cap \text{card} ?B \) by auto

  also with finA finB ABdisj have \( \ldots = \text{card} (?A \cup ?B) \)

  by (simp only: card-Un-disjoint)

  also with finC \( \text{UnBsubC} \) have \( \ldots \leq \text{card} ?C \) by (simp only: card-mono)

  also with cardC have \( \ldots = \text{nat p} \) by simp

  finally have \( 2 + 2*\text{(nat n)} \leq \text{nat p} \) by simp

  with \( n \) show \( False \) by arith

  qed

then obtain \( z \) where \( z \in ?A \land z \in ?B \) by auto

then obtain \( x y \) where \( xy \): \( x \in ?D \land y \in ?D \land z = x^2 \) mod \( p \land z = (-1 - y^2) \)
mod $p$ by blast
with $p0$ have $[x^2=\neg1-y^2](mod p)$ by (simp add: cong-def)
hence $p$ dvd $x^2-(\neg1-y^2)$ by (simp only: cong-altdef-int)
moreover have $x^2-(\neg1-y^2)=x^2+y^2+1$ by arith
ultimately have $p$ dvd sum4sq-int($x,y,1,0$) by (auto simp add: sum4sq-int-def)
then obtain $t$ where $t: p*t = sum4sq-int(x,y,1,0)$ by (auto simp only: dvd-def eq-refl)
hence is-sum4sq-int ($p*t$) by (unfold is-sum4sq-int-def, auto)
moreover have $t > 0 \land t < p$ proof
have $x^2 \geq 0 \land y^2 \geq 0$ by simp
hence $x^2+y^2+1 > 0$ by arith
with $t$ have $p*t > 0$ by (unfold sum4sq-int-def, auto)
moreover
{ assume $t < 0$ with $p0$ have $p*t < p*0$ by (simp only: zmult-zless_mono2)
  hence $p*t < 0$ by simp }
moreover
{ assume $t = 0$ hence $p*t = 0$ by simp }
ultimately have $\neg t < 0 \land t \neq 0$ by auto
thus $t > 0$ by simp
from $xy$ have $x^2 \leq n^2 \land y^2 \leq n^2$ by (auto simp add: power-mono)
hence $x^2+y^2+1 \leq 2*n^2+1$ by auto
with $t$ have contr: $p*t \leq 2*n^2+1$ by (simp add: sum4sq-int-def)
moreover
{ assume $t > n+1$
  with $p0$ have $p*(n+1) < p*t$ by (simp only: zmult-zless_mono2)
  with $n$ have $p*t > (2*n+1)*n + (2*n+1)*1$ by (simp only: distrib-left)
  hence $p*t > 2*n+n + n + 2*n + 1$ by (simp only: distrib-right mul1-left)
  with $n0$ have $p*t > 2*n^2 + 1$ by (simp add: power2_eq_square) }
ultimately have $\neg t > n+1$ by auto
with $n0$ show $t < p$ by auto
qed
ultimately show $?thesis$ by blast
qed

private lemma zprime-is-sum4sq: prime $(nat p) \Rightarrow is-sum4sq-int p$
proof (cases)
  assume $p2$: $p=2$
  hence $p = sum4sq-int(1,1,0,0)$ by (auto simp add: sum4sq-int-def)
  thus $?thesis$ by (auto simp add: is-sum4sq-int-def)
next
  assume $\neg p =2$ and $prp: prime (nat p)$
  hence $\neg nat p = 2$ by simp
  with $prp$ have $2 < nat p$ using prime-nat iff by force
moreover with $prp$ have odd $(nat p)$ using prime-odd-nat[of nat p] by blast
ultimately have odd $p$ by (simp add: even-nat iff)
with $prp$ have $\exists t. \ 0<t \land t<p \land is-sum4sq-int (p*t)$ by (rule mult-oddprime_is_sum4sq)
then obtain $a$ $b$ $c$ $d$ $t$ where $pt-sol$: $0<t \land t<p \land p*t = sum4sq-int(a,b,c,d)$
  by (unfold is-sum4sq-int-def, blast)
hence $Qt$: $0<t \land t<p \land (\exists \ a1 \ a2 \ a3 \ a4. \ p*t = sum4sq-int(a1,a2,a3,a4))$
  (is $?Q t$) by blast
have $?Q \ I$
proof (rule ccontr)
assume nQ1: ¬ ?Q 1
have ¬ ?Q t
proof (induct t rule: infinite-descent0-measure[where \( V=\lambda x. (nat\ x) - 1 \)], clarify)
fix \( x \ a \ b \ c \ d \)
assume nat \( x - 1 = 0 \) and \( x > 0 \) and \( s: p*x=sum4sq-int(a,b,c,d) \) and \( x < p \)
moreover hence \( x = 1 \) by arith
ultimately have ?Q 1 by auto
with nQ1 show False by auto
next
fix \( x \)
assume \( 0 < nat\ x - 1 \) and \( \neg\ ?Q x \)
then obtain \( a1 a2 a3 a4 \) where \( as: I<x \land x\ < p \land p*x = sum4sq-int(a1,a2,a3,a4) \)
by auto
have \( \exists \ y. nat\ y - 1 < nat\ x - 1 \land ?Q y \)
proof (cases)
assume evenx: even \( x \)
hence even \( (x*p) \) by simp
with \( as \) have ev1234: even \( (a1^2+a2^2 + a3^2+a4^2) \)
by (auto simp add: sum4sq-int-def ac-simps)
have \( \exists \ b1 b2 b3 b4. p*x=sum4sq-int(b1,b2,b3,b4) \land \) even \( (b1+b2) \land \) even \( (b3+b4) \)
proof (cases)
assume ev12: even \( (a1^2+a2^2) \)
with ev1234 ass show ?thesis by auto
next
assume ¬ even \( (a1^2+a2^2) \)
hence odd12: odd \( (a1^2+a2^2) \) by simp
with ev1234 have odd34: odd \( (a3^2+a4^2) \) by auto
show ?thesis
proof (cases)
assume ev1: even \( (a1^2) \)
with odd12 have odd2: odd \( (a2^2) \) by simp
show ?thesis
proof (cases)
assume even \( (a3^2) \)
moreover from \( as \) have \( p*x = sum4sq-int(a1,a3,a2,a4) \) by (auto simp add: sum4sq-int-def)
ultimately show ?thesis using odd2 odd34 ev1 by auto
next
assume ¬ even \( (a3^2) \)
moreover from \( as \) have \( p*x = sum4sq-int(a1,a4,a2,a3) \) by (auto simp add: sum4sq-int-def)
ultimately show ?thesis using odd34 odd2 ev1 by auto
qed
next
assume odd1: ¬ even \( (a1^2) \)
with odd12 have ev2: even \( (a2^2) \) by simp
show ?thesis
proof (cases)
assume even \( (a1^2) \)
moreover from \( as \) have \( sum4sq-int(a1,a4,a2,a3)=p*x \) by (auto simp add:
ultimately show \( \forall x: \text{odd3} \text{ odd1 even2 by force} \)

next

assume \( \neg \text{ even } (a3^2) \)

moreover from \( \text{ass} \) have \( \text{sum4sq-int}(a1,a3,a2,a4) = p \times x \) by (auto simp add: \( \text{sum4sq-int-def} \))

ultimately show \( \forall x: \text{odd3} \text{ odd1 even2 by force} \)

qed

then obtain \( b1 \ b2 \ b3 \ b4 \)

where \( b: p \times x = \text{sum4sq-int}(b1,b2,b3,b4) \) ∧ even \((b1+b2)\) ∧ even \((b3+b4)\) by auto

then obtain \( c1 \ c3 \) where \( c13: b1+b2 = 2 \times c1 \) ∧ \( b3+b4 = 2 \times c3 \)

using evenE[of \( b1+b2 \)] evenE[of \( b3+b4 \)] by meson

from \( b \) have even \((b1-b2)\) ∧ even \((b3-b4)\) by simp

then obtain \( c2 \ c4 \) where \( c24: b1-b2 = 2 \times c2 \) ∧ \( b3-b4 = 2 \times c4 \)

using evenE[of \( b1-b2 \)] evenE[of \( b3-b4 \)] by meson

from \( ev \) obtain \( y \) where \( y: x = 2 \times y \) using evenE by blast

hence \( 4 \times (p \times y) = 2 \times (p \times x) \) by (simp add: ac-simps)

also from \( b \) have \( \ldots = 2 \times b1^2 + 2 \times b2^2 + 2 \times b3^2 + 2 \times b4^2 \)

by (auto simp: \( \text{sum4sq-int-def} \))

also have \( \ldots = 4 \times (c1^2 + c2^2 + c3^2 + c4^2) \)

by (auto simp add: power2-eq-square algebra-simps)

also with \( c13 \ c24 \) have \( \ldots = 4 \times (c1^2 + c2^2 + c3^2 + c4^2) \)

by (auto simp add: power-mult-distrib)

finally have \( p \times y = \text{sum4sq-int}(c1,c2,c3,c4) \) by (auto simp add: \( \text{sum4sq-int-def} \))

moreover from \( y \) ass have \( 0 < y \land y < p \land (\text{nat } y) = 1 < (\text{nat } x) = 1 \) by arith

ultimately show \( \forall x: \text{thesis by blast} \)

next

assume \( xodd: \neg \text{ even } x \)

with \( \text{ass have } 3 \ c1 \ c2 \ c3 \ c4 \) 2 \times (a1-c1 \times x) \leq x \land 2 \times (a2-c2 \times x) \leq x \land 2 \times (a3-c3 \times x) \leq x \land 2 \times (a4-c4 \times x) \leq x \)

by (simp add: best-division-abs)

then obtain \( b1 \ c1 \ b2 \ c2 \ b3 \ c3 \ b4 \ c4 \) where \( bc\text{-def: } b1 = a1-c1 \times x \land b2 = a2-c2 \times x \land b3 = a3-c3 \times x \land b4 = a4-c4 \times x \)

and \( 2 \times |b1| \leq x \land 2 \times |b2| \leq x \land 2 \times |b3| \leq x \land 2 \times |b4| \leq x \)

by blast

moreover have \( 2 \times |b1| \neq x \land 2 \times |b2| \neq x \land 2 \times |b3| \neq x \land 2 \times |b4| \neq x \) using \( xodd \)

by fastforce

ultimately have \( bc\text{-abs: } 2 \times |b1| < x \land 2 \times |b2| < x \land 2 \times |b3| < x \land 2 \times |b4| < x \) by auto

let \( \text{?B} = b1^2 + b2^2 + b3^2 + b4^2 \)

let \( \text{?C} = c1^2 + c2^2 + c3^2 + c4^2 \)

have \( x \ \text{dvd } \text{?B} \)

proof

from \( bc\text{-def} \) ass have

\( \text{?B} = p \times x - 2 \times (a1+c1+a2+c2+a3+c3+a4+c4) \times x + ?C \times x^2 \)

unfolding \( \text{sum4sq-int-def} \) by (auto simp add: power2-eq-square algebra-simps)

thus \( \text{?B} = x \times (p - 2 \times (a1+c1+a2+c2+a3+c3+a4+c4) + ?C \times x) \)

by (auto simp add: ac-simps power2-eq-square distrib-left right-diff-distrib)

qed
then obtain $y$ where $y$: $\exists B = x * y$ by (auto simp add: dvd-def)
let $\forall A = a1*b1 + a2*b2 + a3*b3 + a4*b4$
let $\forall A2 = a1*b2 - a2*b1 - a3*b4 + a4*b3$
let $\forall A3 = a1*b3 + a2*b4 - a3*b1 - a4*b2$
let $\forall A4 = a1*b4 - a2*b3 + a3*b2 - a4*b1$
let $\forall A = \text{sum4sq-int}(\forall A1,\forall A2,\forall A3,\forall A4)$

have $x$ dvd $\forall A1$ & $x$ dvd $\forall A2$ & $x$ dvd $\forall A3$ & $x$ dvd $\forall A4$
proof (safe)

from $bc\text{-def}$ have
$\forall A1 = (b1+c1*x)*b1 + (b2+c2*x)*b2 + (b3+c3*x)*b3 + (b4+c4*x)*b4$
  by simp
also with $y$ have ... = $x * (y + c1*b1 + c2*b2 + c3*b3 + c4*b4)$
  by (auto simp add: $bc\text{-def}$ right-diff-distrib ac-simps)
finally show $x$ dvd $\forall A1$ by auto

from $bc\text{-def}$ have
$\forall A2 = (b1+c1*x)*b2 - (b2+c2*x)*b1 - (b3+c3*x)*b4 + (b4+c4*x)*b3$
  by simp
also have ... = $x * (c1*b2 - c2*b1 - c3*b4 + c4*b3)$
  by (auto simp add: distrib-left right-diff-distrib ac-simps)
finally show $x$ dvd $\forall A2$ by auto

from $bc\text{-def}$ have
$\forall A3 = (b1+c1*x)*b3 + (b2+c2*x)*b4 - (b3+c3*x)*b1 - (b4+c4*x)*b2$
  by simp
also have ... = $x * (c1*b3 + c2*b4 - c3*b1 - c4*b2)$
  by (auto simp add: distrib-left right-diff-distrib ac-simps)
finally show $x$ dvd $\forall A3$ by auto

from $bc\text{-def}$ have
$\forall A4 = (b1+c1*x)*b4 - (b2+c2*x)*b3 + (b3+c3*x)*b2 - (b4+c4*x)*b1$
  by simp
also have ... = $x * (c1*b4 - c2*b3 + c3*b2 - c4*b1)$
  by (auto simp add: distrib-left right-diff-distrib ac-simps)
finally show $x$ dvd $\forall A4$ by auto

qed

then obtain $d1$ $d2$ $d3$ $d4$ where $d$:
$\exists A1=x*d1$ & $\exists A2=x*d2$ & $\exists A3=x*d3$ & $\exists A4=x*d4$
  by (auto simp add: dvd-def)

let $\exists D = \text{sum4sq-int}(d1,d2,d3,d4)$

from $d$ have $x^2*D = \exists A$
  by (auto simp only: $\text{sum4sq-int-def}$ power-mul-distrib distrib-left)
also have ... = $\text{sum4sq-int}(a1,a2,a3,a4)*\text{sum4sq-int}(b1,b2,b3,b4)$
  by (simp only: $\text{mult-sum4sq-int}$)
also with $y$ ass have ... = $(p*x)*(x*y)$
  by (auto simp add: $\text{sum4sq-int-def}$)
also have ... = $x^2*(p*y)$
  by (simp only: power2-eg-square ac-simps)
finally have $x^2*(\exists D-p*y) = 0$
  by (auto simp add: right-diff-distrib)

with $\exists$ have $p*y = \exists D$ by auto

moreover have $y-l-x$: $y < x$

proof -

have $4*b1^2 = (2*|b1|)^2$ & $4*b2^2 = (2*|b2|)^2$ &
  $4*b3^2 = (2*|b3|)^2$ & $4*b4^2 = (2*|b4|)^2$
  by simp

with $bc\text{-abs}$ have $4*b1^2 < x^2$ & $4*b2^2 < x^2$ &
  $4*b3^2 < x^2$ & $4*b4^2 < x^2$
  using power-strict-mono [of $2*|b|$ $2$ for $b$]
  by auto
1 Lagrange’s four-square theorem

hence $B < x^2$ by auto
with $y$ have $x \cdot (x - y) > 0$
  by (auto simp add: power2-eq-square right-diff-distrib)
moreover from $ass$ have $x > 0$ by simp
ultimately show $\neg\neg ?thesis$ using zero-less-mult-pos by fastforce
qed
moreover have $y > 0$
proof -
  have $b2pos$: $b_1^2 \geq 0 \land b_2^2 \geq 0 \land b_3^2 \geq 0 \land b_4^2 \geq 0$ by simp
hence $B = 0 \lor B > 0$ by arith
moreover
  { assume $B = 0$
    ultimately have $b1^2 = 0 \land b2^2 = 0 \land b3^2 = 0 \land b4^2 = 0$ by arith
    with $bc$-def have $x \ d v d \ a_1 \land x \ d v d \ a_2 \land x \ d v d \ a_3 \land x \ d v d \ a_4$
      by auto
    hence $x^2 \ d v d \ a_1^2 \land x^2 \ d v d \ a_2^2 \land x^2 \ d v d \ a_3^2 \land x^2 \ d v d \ a_4^2$ by simp
    hence $x \cdot x \ d v d \ x \cdot x$ by (simp only: sum4sq-int-def)
    with $ass$ have $nat \ x \ d v d \ nat \ p$
      by (simp add: nat-dvd-iff)
    moreover from $ass \ p r p$ have $x \geq 0 \land x \neq 1 \land x \neq p \land prime (nat \ p)$ by simp
    ultimately have $False$ unfolding prime-nat-iff by auto } 
moreover
  { assume $B > 0$
    with $y$ have $x \cdot y > 0$ by simp
    moreover from $ass$ have $x > 0$ by simp
    ultimately have $?thesis$ using zero-less-mult-pos by blast } 
ultimately show $?thesis$ by auto
qed
moreover with $y-l-x$ have $(nat \ y) - 1 < (nat \ x) - 1$ by arith
moreover from $y-l-x$ have $y < p$ by auto
ultimately show $?thesis$ by blast
qed
thus $\exists \ y. \ nat \ y - 1 < nat \ x - 1 \land \neg \neg ?Q \ y$ by blast
qed
with $Qt$ show $False$ by simp
qed
thus $is-sum4sq-int \ p$ by (auto simp add: is-sum4sq-int-def)
qed
private lemma prime-is-sum4sq: $prime \ p \implies is-sum4sq-nat \ p$
  using $zprime-is-sum4sq$ is-sum4sq-int-nat-eq by simp
theorem sum-of-four-squares: $is-sum4sq-nat \ n$
proof (induction $n$ rule: nat-less-induct)
case $(1 \ n)$
show ?case
proof (cases n>1)
case False
  hence n = 0 ∨ n = 1 by auto
  moreover have 0 = sum4sq-nat 0 0 0 1 = sum4sq-nat 1 0 0 0 unfolding sum4sq-nat-def by auto
  ultimately show ?thesis unfolding is-sum4sq-nat-def by blast
next
case True
  then obtain p m where dec: prime p ∧ n = p * m using prime-factor-nat[of n]
    by (auto elim: dvdE)
  moreover hence m<n using n-less-m-mult-n[of m p] prime-gt-Suc-0-nat[of p] True by linarith
  ultimately have is-sum4sq-nat m is-sum4sq-nat p using 1 prime-is-sum4sq by blast+
  thus ?thesis using dec is-mult-sum4sq-nat by blast
qed
qed

end

References


