

Sums of two and four squares

Roelof Oosterhuis
University of Groningen

August 21, 2007

Abstract

This document gives the formal proofs, verified by the proof assistant ‘Isabelle’¹, of the following results about the sums of two and four squares:

1. Any prime number $p \equiv 1 \pmod{4}$ 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 chapter II and III of André Weil, *Number theory: an approach through history; From Hammurapi to Legendre*. Boston (etc.): Birkhäuser, 1983.

We remind the reader that the results have been formalised before in the proof assistant HOL-light². A more complete study of the sum of two squares, including the first result, has been formalised in ‘Coq’³. Moreover it should be mentioned that the results can be found as numbers 20 and 19 on the list of ‘top 100 mathematical theorems’.⁴

This research is part of a M.Sc. thesis under supervision of Jaap Top and Wim H. Hesselink (RU Groningen).

More information: <http://www.roelofosterhuis.nl/MScthesis.pdf>

¹See <http://isabelle.in.tum.de/>

²See <http://www.cl.cam.ac.uk/~jrh13/hol-light/>

³See <http://coq.inria.fr/contribs/SumOfTwoSquare.html>

⁴See <http://www.cs.ru.nl/~freek/100/>

Contents

1	Sums of two squares	3
2	Lagrange's four-square theorem	8

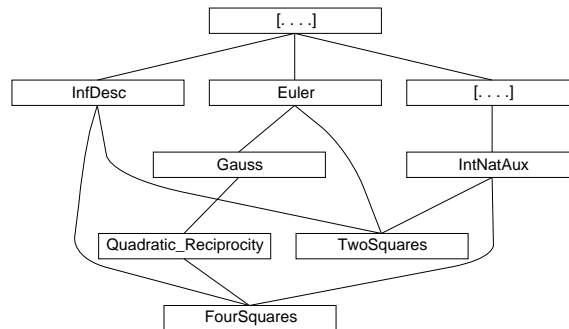


Figure 1: The dependence on existing files in the Isabelle library.

1 Sums of two squares

```

theory TwoSquares
  imports IntNatAux InfDesc
  ~~/src/HOL/NumberTheory/Euler
begin

```

Show that $\left(\frac{-1}{p}\right) = +1$ for primes $p \equiv 1 \pmod{4}$.

constdefs

```

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

```

```

is-sum2sq :: int ⇒ bool
is-sum2sq x == ∃ a b. sum2sq(a,b) = x

```

lemma *mult-sum2sq*: $sum2sq(a,b) * sum2sq(p,q) = sum2sq(a*p+b*q, a*q-b*p)$
by (*unfold sum2sq-def, simp add: nat-number ring-simps*)

lemma *is-mult-sum2sq*: $is-sum2sq x \implies is-sum2sq y \implies is-sum2sq (x*y)$
by (*unfold is-sum2sq-def, auto simp only: mult-sum2sq, blast*)

lemma *Legendre-1mod4*: $zprime (4*m+1) \implies (Legendre (-1) (4*m+1)) = 1$

proof –

```

let ?p = 4*m+1
let ?L = Legendre (-1) ?p
assume p: zprime ?p
have m ≥ 1
proof (rule ccontr)
  assume ¬ m ≥ 1 hence m ≤ 0 by auto
  hence ?p ≤ 1 by auto
  with p show False by (simp add: zprime-def)
qed
hence p2: ?p > 2 by simp
with p have [?L = (-1)^nat((?p - 1) div 2)] (mod ?p)
  by (simp only: Euler-Criterion)
hence [?L = (-1)^(2*nat m)] (mod ?p) by (auto simp add: nat-mult-distrib)
hence [1 = ?L] (mod ?p) by (auto simp add: power-mult power2-minus zcong-sym)
hence ?p dvd 1-?L by (simp add: zcong-def)
moreover have ?L=1 ∨ ?L=0 ∨ ?L=-1 by (simp add: Legendre-def)
ultimately have ?L = 1 ∨ ?p dvd 1 ∨ ?p dvd 2 by auto
moreover
{ assume ?p dvd 1 ∨ ?p dvd 2
  with p2 have False by (auto simp add: zdvd-not-zless) }
ultimately show ?thesis by auto

```

qed

Use this to prove that such primes can be written as the sum of two squares.

lemma *qf1-prime-exists*: $zprime (4*m+1) \implies \exists x y. x^2 + y^2 = 4*m+1$

proof –

```

let ?p = 4*m+1

```

```

assume p: zprime ?p
hence Legendre (-1) ?p = 1 by (rule Legendre-1mod4)
moreover
{ assume ¬ QuadRes ?p (-1)
  hence Legendre (-1) ?p ≠ 1 by (unfold Legendre-def, auto) }
ultimately have QuadRes ?p (-1) by auto
then obtain s1 where s1: [s1^2 = -1] (mod ?p) by (auto simp add: QuadRes-def)
from p have p0: ?p > 0 by (simp add: zprime-def)
hence ∃! s. 0 ≤ s ∧ s < ?p ∧ [s1 = s] (mod ?p)
  by (simp only: zcong-zless-unique)
then obtain s where s0p: 0 ≤ s ∧ s < ?p ∧ [s1 = s] (mod ?p)
  by auto
hence [s^2 = s1^2] (mod ?p) by (simp only: zcong-sym power2-eq-square zcong-zmult)
with s1 have s: [s^2 = -1] (mod ?p) by (auto dest: zcong-trans)
hence ?p dvd s^2 - (-1::int) by (unfold zcong-def, simp)
moreover have s^2 - (-1::int) = s^2 + 1 by arith
ultimately have ?p dvd s^2 + 1 by simp
then obtain t where t: s^2 + 1 = ?p*t by (auto simp add: dvd-def)
hence sum2sq(s,1) = ?p*t by (simp add: sum2sq-def)
hence qf1pt: is-sum2sq (?p*t) by (auto simp add: is-sum2sq-def)
have t-l-p: t < ?p
proof (rule ccontr)
  assume ¬ t < ?p hence t > ?p - 1 by simp
  with p0 have ?p*(?p - 1) < ?p*t by (simp only: zmult-zless-mono2)
  also with t have ... = s^2 + 1 by simp
  also have ... ≤ ?p*(?p - 1) - ?p + 2
  proof -
    from s0p have s ≤ ?p - 1 by (auto simp add: less-int-def)
    with s0p have s^2 ≤ (?p - 1)^2 by (simp only: power-mono)
    also have ... = ?p*(?p - 1) - 1*(?p - 1)
      by (simp only: power2-eq-square zdiff-zmult-distrib)
    finally show ?thesis by auto
  qed
  finally have ?p < 2 by arith
  with p show False by (unfold zprime-def, auto)
qed
have tpos: t ≥ 1
proof (rule ccontr)
  assume ¬ t ≥ 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-zless-mono2)
    with t have s^2 + 1 < 0 by auto }
  moreover have s^2 ≥ 0 by (simp only: zero-le-power2)
  ultimately show False by (auto simp add: less-int-def)
qed
moreover
{ assume t = 1
  with qf1pt have is-sum2sq ?p by auto
  hence ?thesis by (unfold is-sum2sq-def sum2sq-def, auto) }

```

```

moreover
{ assume t1:  $t > 1$ 
  then obtain tn where tn:  $tn = (nat\ t) - 1$  and tn0:  $tn > 0$  by auto
  have is-sum2sq (?p*(1+ int 0)) (is ?Q 0)
  — So, Q n = there exist x, y such that  $x^2 + y^2 = (p * (1 + int(n)))$ 
  proof (rule ccontr)
    assume nQ1:  $\neg ?Q\ 0$ 
    have  $(1+int\ tn) < ?p \implies \neg ?Q\ tn$ 
    proof (induct tn rule: infinite-descent)
      case 0
      from nQ1 show  $1+int\ 0 < ?p \implies \neg ?Q\ 0$  by simp
    next
    case (smaller n)
    hence n0:  $n > 0$  and IH:  $1+int\ n < ?p \wedge ?Q\ n$  by auto
    then obtain x y where xy:  $x^2 + y^2 = ?p*(1+int\ n)$ 
    by (unfold is-sum2sq-def sum2sq-def, auto)
    let ?n1 =  $(1+int\ n)$ 
    from n0 have n1pos: ?n1 > 0 by simp
    then obtain r v where rv:  $v = x - r*?n1 \wedge 2*|v| \leq ?n1$ 
    by (frule-tac x=?n1 in best-division-abs, auto)
    from n1pos obtain s w where sw:  $w = y - s*?n1 \wedge 2*|w| \leq ?n1$ 
    by (frule-tac x=?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 ... =
         $x^2 + y^2 - 2*x*(r*?n1) - 2*y*(s*?n1) + (r*?n1)^2 + (s*?n1)^2$ 
        by (simp only: zdiff-power2)
      also with xy have ... =
         $?n1*?p - ?n1*(2*x*r) - ?n1*(2*y*s) + ?n1^2*r^2 + ?n1^2*s^2$ 
        by (simp only: mult-ac 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 zadd-zmult-distrib2 mult-ac
        zdiff-zmult-distrib zdiff-zmult-distrib2 power2-eq-square)
    qed
    then obtain m1 where m1: ?C = ?n1*m1 by (auto simp add: dvd-def)
    have mn: m1 < ?n1
    proof (rule ccontr)
      assume  $\neg m1 < ?n1$  hence ?n1 - m1  $\leq 0$  by simp
      hence  $4*?n1 - 4*m1 \leq 0$  by simp
      with n1pos have  $2*?n1 - 4*m1 < 0$  by simp
      moreover from n1pos have ?n1 > 0 by simp
    ultimately have ?n1*( $2*?n1 - 4*m1$ ) < ?n1*0 by (simp only: zmult-zless-mono2)
    hence contr: ?n1*( $2*?n1 - 4*m1$ ) < 0 by simp
    have hlp:  $2*|v| \geq 0 \wedge 2*|w| \geq 0$  by simp
    from m1 have  $4*?n1*m1 = 4*v^2 + 4*w^2$  by arith
    also have ... =  $(2*|v|)^2 + (2*|w|)^2$ 
    by (auto simp add: power2-abs power-mult-distrib)
    also from rv hlp have ...  $\leq ?n1^2 + (2*|w|)^2$ 
    by (auto simp add: power-mono)
    also from sw hlp have ...  $\leq ?n1^2 + ?n1^2$ 

```

by (auto simp add: power-mono)
 finally have $?n1 * m1^4 \leq ?n1 * ?n1^2$
 by (simp add: power2-eq-square mult-ac)
 hence $?n1 * (2 * ?n1 - 4 * m1) \geq 0$
 by (auto simp add: zdiff-zmult-distrib2 mult-ac)
 hence $?n1 * (2 * ?n1 - 4 * m1) > -1$ by auto
 with contr show False by auto
 qed
 have $(r*v + s*w + m1)^2 + (r*w - s*v)^2 = ?p*m1$
 proof -
 from $m1$ xy have $(?p * ?n1) * ?C = (x^2 + y^2) * (v^2 + w^2)$ by simp
 also have $\dots = (x*v + y*w)^2 + (x*w - y*v)^2$
 by (simp add: nat-number ring-simps)
 also with rv sw have $\dots =$
 $((r * ?n1 + v) * v + (s * ?n1 + w) * w)^2 + ((r * ?n1 + v) * w - (s * ?n1 + w) * v)^2$
 by simp
 also have $\dots =$
 $(?n1 * (r*v) + ?n1 * (s*w) + (v^2 + w^2))^2 + (?n1 * (r*w) - ?n1 * (s*v))^2$
 by (simp add: nat-number ring-simps)
 also from $m1$ have $\dots =$
 $(?n1 * (r*v) + ?n1 * (s*w) + ?n1 * m1)^2 + (?n1 * (r*w) - ?n1 * (s*v))^2$
 by simp
 finally have
 $(?p * ?n1) * ?C = ?n1^2 * (r*v + s*w + m1)^2 + ?n1^2 * (r*w - s*v)^2$
 by (simp add: nat-number ring-simps)
 with $m1$ have
 $?n1^2 * (?p * m1) = ?n1^2 * ((r*v + s*w + m1)^2 + (r*w - s*v)^2)$
 by (simp only: mult-ac power2-eq-square, simp add: zadd-zmult-distrib2)
 hence $?n1^2 * (?p * m1 - (r*v + s*w + m1)^2 - (r*w - s*v)^2) = 0$
 by (auto simp add: zadd-zmult-distrib2 zdiff-zmult-distrib2)
 moreover from $n1pos$ have $?n1^2 \neq 0$ by (simp add: power2-eq-square)
 ultimately show ?thesis by simp
 qed
 hence $qf1pm1$: is-sum2sq (?p * m1) by (unfold is-sum2sq-def sum2sq-def, auto)
 have $m1pos$: $m1 > 0$
 proof -
 { assume $v^2 + w^2 = 0$
 moreover
 { assume $v \neq 0$
 hence $v^2 > 0$ by (simp add: zero-less-power2)
 moreover have $w^2 \geq 0$ by (rule zero-le-power2)
 ultimately have $v^2 + w^2 > 0$ by arith }
 moreover
 { assume $w \neq 0$
 hence $w^2 > 0$ by (simp add: zero-less-power2)
 moreover have $v^2 \geq 0$ by (rule zero-le-power2)
 ultimately have $v^2 + w^2 > 0$ by arith }
 ultimately have $v = 0 \wedge w = 0$ by auto
 with rv sw have $?n1 \text{ dvd } x \wedge ?n1 \text{ dvd } y$ by (unfold dvd-def, auto)
 hence $?n1^2 \text{ dvd } x^2 \wedge ?n1^2 \text{ dvd } y^2$ by (simp add: zpower-zdvd-mono)
 hence $?n1^2 \text{ dvd } x^2 + y^2$ by (simp only: zdvd-zadd)
 with xy have $?n1 * ?n1 \text{ dvd } ?n1 * ?p$

```

    by (simp only: power2-eq-square mult-ac)
    moreover from n1pos have ?n1 ≠ 0 by simp
    ultimately have ?n1 dvd ?p by (rule zdvd-mult-cancel)
    with n1pos have ?n1 ≥ 0 ∧ ?n1 dvd ?p by simp
    with p have ?n1 = 1 ∨ ?n1 = ?p by (unfold zprime-def, blast)
    with IH have ?Q 0 by auto
    with nQ1 have False by simp }
  moreover
  { assume v^2 + 1*w^2 ≠ 0
    moreover have v^2 + w^2 ≥ 0
    proof -
      have v^2 ≥ 0 ∧ w^2 ≥ 0 by (auto simp only: zero-le-power2)
      thus ?thesis by arith
    qed
    ultimately have vwpos: v^2 + w^2 > 0 by simp
    with m1 have m1 ≠ 0 by auto
    moreover have m1 ≥ 0
    proof (rule ccontr)
      assume ¬ m1 ≥ 0 hence m1 < 0 by simp
      with n1pos have ?n1*m1 < ?n1*0 by (simp only: zmult-zless-mono2)
      with m1 vwpos show False by simp
    qed
    ultimately have ?thesis m1 > 0 by auto }
  ultimately show ?thesis by auto
qed
hence 1 + int((nat m1) - 1) = m1 by arith
with qf1pm1 have Qm1: ?Q ((nat m1) - 1) by auto
then obtain m where tmp: m = (nat m1) - 1 ∧ ?Q m by auto
moreover have m < n
proof -
  from tmp mn m1pos have int m < int n by arith
  thus ?thesis by arith
qed
moreover with IH have 1 + int m < ?p by auto
ultimately show ∃ m. m < n ∧ ¬ (1 + int m < ?p → ¬ ?Q m) by auto
qed
moreover from tn tpos t-l-p have 1 + int tn < ?p ∧ tn = nat t - 1
  by arith
ultimately have ¬ ?Q ((nat t) - 1) by simp
moreover from tpos have 1 + int ((nat t) - 1) = t by arith
ultimately have ¬ is-sum2sq (?p*t) by auto
with qf1pt show False by simp
qed
hence ?thesis by (unfold is-sum2sq-def sum2sq-def, auto) }
ultimately show ?thesis by (auto simp add: less-int-def)
qed
end

```

2 Lagrange's four-square theorem

theory *FourSquares*

imports *IntNatAux InfDesc*

~/src/HOL/NumberTheory/Quadratic-Reciprocity

begin

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

$$\{x^2 \bmod p \mid 0 \leq x \leq n\} \text{ and } \{-1 - y^2 \bmod p \mid 0 \leq y \leq n\}$$

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.

constdefs

sum4sq :: *int* × *int* × *int* × *int* ⇒ *int*

sum4sq == λ(*a, b, c, d*). $a^2 + b^2 + c^2 + d^2$

is-sum4sq :: *int* ⇒ *bool*

is-sum4sq *x* == ∃ *a b c d*. *sum4sq*(*a, b, c, d*) = *x*

lemma *mult-sum4sq*: *sum4sq*(*a, b, c, d*) * *sum4sq*(*p, q, r, s*) =

sum4sq($a*p + b*q + c*r + d*s$, $a*q - b*p - c*s + d*r$,
 $a*r + b*s - c*p - d*q$, $a*s - b*r + c*q - d*p$)

by (*unfold sum4sq-def*, *simp add: nat-number ring-simps*)

lemma *is-mult-sum4sq*: *is-sum4sq* *x* ⇒ *is-sum4sq* *y* ⇒ *is-sum4sq* (*x*y*)

by (*unfold is-sum4sq-def*, *auto simp only: mult-sum4sq, blast*)

lemma *mult-oddprime-is-sum4sq*: [*zprime* *p*; *p* ∈ *zOdd*] ⇒

∃ *t*. $0 < t \wedge t < p \wedge \text{is-sum4sq } (p*t)$

proof –

assume *p1*: *zprime* *p*

hence *p0*: $p > 1$ **by** (*simp add: zprime-def*)

assume *p2*: $p \in \text{zOdd}$

then obtain *n* **where** $n: p = 2*n + 1$ **by** (*auto simp add: zOdd-def*)

with *p1* **have** *n0*: $n > 0$ **by** (*auto simp add: zprime-def*)

let *?C* = {*y*. $0 \leq y \wedge y < p$ }

let *?D* = {*y*. $0 \leq y \wedge y \leq n$ }

let *?f* = %*x*. $x^2 \bmod p$

let *?g* = %*x*. $(-1 - x^2) \bmod p$

```

let ?A = ?f ' ?D
let ?B = ?g ' ?D
have finC: finite ?C by (rule bdd-int-set-l-finite)
have finD: finite ?D by (rule bdd-int-set-le-finite)
from p0 have AsubC: ?A ⊆ ?C and BsubC: ?B ⊆ ?C
  by (auto simp add: pos-mod-conj)
with finC have finA: finite ?A and finB: finite ?B
  by (auto simp add: finite-subset)
from AsubC BsubC have AunBsubC: ?A ∪ ?B ⊆ ?C by (rule Un-least)
from p0 have cardC: card ?C = nat p by (simp only: card-bdd-int-set-l)
from n0 have cardD: card ?D = 1 + nat n by (simp only: card-bdd-int-set-le)
have cardA: card ?A = card ?D
proof -
  have inj-on ?f ?D
  proof (unfold inj-on-def, auto)
    fix x fix y
    assume x0: 0 ≤ x and xn: x ≤ n and y0: 0 ≤ y and yn: y ≤ n
      and xyp: x2 mod p = y2 mod p
    with p0 have [x2 = y2] (mod p) by (simp only: zcong-zmod-eq)
    hence p dvd x2 - y2 by (simp only: zcong-def)
    hence p dvd (x+y)*(x-y) by (simp only: zspecial-product)
    with p1 have p dvd x+y ∨ p dvd x-y by (simp only: zprime-zdvd-zmult-general)
  moreover
  { assume p dvd x+y
    moreover from xn yn n have x+y < p by auto
    ultimately have ¬ x+y > 0 by (auto simp add: zdvd-not-zless)
    with x0 y0 have x = y by auto } — both are zero
  moreover
  { assume ass: p dvd x-y
    have x = y
    proof (rule ccontr, case-tac x-y ≥ 0, auto)
      assume x-y ≥ 0 and x ≠ y hence x-y > 0 by auto
      with ass have ¬ x-y < p by (auto simp add: zdvd-not-zless)
      with xn y0 n p0 show False by auto
    next
      assume ¬ 0 ≤ x-y hence y-x > 0 by auto
      moreover from x0 yn n p0 have y-x < p by auto
      ultimately have ¬ p dvd y-x by (auto simp add: zdvd-not-zless)
      moreover from ass have p dvd -(x-y) by (simp only: zdvd-zminus-iff)
      ultimately show False by auto
    qed }
    ultimately show x=y by auto
  qed
with finD show ?thesis by (simp only: inj-on-iff-eq-card)
qed
have cardB: card ?B = card ?D
proof -
  have inj-on ?g ?D
  proof (unfold inj-on-def, auto)
    fix x fix y
    assume x0: 0 ≤ x and xn: x ≤ n and y0: 0 ≤ y and yn: y ≤ n
      and xyp: (-1-x2) mod p = (-1-y2) mod p

```

```

with p0 have [-1-y^2 = -1-x^2] (mod p) by (simp only: zcong-zmod-eq)
hence p dvd (-1-y^2) - (-1-x^2) by (simp only: zcong-def)
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 only: zspecial-product)
with p1 have p dvd x+y ∨ p dvd x-y by (simp only: zprime-zdvd-zmult-general)
moreover
{ assume p dvd x+y
  moreover from xn yn n have x+y < p by auto
  ultimately have ¬ x+y > 0 by (auto simp add: zdvd-not-zless)
  with x0 y0 have x = y by auto } — both are zero
moreover
{ assume ass: p dvd x-y
  have x = y
  proof (rule ccontr, case-tac x-y ≥ 0, auto)
    assume x-y ≥ 0 and x ≠ y hence x-y > 0 by auto
    with ass have ¬ x-y < p by (auto simp add: zdvd-not-zless)
    with xn y0 n p0 show False by auto
  next
    assume ¬ 0 ≤ x-y hence y-x > 0 by auto
    moreover from x0 yn n p0 have y-x < p by auto
    ultimately have ¬ p dvd y-x by (auto simp add: zdvd-not-zless)
    moreover from ass have p dvd -(x-y) by (simp only: zdvd-zminus-iff)
    ultimately show False by auto
  qed }
ultimately show x=y by auto
qed
with finD show ?thesis by (simp only: inj-on-iff-eq-card)
qed
have ?A ∩ ?B ≠ {}
proof (rule ccontr, auto)
  assume ABdisj: ?A ∩ ?B = {}
  from cardA cardB cardD have 2 + 2*(nat n) = card ?A + card ?B by auto
  also with finA finB ABdisj have ... = card (?A ∪ ?B)
  by (simp only: card-Un-disjoint)
  also with finC AunBsubC have ... ≤ card ?C by (simp only: card-mono)
  also with cardC have ... = nat p by simp
  finally have 2 + 2*(nat n) ≤ nat p by simp
  with n show False by arith
qed
then obtain z where z ∈ ?A ∧ z ∈ ?B by auto
then obtain x y where xy: x ∈ ?D ∧ y ∈ ?D ∧ z = x^2 mod p ∧
  z = (-1-y^2) mod p by auto
with p0 have [x^2=-1-y^2](mod p) by (simp add: zcong-zmod-eq)
hence p dvd x^2-(-1-y^2) by (simp only: zcong-def)
moreover have x^2-(-1-y^2)=x^2+y^2+1 by arith
ultimately have p dvd sum4sq(x,y,1,0) by (auto simp add: sum4sq-def)
then obtain t where t: sum4sq(x,y,1,0) = p*t by (auto simp add: dvd-def)
hence is-sum4sq (p*t) by (unfold is-sum4sq-def, auto)
moreover have t > 0 ∧ t < p
proof
  have x^2 ≥ 0 ∧ y^2 ≥ 0 by (simp add: zero-le-power2)

```

hence $x^2+y^2+1 > 0$ by *arith*
 with t have $p*t > 0$ by (*unfold sum4sq-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 $\neg t < 0 \wedge t \neq 0$ by *auto*
 thus $t > 0$ by *simp*
 from xy have $x^2 \leq n^2 \wedge y^2 \leq n^2$ 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-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: zadd-zmult-distrib2*)
 hence $p*t > 2*n*n + n + 2*n + 1$ by (*simp only: zadd-zmult-distrib zmult-1*)
 with $n0$ have $p*t > 2*n^2 + 1$ by (*simp add: power2-eq-square*) }
 ultimately have $\neg t > n+1$ by *auto*
 with $n0$ n show $t < p$ by *auto*
 qed
 ultimately show *?thesis* by *blast*
 qed

lemma *zprime-is-sum4sq*: $zprime\ p \implies is-sum4sq\ p$

proof (*cases*)

assume $p2$: $p=2$

hence $p = sum4sq(1,1,0,0)$ by (*auto simp add: sum4sq-def*)

thus *?thesis* by (*auto simp add: is-sum4sq-def*)

next

assume $\neg p = 2$ and *prp*: $zprime\ p$

hence $2 < p$ by (*simp add: zprime-def*)

with *prp* have $p \in zOdd$ by (*simp only: zprime-zOdd-eq-grt-2*)

with *prp* have $\exists t. 0 < t \wedge t < p \wedge is-sum4sq\ (p*t)$

by (*rule mult-oddprime-is-sum4sq*)

then obtain $a\ b\ c\ d\ t$ where *pt-sol*: $0 < t \wedge t < p \wedge sum4sq(a,b,c,d)=p*t$

by (*unfold is-sum4sq-def, blast*)

hence *Qt*: $0 < t \wedge t < p \wedge (\exists a1\ a2\ a3\ a4. sum4sq(a1,a2,a3,a4)=p*t)$

(is *?Q t*) by *blast*

have *?Q 1*

proof (*rule ccontr*)

assume *nQ1*: $\neg ?Q\ 1$

have $\neg ?Q\ t$

proof (*rule-tac x=t and V= $\lambda x. (nat\ x) - 1$ in val-infinite-descent, clarify*)

fix $x\ a\ b\ c\ d$

assume $nat\ x - 1 = 0$ and $x > 0$ and s : $sum4sq(a,b,c,d)=p*x$ 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 $\neg \neg ?Q\ x$

then obtain $a1\ a2\ a3\ a4$ **where** $ass: 1 < x \wedge x < p \wedge sum4sq(a1, a2, a3, a4) = p * x$
by *auto*
have $\exists y. nat\ y - 1 < nat\ x - 1 \wedge ?Q\ y$
proof (*cases*)
assume $evx: x \in zEven$
hence $x * p \in zEven$ **by** (*rule even-times-either*)
with ass **have** $ev1234: a1^2 + a2^2 + a3^2 + a4^2 \in zEven$
by (*auto simp add: sum4sq-def mult-ac*)
have $\exists b1\ b2\ b3\ b4. sum4sq(b1, b2, b3, b4) = p * x \wedge$
 $b1 + b2 \in zEven \wedge b3 + b4 \in zEven$
proof (*cases*)
assume $ev12: a1^2 + a2^2 \in zEven$
moreover **have** $2 * a1 * a2 \in zEven$ **by** (*auto simp add: zEven-def*)
ultimately **have** $a1^2 + a2^2 + 2 * a1 * a2 \in zEven$ **by** (*rule even-plus-even*)
hence $(a1 + a2)^2 \in zEven$ **by** (*auto simp add: zadd-power2 add-ac*)
hence $tmp: a1 + a2 \in zEven$ **by** (*auto simp add: power-preserves-even*)
from $ev12\ ev1234$ **have** $a1^2 + a2^2 + a3^2 + a4^2 - (a1^2 + a2^2) \in zEven$
by (*simp only: even-minus-even*)
hence $a3^2 + a4^2 \in zEven$ **by** *auto*
moreover **have** $2 * a3 * a4 \in zEven$ **by** (*auto simp add: zEven-def*)
ultimately **have** $a3^2 + a4^2 + 2 * a3 * a4 \in zEven$ **by** (*rule even-plus-even*)
hence $(a3 + a4)^2 \in zEven$ **by** (*auto simp add: zadd-power2 add-ac*)
hence $a3 + a4 \in zEven$ **by** (*auto simp add: power-preserves-even*)
with $tmp\ ass$ **show** *?thesis* **by** *blast*
next
assume $\neg a1^2 + a2^2 \in zEven$
hence $odd12: a1^2 + a2^2 \in zOdd$ **by** (*simp add: odd-iff-not-even*)
with $ev1234$ **have** $a1^2 + a2^2 + a3^2 + a4^2 - (a1^2 + a2^2) \in zOdd$
by (*simp only: even-minus-odd*)
hence $odd34: a3^2 + a4^2 \in zOdd$ **by** *auto*
show *?thesis*
proof (*cases*)
assume $ev1: a1^2 \in zEven$
with $odd12$ **have** $odd2: a2^2 \in zOdd$ **by** (*rule even-plus-odd-prop2*)
show *?thesis*
proof (*cases*)
assume $ev3: a3^2 \in zEven$
with $odd34$ **have** $a4^2 \in zOdd$ **by** (*rule even-plus-odd-prop2*)
with $odd2$ **have** $a2 \in zOdd \wedge a4 \in zOdd$
by (*auto simp add: power-preserves-odd*)
hence $tmp: a2 + a4 \in zEven$ **by** (*simp only: odd-plus-odd*)
from $ev3\ ev1$ **have** $a1 \in zEven \wedge a3 \in zEven$
by (*auto simp add: power-preserves-even*)
hence $tmp2: a1 + a3 \in zEven$ **by** (*simp only: even-plus-even*)
from ass **have** $sum4sq(a1, a3, a2, a4) = p * x$
by (*auto simp add: sum4sq-def*)
with $tmp\ tmp2$ **show** *?thesis* **by** *blast*
next
assume $\neg a3^2 \in zEven$
hence $odd3: a3^2 \in zOdd$ **by** (*simp add: odd-iff-not-even*)
with $odd34$ **have** $a4^2 \in zEven$ **by** (*rule even-plus-odd-prop1*)
with $ev1$ **have** $a1 \in zEven \wedge a4 \in zEven$

```

    by (auto simp add: power-preserves-even)
  hence tmp:  $a1+a4 \in zEven$  by (simp only: even-plus-even)
  from odd2 odd3 have  $a2 \in zOdd \wedge a3 \in zOdd$ 
    by (auto simp add: power-preserves-odd)
  hence tmp2:  $a2+a3 \in zEven$  by (simp only: odd-plus-odd)
  from ass have  $sum4sq(a1,a4,a2,a3)=p*x$ 
    by (auto simp add: sum4sq-def)
  with tmp tmp2 show ?thesis by blast
qed
next
assume  $\neg a1^2 \in zEven$ 
hence odd1:  $a1^2 \in zOdd$  by (simp add: odd-iff-not-even)
with odd12 have ev2:  $a2^2 \in zEven$  by (rule even-plus-odd-prop1)
show ?thesis
proof (cases)
  assume ev3:  $a3^2 \in zEven$ 
  with odd34 have  $a4^2 \in zOdd$  by (rule even-plus-odd-prop2)
  with odd1 have  $a1 \in zOdd \wedge a4 \in zOdd$ 
    by (auto simp add: power-preserves-odd)
  hence tmp:  $a1+a4 \in zEven$  by (simp only: odd-plus-odd)
  from ev3 ev2 have  $a2 \in zEven \wedge a3 \in zEven$ 
    by (auto simp add: power-preserves-even)
  hence tmp2:  $a2+a3 \in zEven$  by (simp only: even-plus-even)
  from ass have  $sum4sq(a1,a4,a2,a3)=p*x$ 
    by (auto simp add: sum4sq-def)
  with tmp tmp2 show ?thesis by blast
next
  assume  $\neg a3^2 \in zEven$ 
  hence odd3:  $a3^2 \in zOdd$  by (simp add: odd-iff-not-even)
  with odd34 have  $a4^2 \in zEven$  by (rule even-plus-odd-prop1)
  with ev2 have  $a2 \in zEven \wedge a4 \in zEven$ 
    by (auto simp add: power-preserves-even)
  hence tmp:  $a2+a4 \in zEven$  by (simp only: even-plus-even)
  from odd1 odd3 have  $a1 \in zOdd \wedge a3 \in zOdd$ 
    by (auto simp add: power-preserves-odd)
  hence tmp2:  $a1+a3 \in zEven$  by (simp only: odd-plus-odd)
  from ass have  $sum4sq(a1,a3,a2,a4)=p*x$ 
    by (auto simp add: sum4sq-def)
  with tmp tmp2 show ?thesis by blast
qed
qed
then obtain  $b1\ b2\ b3\ b4$  where  $b: sum4sq(b1,b2,b3,b4)=p*x \wedge$ 
 $b1+b2 \in zEven \wedge b3+b4 \in zEven$  by auto
then obtain  $c1\ c3$  where  $c13: b1+b2 = 2*c1 \wedge b3+b4 = 2*c3$ 
  by (auto simp add: zEven-def)
have  $2*b2 \in zEven \wedge 2*b4 \in zEven$  by (auto simp add: zEven-def)
with b have  $b1+b2 - 2*b2 \in zEven \wedge b3+b4 - 2*b4 \in zEven$ 
  by (auto simp only: even-minus-even)
moreover have  $b1+b2 - 2*b2 = b1-b2 \wedge b3+b4 - 2*b4 = b3-b4$  by auto
ultimately have  $b1-b2 \in zEven \wedge b3-b4 \in zEven$  by simp
then obtain  $c2\ c4$  where  $c24: b1-b2 = 2*c2 \wedge b3-b4 = 2*c4$ 

```

by (auto simp add: zEven-def)
 from evx obtain y where y: $x = 2*y$ by (auto simp add: zEven-def)
 hence $4*(p*y) = 2*(p*x)$ by (simp add: mult-ac)
 also from b have $\dots = 2*b1^2 + 2*b2^2 + 2*b3^2 + 2*b4^2$
 by (auto simp only: sum4sq-def)
 also have $\dots = (b1 + b2)^2 + (b1 - b2)^2 + (b3 + b4)^2 + (b3 - b4)^2$
 by (auto simp add: zadd-power2 zdiff-power2)
 also with c13 c24 have $\dots = 4*(c1^2 + c2^2 + c3^2 + c4^2)$
 by (auto simp add: power-mult-distrib)
 finally have $sum4sq(c1,c2,c3,c4) = p*y$ by (auto simp add: sum4sq-def)
 moreover from y ass have $0 < y \wedge y < p \wedge (nat\ y) - 1 < (nat\ x) - 1$ by arith
 ultimately show ?thesis by blast
 next
 assume $\neg x \in zEven$
 hence xodd: $x \in zOdd$ by (simp add: odd-iff-not-even)
 with ass have $\exists c1\ c2\ c3\ c4. 2*|a1 - c1*x| < x \wedge 2*|a2 - c2*x| < x$
 $\wedge 2*|a3 - c3*x| < x \wedge 2*|a4 - c4*x| < x$
 by (simp add: best-odd-division-abs)
 then obtain b1 c1 b2 c2 b3 c3 b4 c4 where
 bc-def: $b1 = a1 - c1*x \wedge b2 = a2 - c2*x \wedge b3 = a3 - c3*x \wedge b4 = a4 - c4*x$
 and bc-abs: $2*|b1| < x \wedge 2*|b2| < x \wedge 2*|b3| < x \wedge 2*|b4| < x$
 by blast
 let ?B = $b1^2 + b2^2 + b3^2 + b4^2$
 let ?C = $c1^2 + c2^2 + c3^2 + c4^2$
 have x dvd ?B
 proof
 from bc-def ass have
 ?B = $p*x - 2*(a1*c1 + a2*c2 + a3*c3 + a4*c4)*x + ?C*x^2$
 by (auto simp add: zdiff-power2 sum4sq-def
 zadd-zmult-distrib power-mult-distrib)
 thus ?B = $x*(p - 2*(a1*c1 + a2*c2 + a3*c3 + a4*c4) + ?C*x)$
 by (auto simp add: mult-ac power2-eq-square
 zadd-zmult-distrib2 zdiff-zmult-distrib2)
 qed
 then obtain y where y: $?B = x * y$ by (auto simp add: dvd-def)
 let ?A1 = $a1*b1 + a2*b2 + a3*b3 + a4*b4$
 let ?A2 = $a1*b2 - a2*b1 - a3*b4 + a4*b3$
 let ?A3 = $a1*b3 + a2*b4 - a3*b1 - a4*b2$
 let ?A4 = $a1*b4 - a2*b3 + a3*b2 - a4*b1$
 let ?A = $sum4sq(?A1, ?A2, ?A3, ?A4)$
 have x dvd ?A1 \wedge x dvd ?A2 \wedge x dvd ?A3 \wedge x dvd ?A4
 proof (safe)
 from bc-def have
 ?A1 = $(b1 + c1*x)*b1 + (b2 + c2*x)*b2 + (b3 + c3*x)*b3 + (b4 + c4*x)*b4$
 by simp
 also with y have $\dots = x*(y + c1*b1 + c2*b2 + c3*b3 + c4*b4)$
 by (auto simp add: zadd-zmult-distrib2 power2-eq-square mult-ac)
 finally show x dvd ?A1 by auto
 from bc-def have
 ?A2 = $(b1 + c1*x)*b2 - (b2 + c2*x)*b1 - (b3 + c3*x)*b4 + (b4 + c4*x)*b3$
 by simp
 also have $\dots = x*(c1*b2 - c2*b1 - c3*b4 + c4*b3)$

by (auto simp add: zadd-zmult-distrib2 zdiff-zmult-distrib2 mult-ac)
 finally show $x \text{ dvd } ?A2$ by auto
 from bc-def have
 $?A3 = (b1+c1*x)*b3 + (b2+c2*x)*b4 - (b3+c3*x)*b1 - (b4+c4*x)*b2$
 by simp
 also have $\dots = x*(c1*b3 + c2*b4 - c3*b1 - c4*b2)$
 by (auto simp add: zadd-zmult-distrib2 zdiff-zmult-distrib2 mult-ac)
 finally show $x \text{ dvd } ?A3$ by auto
 from bc-def have
 $?A4 = (b1+c1*x)*b4 - (b2+c2*x)*b3 + (b3+c3*x)*b2 - (b4+c4*x)*b1$
 by simp
 also have $\dots = x*(c1*b4 - c2*b3 + c3*b2 - c4*b1)$
 by (auto simp add: zadd-zmult-distrib2 zdiff-zmult-distrib2 mult-ac)
 finally show $x \text{ dvd } ?A4$ by auto
 qed
 then obtain $d1 \ d2 \ d3 \ d4$ where d :
 $?A1=x*d1 \wedge ?A2=x*d2 \wedge ?A3=x*d3 \wedge ?A4=x*d4$
 by (auto simp add: dvd-def)
 let $?D = \text{sum4sq}(d1, d2, d3, d4)$
 from d have $x^2 * ?D = ?A$
 by (auto simp only: sum4sq-def power-mult-distrib zadd-zmult-distrib2)
 also have $\dots = \text{sum4sq}(a1, a2, a3, a4) * \text{sum4sq}(b1, b2, b3, b4)$
 by (simp only: mult-sum4sq)
 also with y ass have $\dots = (p*x)*(x*y)$ by (auto simp add: sum4sq-def)
 also have $\dots = x^2*(p*y)$ by (simp only: power2-eq-square mult-ac)
 finally have $x^2*(?D - p*y) = 0$ by (auto simp add: zdiff-zmult-distrib2)
 with ass have $?D = p*y$ by auto
 moreover have $y \text{ l-} x: y < x$
 proof -
 have $4*b1^2 = (2*|b1|)^2 \wedge 4*b2^2 = (2*|b2|)^2 \wedge$
 $4*b3^2 = (2*|b3|)^2 \wedge 4*b4^2 = (2*|b4|)^2$
 by (auto simp add: power-mult-distrib abs-mult power2-abs)
 with bc-abs have $4*b1^2 < x^2 \wedge 4*b2^2 < x^2 \wedge 4*b3^2 < x^2 \wedge 4*b4^2 < x^2$
 by (auto simp add: power-strict-mono)
 hence $?B < x^2$ by auto
 with y have $x*(x-y) > 0$
 by (auto simp add: power2-eq-square zdiff-zmult-distrib2)
 moreover from ass have $x > 0$ by simp
 ultimately show $?thesis$ by (auto dest: pos-zmult-pos)
 qed
 moreover have $y > 0$
 proof -
 have $b2pos: b1^2 \geq 0 \wedge b2^2 \geq 0 \wedge b3^2 \geq 0 \wedge b4^2 \geq 0$
 by (auto simp add: zero-le-power2)
 hence $?B = 0 \vee ?B > 0$ by arith
 moreover
 { assume $?B = 0$
 moreover from $b2pos$ have
 $?B - b1^2 \geq 0 \wedge ?B - b2^2 \geq 0 \wedge ?B - b3^2 \geq 0 \wedge ?B - b4^2 \geq 0$ by arith
 ultimately have $b1^2 \leq 0 \wedge b2^2 \leq 0 \wedge b3^2 \leq 0 \wedge b4^2 \leq 0$ by auto
 with $b2pos$ have $b1^2 = 0 \wedge b2^2 = 0 \wedge b3^2 = 0 \wedge b4^2 = 0$ by arith
 hence $b1 = 0 \wedge b2 = 0 \wedge b3 = 0 \wedge b4 = 0$ by auto

```

with bc-def have x dvd a1 ∧ x dvd a2 ∧ x dvd a3 ∧ x dvd a4
  by (auto simp add: zdvd-triv-right)
hence x^2 dvd a1^2 ∧ x^2 dvd a2^2 ∧ x^2 dvd a3^2 ∧ x^2 dvd a4^2
  by (auto simp only: zpower-zdvd-mono)
hence x^2 dvd a1^2+a2^2+a3^2+a4^2 by (simp only: zdvd-zadd)
with ass have x^2 dvd p*x by (auto simp only: sum4sq-def)
hence x*x dvd x*p by (simp only: power2-eq-square mult-ac)
with ass have x dvd p by (auto dest: zdvd-mult-cancel)
moreover from ass prp have x ≥ 0 ∧ x ≠ 1 ∧ x ≠ p ∧ zprime p by simp
ultimately have False by (unfold zprime-def, auto) }
moreover
{ assume ?B > 0
  with y have x*y > 0 by simp
  moreover from ass have x > 0 by simp
  ultimately have ?thesis by (auto dest: pos-zmult-pos) }
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 ass have y < p by auto
ultimately show ?thesis by blast
qed
thus ∃ y. nat y - 1 < nat x - 1 ∧ ¬ ¬ ?Q y by blast
qed
with Qt show False by simp
qed
thus is-sum4sq p by (auto simp add: is-sum4sq-def)
qed

```

theorem four-squares: $(n::int) \geq 0 \implies \exists a b c d. a^2 + b^2 + c^2 + d^2 = n$

proof –

```

assume n ≥ 0
hence n = 0 ∨ n > 0 by auto
moreover
{ assume n = 0
  hence n = sum4sq(0,0,0,0) by (auto simp add: sum4sq-def)
  hence is-sum4sq n by (auto simp add: is-sum4sq-def) }
moreover
{ assume npos: n > 0
  hence nat n ≠ 0 by simp
  then obtain ps where ps: primel ps ∧ prod ps = nat n
    by (frule-tac a=nat n in factor-exists-general, auto)
  have primel ps ⟹ is-sum4sq (int (prod ps))
  proof (induct ps, auto)
    have sum4sq(1,0,0,0) = 1 by (auto simp add: sum4sq-def)
    thus is-sum4sq 1 by (auto simp add: is-sum4sq-def)
  next
    fix p ps
    let ?X = int (prod ps)
    assume primel ps ⟹ is-sum4sq ?X and primel (p#ps)
    hence prime p and x: is-sum4sq ?X by (auto simp add: primel-hd-tl)
    hence zprime (int p) by (simp only: prime-impl-zprime-int)
    hence is-sum4sq (int p) by (rule zprime-is-sum4sq)
  }

```

```
  with x have is-sum4sq((int p)*?X) by (simp add: is-mult-sum4sq)
  thus is-sum4sq (int (p*prod ps)) by (auto simp only: int-mult)
qed
with ps npos have is-sum4sq n by auto }
ultimately have is-sum4sq n by auto
thus ?thesis by (auto simp only: is-sum4sq-def sum4sq-def)
qed
end
```