Sums of two and four squares

Roelof Oosterhuis
University of Groningen

May 28, 2015

Abstract

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

1. Any prime number $p \equiv 1 \text{ mod } 4$ 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].
1 Sums of two squares

theory TwoSquares
imports ../Fermat3-4/IntNatAux ~/src/HOL/Old-Number-Theory/Euler
begin

Show that \((-\frac{1}{p}) = +1\) for primes \(p \equiv 1 \mod 4\).

definition
sum2sq :: \(\text{int} \times \text{int} \Rightarrow \text{int}\)
where
\(\text{sum2sq} = (\lambda (a, b). a^2 + b^2)\)

definition
is-sum2sq :: \(\text{int} \Rightarrow \text{bool}\)
where
\(\text{is-sum2sq } x \iff (\exists a b. \text{sum2sq} (a, b) = x)\)

lemma mult-sum2sq: \(\text{sum2sq} (a, b) \ast \text{sum2sq} (p, q) = \text{sum2sq} (a \ast p + b \ast q, a \ast q - b \ast p)\)
by (unfold \text{sum2sq-def}, simp add: \text{eval-nat-numeral field-simps})

lemma is-mult-sum2sq: \(\text{is-sum2sq } x = \Rightarrow \text{is-sum2sq } y = \Rightarrow \text{is-sum2sq} \ (x \ast y)\)
by (unfold \text{is-sum2sq-def}, auto simp only: \text{mult-sum2sq}, blast)

lemma Legendre-1mod4: \(\text{zprime } (4 \ast m + 1) \Rightarrow (\text{Legendre } -1) \ (4 \ast m + 1) = 1\)
proof –
let \(?p = 4 \ast m + 1\)
let \(?L = \text{Legendre } -1 \ ?p\)
assume \(p: \text{zprime } ?p\)
have \(m \geq 1\)
proof (rule \text{contr})
  assume \(\neg m \geq 1\) hence \(m \leq 0\) by \text{auto}
  hence \(?p \leq 1\) by \text{auto}
  with \(p\) show \(\text{False}\) by (simp add: \text{zprime-def})
qed

hence \(?p \geq 2\) by \text{simp}

with \(p\) have \([?L = (-1) \ ^\text{nat}((?p - 1) \ div 2)] \ (mod \ ?p)\)
by (simp only: \text{Euler-Criterion})

hence \([?L = (-1) \ ^\text{nat} m]\) \ (mod \ ?p) by (auto simp add: \text{nat-mult-distrib})

hence \([1 = ?L]\) \ (mod \ ?p) by (auto simp add: \text{power-mul \ power2-minus \ zcong-sym})

hence \(?p \ dvd 1 - ?L\) by (simp add: \text{zcong-def})

moreover have \(?L = 1 \lor ?L = 0 \lor ?L = -1\) by (simp add: \text{Legendre-def})
ultimately have \(\forall L = 1 \lor \ ?p \ dvd 1 \lor \ ?p \ dvd 2\) by \text{auto}

moreover
\{ assume \(?p \ dvd 1 \lor \ ?p \ dvd 2\)
  with \(?p2\) have \(\text{False}\) by (auto simp add: \text{zdvd-not-zless}) \}
ultimately show \(?\text{thesis}\) by \text{auto}
Use this to prove that such primes can be written as the sum of two squares.

**Lemma**: \( qf1 \)-prime-exists: \( \text{zprime} \ (4 \cdot m + 1) \implies \exists \ x, y. \ x^2 + y^2 = 4 \cdot m + 1 \)

**Proof**

- Let \( ?p = 4 \cdot m + 1 \)
- Assume \( p \): \( \text{zprime} \ ?p \)
- Hence Legendre \((-1) \ ?p = 1 \) by (rule Legendre-1mod4)
- Moreover
  - Assume \( \neg \quad \text{QuadRes} \ ?p (-1) \)
  - Hence Legendre \((-1) \ ?p \neq 1 \) by (unfold Legendre-def, auto)
- Ultimately have \( \text{QuadRes} \ ?p (-1) \) 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 \( \exists \ s. \ 0 \leq s \land s < \ ?p \land [s1 = s] \) (mod \ ?p)
  - by (simp only: zcong-zless-unique)
  - Then obtain \( s \) where \( s0p: \ 0 \leq s \land s < \ ?p \land [s1 = s] \) (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 \( ?p \ \text{dvd} \ s^2 - (-1::int) \) by (unfold zcong-def, simp)
- Moreover have \( s^2 - (-1::int) = s^2 + 1 \) by arith
- Ultimately have \( ?p \ \text{dvd} \ s^2 + 1 \) by simp
- Then obtain \( t \) where \( t: \ s^2 + 1 = ?p \cdot t \) by (auto simp add: dvd-def)
- Hence \( \text{sum2sq}(s,1) = ?p \cdot t \) by (simp add: sum2sq-def)
- Hence \( qf1pt: \ \text{is-sum2sq} \ (?p \cdot t) \) by (auto simp add: is-sum2sq-def)
- Have \( t-lp: \ t < ?p \)
- Proof (rule ccontr)
  - Assume \( \neg \quad t < ?p \) hence \( t > ?p - 1 \) by simp
  - With \( p0 \) have \( ?p \cdot ( ?p - 1 ) < ?p \cdot t \) by (simp only: zmult-zless-mono2)
  - Also with \( t \) have \( \ldots = s^2 + 1 \) by simp
  - Also have \( \ldots \leq ?p \cdot ( ?p - 1 ) - ?p + 2 \)
- Proof
  - From \( s0p \) have \( s \leq ?p - 1 \) by (auto simp add: less-le)
  - With \( s0p \) have \( s^2 \leq ( ?p - 1 )^2 \) by (simp only: power-mono)
  - Also have \( \ldots = ?p \cdot ( ?p - 1 ) - 1 \cdot ( ?p - 1 ) \)
    - by (simp only: power2-eq-square left-diff-distrib)
  - Finally show \( \neg \text{thesis} \) by auto
- Qed

Finally have \( ?p < 2 \) by arith

With \( p \) show \( False \) by (unfold zprime-def, auto)

Qed

**Have** \( tpos: \ t \geq 1 \)

**Proof** (rule ccontr)

- Assume \( \neg \quad t \geq 1 \) 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 \( ?p \cdot t < ?p \cdot 0 \) by (simp only: zmult-zless-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-lc)

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 contr)
      assume nQ1: ~ ?Q 0
      have (1+int tn) < ?p == > ~ ?Q tn
      proof (induct tn rule: infinite-descent0)
         case 0
         from nQ1 show 1+int 0 < ?p == ~ ?Q 0 by simp
      next
      case (smaller n)
      hence n0: n > 0 and IH: 1+int n < ?p ∧ ?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 ∧ 2*|v| ≤ ?n1
         by (frule-tac x=?n1 in best-division-abs, auto)
      from n1pos obtain s w where sw: w = y−s*?n1 ∧ 2*|w| ≤ ?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*?r^2 + ?n1*?s^2
         by (simp only: ac-simps power-mul-distrib)
      finally show ?C = ?n1*(?p − 2*x*r − 2*y*s + ?n1*(?r^2 + ?s^2))
         by (simp only: power-mul-distrib distrib-left ac-simps
         left-diff-distrib right-diff-distrib power2-eq-square)
      qed
      then obtain m1 where m1: ?C = ?n1*m1 by (auto simp add: dvd-def)
      have mn: m1 < ?n1
      proof (rule contr)
         assume = m1 < ?n1 hence ?n1−m1 ≤ 0 by simp
         hence 4*?n1 − 4*m1 ≤ 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-mon0)
         hence contr: ?n1*(2*?n1 − 4*m1) < 0 by simp
         have 6: 2*|v| ≥ 0 ∧ 2*|w| ≥ 0 by simp
from m1 have \(4 \cdot \?n1 \cdot m1 = 4 \cdot v^2 + 4 \cdot w^2\) by arith
also have \(\ldots = (2 \cdot |v|)^2 + (2 \cdot |w|)^2\)
  by (auto simp add: power2-abs power-mult-distrib)
also from rv hlp have \(\ldots \leq ?n1 \cdot 2 + (2 \cdot |w|)^2\)
  using power-mono [of \(2 \cdot |b|\) \(1 + \text{int n} 2\) for \(b\)]
  by auto
also from sw hlp have \(\ldots \leq ?n1 \cdot 2 + ?n1 \cdot 2\)
  using power-mono [of \(2 \cdot |b|\) \(1 + \text{int n} 2\) for \(b\)]
  by auto
finally have \(?n1 \cdot m1 \cdot 4 \leq \?n1 \cdot ?n1 \cdot 2\)
  by (simp add: power2-eq-square ac-simps)

have \((r \cdot v + s \cdot w + m1)^2 + (r \cdot w - s \cdot v)^2 = ?p \cdot m1\)
proof
  from m1 xy have \((?p \cdot ?n1) \cdot ?C = (x^2 + y^2) \cdot (v^2 + w^2)\) by simp
  also have \(\ldots = (x \cdot v + y \cdot w)^2 + (x \cdot w - y \cdot v)^2\)
    by (simp add: eval-nat-numeral field-simps)
  also with rv sw have \(\ldots =\)
    \((r \cdot ?n1 + v \cdot v) \cdot v + (s \cdot ?n1 + w \cdot w) \cdot w \cdot w + ((r \cdot ?n1 + v \cdot w) \cdot w - (s \cdot ?n1 + w) \cdot w)^2\)
  by simp
  also have \(\ldots =\)
    \((?n1 \cdot (r \cdot v) + ?n1 \cdot (s \cdot w) + (v^2 + w^2)) \cdot 2 + (\ldots \cdot ?n1 \cdot (r \cdot w) - ?n1 \cdot (s \cdot v)) \cdot 2\) \(\ldots\)
  by (simp add: eval-nat-numeral field-simps)
  also from m1 have \(\ldots =\)
    \((?n1 \cdot (r \cdot v) + ?n1 \cdot (s \cdot w) + ?n1 \cdot m1)^2 + (?n1 \cdot (r \cdot w) - ?n1 \cdot (s \cdot v))^2\)
  by simp
finally have \((?p \cdot ?n1) \cdot ?C = \?n1 \cdot 2 \cdot (r \cdot v + s \cdot w + m1)^2 + \?n1 \cdot 2 \cdot (r \cdot w - s \cdot v)^2\)
  by (simp add: eval-nat-numeral field-simps)

with m1 have \(?n1 \cdot 2 \cdot (?p \cdot m1) = \?n1 \cdot 2 \cdot (r \cdot v + s \cdot w + m1)^2 + (r \cdot w - s \cdot v)^2\)
  by (simp only: ac-simps power2-eq-square, simp add: distrib-left)

have \(?n1 \cdot 2 \cdot (?p \cdot m1 - (r \cdot v + s \cdot w + m1)^2 - (r \cdot w - s \cdot v)^2) = 0\)
  by (auto simp add: distrib-left right-diff-distrib)

moreover have \(\?n1 \cdot 2 \neq 0\)
  by (simp add: power2-eq-square)
ultimately show \(?thesis\) by simp
qed

hence \(g1pm1\): is-sum2sq \((?p \cdot m1)\) 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 \(v^2 > 0\)
    moreover have \(w^2 \geq 0\)
  moreover 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 $v = 0 \land w = 0$ by auto
with ru su have ?n1 dvd x \land ?n1 dvd y by (unfold dvd-def, auto)
hence ?n1 \^ 2 dvd x \^ 2 \land ?n1 \^ 2 dvd y \^ 2 by (simp add: zpower-zdvd-mono)
hence ?n1 \^ 2 dvd x \^ 2 + y \^ 2 by (simp only: dvd-add)
with xy have ?n1 \* ?n1 dvd ?n1 \* ?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 $?n1 = 1 \lor ?n1 = ?p$ by (unfold zprime-def, blast)
with H have $?Q 0$ by auto
ultimately have $?thesis$ by auto

ultimately show $?thesis$ by auto

moreover { assume $v^2 + 1 \* w^2 \neq 0$
moreover have $v^2 + w^2 \geq 0$

proof –
\hspace{1em} have $v^2 \geq 0 \land w^2 \geq 0$ by (auto simp only: zero-le-power2)
\hspace{1em} thus $?thesis$ by arith
qed
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)
\hspace{1em} assume $\neg m1 \geq 0$ hence $m1 < 0$ by simp
\hspace{1em} with n1pos have $?n1 \* m1 < ?n1 \* 0$ by (simp only: zmult-zless-mono2)
\hspace{1em} with m1 vwpos show False by simp
qed
ultimately have $?thesis$ m1 \> 0 by auto }
ultimately show $?thesis$ by auto

moreover from u v p have $?n1 \neq 0$ by simp
ultimately have $(nat m1) - 1 = n1$ by arith
with gfp pm1 have Qm1: $?Q ((nat m1) - 1)$ by auto
then obtain mn where tmp: $mn = (nat m1) - 1 \land ?Q mn$ by auto
moreover have $mn < n$
proof –
\hspace{1em} from tmp mn m1pos have int mn < int n by arith
\hspace{1em} thus $?thesis$ by arith
qed
moreover with H have $1 + int mn < ?p$ by auto
ultimately show $\exists m. m < n \land (1 + int m < ?p \longrightarrow \neg?vQ m)$ by auto
qed

moreover from u v p have $?n1 \neq 0$ by simp
moreover from tpos t-l-p have $1 + int tn < ?p \land tn = nat t - 1$
\hspace{1em} by arith
ultimately have $\neg?vQ ((nat t) - 1)$ by simp
moreover from tpos have $1 + int ((nat t) - 1) = t$ by arith
ultimately have $\neg is-sum2sq (?p\*t)$ by auto
with gfp t show False by simp
qed
hence $?thesis$ by (unfold is-sum2sq-def sum2sq-def, auto) }


ultimately show \( \text{thesis} \) by (auto simp add: less_le)
qed

end

2 Lagrange’s four-square theorem

theory FourSquares
imports ../Fermat3-4/IntNatAux ``` src/HOL/Old-Number-Theory/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
  \[ \{x^2 \mod p \mid 0 \leq x \leq n\} \text{ and } \{-1 - y^2 \mod p \mid 0 \leq y \leq n\} \]
  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.

definition
  \( \text{sum4sq} :: \text{int} \times \text{int} \times \text{int} \times \text{int} \Rightarrow \text{int} \) where
  \( \text{sum4sq} = (\lambda (a, b, c, d). a^2 + b^2 + c^2 + d^2) \)

definition
  \( \text{is-sum4sq} :: \text{int} \Rightarrow \text{bool} \) where
  \( \text{is-sum4sq} x \leftarrow (\exists a b c d. \text{sum4sq}(a, b, c, d) = x) \)

lemma mult-sum4sq: \( \text{sum4sq}(a, b, c, d) \ast \text{sum4sq}(p, q, r, s) = \\
  \text{sum4sq}(a*p + b*q + c*r + d*s, a*q + b*p - c*s + d*r, \\
  a*r + b*s - c*p + d*q, a*s - b*r + c*q - d*p) \)
  by (unfold \( \text{sum4sq-def} \), simp add: eval-nat-numeral field-simps)

lemma is-mult-sum4sq: \( \text{is-sum4sq} x \Longrightarrow \text{is-sum4sq} y \Longrightarrow \text{is-sum4sq} (x \ast y) \)
  by (unfold \( \text{is-sum4sq-def} \), auto simp only: mult-sum4sq, blast)

lemma mult-oddprime-is-sum4sq: \[ [ \text{zprime} \ p; \ p \in \text{zOdd} ] \Longrightarrow \]
  \( \exists t. \ 0 < t \land t < p \land \text{is-sum4sq} (p*t) \)
  proof
    assume \( p1: \text{zprime} \ p \)
    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 \(?A = \{y. 0 \leq y ∧ y < p\}\)
let \(?D = \{y. 0 \leq y ∧ y ≤ n\}\)
let \(?f = \%x. x^2 \mod p\)
let \(?g = \%x. (1+x^2) \mod p\)
let \(?A = \{?f, ?D\}
let \(?B = \{?g, ?D\}

have \( \text{finC} : \text{finite} \ ?C \) by (rule bdd-int-set-finite)
have \( \text{finD} : \text{finite} \ ?D \) by (rule bdd-int-set-finite)
from \( p0 \) have \( AsubC : ?A ⊆ ?C \) and \( BsubC : ?B ⊆ ?C \)
  by (auto simp add: pos-mod-conj)
with \( \text{finC} \) have \( \text{finA} : \text{finite} \ ?A \) and \( \text{finB} : \text{finite} \ ?B \)
  by (auto simp add: finite-subset)
from \( AsubC \ BsubC \) have \( \text{AunBsubC} : ?A ∪ ?B ⊆ ?C \) by (rule Un-least)
from \( p0 \) have \( \text{cardC} : \text{card} ?C = \text{nat} p \) by (simp only: card-bdd-int-set-le)
from \( n0 \) have \( \text{cardD} : \text{card} ?D = 1+ \text{nat} n \) by (simp only: card-bdd-int-set-le)
have \( \text{cardA} : \text{card} ?A = \text{card} ?D \)
proof

have \( \text{inj-on} \ ?f \ ?D \)
proof (unfold inj-on-def, auto)
  fix \( x \) fix \( y \)
  assume \( x0; 0 ≤ x \) and \( x0; 0 ≤ y \) and \( y0; y ≤ n \)
  and \( xyp: x^2 \mod p = y^2 \mod p \)
with \( p0 \) have \( x^2 = y^2 \) (mod \( p \)) by (simp only: zcong-zmod-eq)
  hence \( p \ dvd x^2 − y^2 \) by (simp only: zcong-def)
  hence \( p \ dvd (x+y)*(x−y) \) by (simp only: zspecial-product)
with \( p1 \) have \( p \ dvd x+y \) by (simp only: zdvd-not-zless)
  ultimately have \( −x+y > 0 \) by (auto simp add: zdvd-not-zless)
  with \( x0 \ y0 \) have \( x = y \) by auto

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 }

moreover
  { assume \( \text{ass} : p \ dvd x−y \)
    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 \( \text{ass} \) have \( −x−y < p \) by (auto simp add: zdvd-not-zless)
      with \( xn \ y0 \ n \ p0 \) show \( \text{False} \) 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 \( \text{ass} \) have \( p \ dvd −(x−y) \) by (simp only: dvd-minus-iff)
      ultimately show \( \text{False} \) by auto
    qed }
    ultimately show \( x=y \) by auto
  qed
with \( \text{finD} \) show \( \text{thesis} \) by (simp only: inj-on-iff-eq-card)
qed
have \( \text{cardB} : \text{card} \ ?B = \text{card} \ ?D \)
proof
  have inj-on ?g ?D
proof (unfold inj-on-def, auto)
  fix x fix y
  assume x0: \( 0 \leq x \) and xx: \( x \leq n \) and y0: \( 0 \leq y \) and yn: \( y \leq n \)
  and xyp: \((-1-x^2) \mod p = (-1-y^2) \mod p \)
  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 \(-1-y^2 + x^2 - y^2 = x^2 - y^2\) 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 \(x_n y_n n\) have \(x+y < p\) by auto
    ultimately have \(-x+y > 0\) by (auto simp add: zdvd-not-zless)
    with \(x_0 y_0\) have \(x = y\) by auto \} — both are zero
  moreover
  { assume ass: \(p \text{ dvd } x-y\)
    have \(x = y\)
    proof (rule ccontr, case-tac \(x-y \geq 0\), auto)
      assume \(x-y \geq 0\) and \(x \neq y\) hence \(x-y > 0\) by auto
      with ass have \(-x-y < p\) by (auto simp add: zdvd-not-zless)
      with \(x_0 y_0 n\) \(p0\) show \(False\) by auto
next
  assume \(0 \leq x-y\) hence \(y-x > 0\) by auto
  moreover from \(x_0 y_n n\) \(p0\) have \(y-x < p\) by auto
  ultimately have \(-p \text{ dvd } y-x\) by (auto simp add: zdvd-not-zless)
  moreover from ass have p dvd \(-x+y\) by (simp only: dvd-minus-iff)
  ultimately show \(False\) by auto
  qed
  ultimately show \(x=y\) by auto
  qed
  with \(\forall D\) show \(?thesis\) by (simp only: inj-on-iff-eq-card)
  qed
have \(?A \cap ?B \neq \{\}\)
proof (rule ccontr, auto)
  assume \(\neg \text{Disj} \): \(?A \cap ?B = \{\}\)
  from \(\text{card}A \text{ card}B \text{ card}D\) have \(2 + 2*(\text{nat } n) = \text{card } \text{?A} + \text{card } \text{?B}\) by auto
  also with \(\text{fin}A \text{ fin}B \text{ ADisj}\) have \(\ldots = \text{card } (\text{?A} \cup \text{?B})\)
    by (simp only: card-Un-disjoint)
  also with \(\text{fin}C \text{ AunBsubC}\) have \(\ldots \leq \text{card } \text{?C}\) by (simp only: card-mono)
  also with \(\text{card}C\) have \(\ldots = \text{nat } p\) by simp
  finally have \(2 + 2*(\text{nat } n) \leq \text{nat } p\) by simp
  with \(n\) show \(\text{False}\) by arith
  qed
then obtain \(z\) where \(z \in \text{?A} \land z \in \text{?B}\) by auto
then obtain \(x y\) where \(\text{xy: } x \in \text{?D} \land y \in \text{?D} \land z = x^2 \mod p \land\)
  \(z = (-1-y^2) \mod p\) by auto
with p0 have \([x^2=\cdot\cdot\cdot\cdot\cdot]\)(mod p) by (simp add: zcong-zmod-eq)
  hence p dvd \(x^2-(\cdot\cdot\cdot\cdot\cdot)\) by (simp only: zcong-def)
  moreover have \(x^2-(\cdot\cdot\cdot\cdot\cdot) = x^2+y^2+1\) by arith
ultimately have \( p \mid \text{sum4sq}(x,y,1,0) \) by \( \text{(auto simp add: sum4sq-def)} \)
then obtain \( t \) where \( \text{sum4sq}(x,y,1,0) = p \cdot t \) by \( \text{(auto simp add: dvd-def)} \)
hence \( \text{is-sum4sq} \ (p \cdot t) \) by \( \text{(unfold is-sum4sq-def, auto)} \)
moreover have \( t > 0 \land t < p \)
proof
  have \( x \cdot 2 \geq 0 \land y \cdot 2 \geq 0 \) by \( \text{(simp add: zero-le-power2)} \)
hence \( x \cdot 2 + y \cdot 2 + 1 > 0 \) by \text{arith}
with \( t \) have \( p \cdot t > 0 \) by \( \text{(unfold sum4sq-def, auto)} \)
moreover
  \{ assume \( t < 0 \) with \( p \cdot 0 \) have \( p \cdot t < p \cdot 0 \) by \( \text{(simp only: zmult-zless-mono2)} \)
    hence \( p \cdot t < 0 \) by \text{simp} \}
moreover
  \{ assume \( t = 0 \) hence \( p \cdot t = 0 \) by \text{simp} \}
ultimately have \( \neg t < 0 \land t \neq 0 \) by \text{auto}
thus \( t > 0 \) by \text{simp}
from \( xy \) have \( x \cdot 2 \leq n \cdot 2 \land y \cdot 2 \leq n \cdot 2 \) by \( \text{(auto simp add: power-mono)} \)
hence \( x \cdot 2 + y \cdot 2 + 1 \leq 2 \cdot n \cdot 2 + 1 \) by \text{auto}
with \( t \) have \( \text{contr}: \ p \cdot t \leq 2 \cdot n \cdot 2 + 1 \) by \( \text{(simp add: sum4sq-def)} \)
moreover
  \{ assume \( t > n + 1 \)
    with \( p \cdot 0 \) have \( p \cdot (n + 1) < p \cdot t \) by \( \text{(simp only: zmult-zless-mono2)} \)
    with \( n \) have \( p \cdot t > (2 \cdot n + 1) \cdot n + (2 \cdot n + 1) \cdot 1 \) by \( \text{(simp only: distrib-left)} \)
    hence \( p \cdot t > 2 \cdot n \cdot n + n + 2 \cdot n + 1 \) by \( \text{(simp only: distrib-right mul-1-left)} \)
    with \( n \cdot 0 \) have \( p \cdot t > 2 \cdot n \cdot 2 + 1 \) by \( \text{(simp add: power2-eq-square)} \} \}
ultimately have \( \neg t > n + 1 \) by \text{auto}
with \( n \cdot 0 \) \( n \) show \( t < p \) by \text{auto}
qed
ultimately show \( \neg \text{thesis} \) by \text{blast}
qed

lemma \text{zprime-is-sum4sq}: \( \text{zprime} \ p \Longrightarrow \text{is-sum4sq} \ p \)
proof (cases)
  assume \( p \cdot 2 \)\( =2 \)
  hence \( p = \text{sum4sq}(1,1,0,0) \) by \( \text{(auto simp add: sum4sq-def)} \)
  thus \( \neg \text{thesis} \) by \( \text{(auto simp add: is-sum4sq-def)} \)
next
assume \( \neg p = 2 \) and \( \text{prp: zprime} \ p \)
hence \( 2 < p \) by \( \text{(simp add: zprime-def)} \)
with \( \text{prp} \) have \( p \in \text{zOdd} \) by \( \text{(simp only: zprime-zOdd-eq-grt-2)} \)
with \( \text{prp} \) have \( \exists t. \ 0 < t \land t < p \land \text{is-sum4sq} \ (p \cdot t) \)
  by \( \text{(rule mult-oddprime-is-sum4sq)} \)
then obtain \( a b c d t \) where \( pt\text{-sol}: \ 0 < t \land t < p \land \text{sum4sq}(a,b,c,d) = p \cdot t \)
  by \( \text{(unfold is-sum4sq-def, blast)} \)
hence \( Qt: \ 0 < t \land t < p \land (\exists a1 a2 a3 a4. \ \text{sum4sq}(a1,a2,a3,a4) = p \cdot t) \)
  (is \( \neg Qt \) by \text{blast}
have \( \neg \text{Q1} \)
proof (rule \text{ecccontr})
  assume \( n\text{Q1}: \neg \text{Q} \ 1 \)
  have \( \neg \text{Q t} \)
proof (induct \( t \) rule: infinite-descent0-measure[where \( V=\lambda x. \ (nat \ x) - 1 \)], \text{clarify})
  fix \( x \ a \ b \ c \ d \)
  assume \( \text{nat x - 1 = 0 and x > 0 and s: sum4sq(a,b,c,d) = p \cdot x and x < p} \)
moreover hence \( x = 1 \) by arith
ultimately have \( ?Q \ 1 \) by auto
with \( \text{nQ1} \) show False by auto
next
\[
\text{fix} \ x
\]
\[
\text{assume} \ 0 < \text{nat}\ x - 1 \text{ and } \neg \neg ?Q \ x
\]
then obtain \( \text{a1 a2 a3 a4} \) where ass: \( 1 < x \land x < p \land \text{sum4sq}(a1,a2,a3,a4) = p*x \)
by auto
have \( \exists \ y, \text{nat} \ y - 1 < \text{nat} \ x - 1 \land ?Q \ y \)
proof (cases)
assume ev12: \( a1^2+a2^2 \in \text{zEven} \)
moreover have \( 2*a1*a2 \in \text{zEven} \) by (auto simp add: zEven-def)
ultimately have \( a1^2+a2^2+2*a1*a2 \in \text{zEven} \) by (rule even-plus-even)

\[
\text{hence} \ (a1+a2)^2 \in \text{zEven} \text{ by (auto simp add: zodd-power2 ac-simps)}
\]
from ev12\( \neg \neg \exists \ b1 b2 b3 b4. \text{sum4sq}(b1,b2,b3,b4) = p*x \)
\[
\text{hence} \ (a1+a2)^2 \in \text{zEven} \text{ by (auto simp add: power-preserves-even)}
\]
\[
\text{hence} \ a3^2+a4^2 \in \text{zEven} \text{ by auto}
\]
moreover have \( 2*a3*a4 \in \text{zEven} \) by (auto simp add: zEven-def)
ultimately have \( a3^2+a4^2+2*a3*a4 \in \text{zEven} \) by (rule even-plus-even)
\[
\text{hence} \ (a3+a4)^2 \in \text{zEven} \text{ by (auto simp add: zodd-power2 ac-simps)}
\]
\[
\text{hence} \ a3+a4 \in \text{zEven} \text{ by (auto simp add: power-preserves-even)}
\]
with \( \text{tmp ass show } ?\text{thesis by blast} \)
next
\[
\text{assume } \neg \neg \exists \ a1^2+a2^2 \in \text{zEven}
\]
\[
\text{hence} \ \text{odd12: } a1^2+a2^2 \in \text{zOdd by (simp add: odd-iff-not-even)}
\]
\[
\text{with } \text{ev1234: } a1^2+a2^2+a3^2+a4^2-(a1^2+a2^2) \in \text{zOdd}
\]
\[
\text{by (simp only: even-minus-odd)}
\]
\[
\text{hence} \ a3^2+a4^2 \in \text{zOdd by auto}
\]
show \( ?\text{thesis} \)
proof (cases)
\[
\text{assume ev1: } a1^2 \in \text{zEven}
\]
\[
\text{with odd12 have } \text{odd2: } a2^2 \in \text{zOdd by (rule even-plus-odd-prop2)}
\]
show \( ?\text{thesis} \)
proof (cases)
\[
\text{assume ev3: } a3^2 \in \text{zEven}
\]
\[
\text{with odd34 have } a4^2 \in \text{zOdd by (rule even-plus-odd-prop2)}
\]
\[
\text{with odd2 have } a2 \in \text{zOdd } \land \text{ a4 } \in \text{zOdd}
\]
\[
\text{by (auto simp add: power-preserves-odd)}
\]
\[
\text{hence} \ \text{tmp: } a2+a4 \in \text{zEven by (simp only: odd-plus-odd)}
\]
from ev3 ev1 have \( a1 \in \text{zEven} \land \text{ a3 } \in \text{zEven}
\]
\[
\text{by (auto simp add: power-preserves-even)}
\]
\[
\text{hence} \ \text{tmp2: } a1+a3 \in \text{zEven by (simp only: even-plus-even)}
\]
\[
\text{from ass have } \text{sum4sq}(a1,a3,a2,a4) = p*x
\]
\[
\text{by (auto simp add: sum4sq-def)}
\]
with tmp tmp2 show ?thesis by blast
next
assume ~ a3'2 ∈ zEven
hence odd3: a3'2 ∈ zOdd by (simp add: odd-iff-not-even)
with odd3' have a4'2 ∈ zEven by (rule even-plus-odd-prop1)
with ev1 have a1 ∈ zEven ∧ a4 ∈ zEven
  by (auto simp add: power-preserves-even)
thus obtain
next
by (simp only: a3'2 ∈ zEven)
odd12
hence with assume ¬
next
by (simp only: even-plus-odd-prop1)
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 ~ a1'2 ∈ zEven
hence odd1: a1'2 ∈ zOdd by (simp add: odd-iff-not-even)
with odd12 have ev2: a2'2 ∈ zEven by (rule even-plus-odd-prop1)
show ?thesis
next
by (auto simp add: even-plus-odd-prop2)
with odd3' have a4'2 ∈ zOdd by (rule even-plus-odd-prop2)
with ev1 have a1 ∈ zOdd ∧ a4 ∈ zOdd
  by (auto simp add: power-preserves-odd)
thus obtain
next
by (simp only: even-plus-odd-prop1)
from ev3 ev2 have a3 ∈ zEven ∧ a3 ∈ zEven
  by (auto simp add: power-preserves-even)
thus obtain
next
by (simp only: even-plus-odd-prop1)
from odd1 odd3 have a1 ∈ zOdd ∧ a3 ∈ zOdd
  by (auto simp add: power-preserves-odd)
thus obtain
next
by (simp only: even-plus-odd-prop1)
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 ∧
b1+b2 ∈ zEven ∧ b3+b4 ∈ zEven by auto
then obtain c1 c3 where c13: b1+b2 = 2*c1 ∧ b3+b4 = 2*c3
  by (auto simp add: zEven-def)
have $2 + b_2 \in z\text{Even} \land 2 + b_4 \in z\text{Even}$ by (auto simp add: zEven-def)

with $b$ have $b_1 + b_2 - 2 + b_2 \in z\text{Even} \land b_3 + b_4 - 2 + b_4 \in z\text{Even}$

by (auto simp only: even-minus-even)

moreover have $b_1 + b_2 - 2 + b_2 = b_1 - b_2 \land b_3 + b_4 - 2 + b_4 = b_3 - b_4$ by auto

ultimately have $b_1 - b_2 \in z\text{Even} \land b_3 - b_4 \in z\text{Even}$ by simp

then obtain $c_2 \cdot c_4$ where $c_2 \cdot c_4 : b_1 - b_2 = 2 + c_2 \land b_3 - b_4 = 2 + c_4$

by (auto simp add: zEven-def)

from $ex\ b_2 \ b_3 \ b_4 \ c_2 \ c_3 \ c_4$ have $y \in \{x \mid x = 2 + y\}$ by (auto simp add: zEven-def)

hence $4 * (p + y) = 2 * (p + x)$ by (simp add: ac-simps)

also from $b$ have $\ldots = 2 * b_1 \cdot 2 + 2 * b_2 \cdot 2 + 2 * b_3 \cdot 2 \land 2 + b_4 \cdot 2$

by (auto simp only: sum4sq-def)

also have $\ldots = (b_1 + b_2) \cdot 2 + (b_1 - b_2) \cdot 2 + (b_3 + b_4) \cdot 2 + (b_3 - b_4) \cdot 2$

by (auto simp add: add-right-diff-distrib)

also with $c_2 \cdot c_4$ have $\ldots = 4 * (c_1 \cdot 3 + c_2 \cdot 2 + c_3 \cdot 3 + c_4 \cdot 2)$

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

finally have $\sum_4 \text{sum4sq}(c_1, c_2, c_3, c_4) = p * y$ by (auto simp add: sum4sq-def)

moreover from $y$ have $\ldots \in \{x \mid x \in y \land y \in p \land (nat y) - 1 < (nat x) - 1\}$ by arith

ultimately show $?\text{thesis}$ by blast

next

assume $\neg x \in z\text{Even}$

hence $xodd : x \in z\text{Odd}$ by (simp add: odd-iff-not-even)

with $ass$ have $\exists c_1 \cdot c_2 \cdot c_3 \cdot c_4 \cdot \{2 * (a_1 \cdot a_1 * x) < x \land 2 * (a_2 \cdot c_2 * x) < x \land 2 * (a_3 \cdot c_3 * x) < x \land 2 * (a_4 \cdot c_4 * x) < x \}$

by (simp add: best-odd-division-abs)

then obtain $b_1 \cdot b_1 \cdot b_2 \cdot b_2 \cdot b_3 \cdot b_3 \cdot b_4 \cdot b_4$ where

bc-def: $b_1 = a_1 - c_1 * x \land b_2 = a_2 - c_2 * x \land b_3 = a_3 - c_3 * x \land b_4 = a_4 - c_4 * x$

and bc-abs: $2 * b_1 < x \land 2 * b_2 < x \land 2 * b_3 < x \land 2 * b_4 < x$

by blast

let $?B = b_1 \cdot 2 + b_2 \cdot 2 + b_3 \cdot 2 + b_4 \cdot 2$

let $?C = c_1 \cdot 2 + c_2 \cdot 2 + c_3 \cdot 2 + c_4 \cdot 2$

have $x \ \text{dvd} \ \ ?B$

proof (safe)

from bc-def ass have $\ldots$

let $?A_1 = a_1 * b_1 + a_2 * b_2 + a_3 * b_3 + a_4 * b_4$

let $?A_2 = a_1 * b_1 - a_2 * b_1 - a_3 * b_4 + a_4 * b_3$

let $?A_3 = a_1 * b_3 + a_2 * b_4 - a_3 * b_1 - a_4 * b_2$

let $?A_4 = a_1 * b_4 - a_2 * b_3 + a_3 * b_2 - a_4 * b_1$

let $?A = \text{sum4sq}(?A_1, ?A_2, ?A_3, ?A_4)$

have $x \ \text{dvd} \ ?A_1 \land x \ \text{dvd} \ ?A_2 \land x \ \text{dvd} \ ?A_3 \land x \ \text{dvd} \ ?A_4$

proof (safe)

from bc-def have $\ldots$

let $?A_1 = (b_1 + c_1 \cdot x) * b_1 + (b_2 + c_2 \cdot x) * b_2 + (b_3 + c_3 \cdot x) * b_3 + (b_4 + c_4 \cdot x) * b_4$

by simp

also with $y$ have $\ldots = x * (y + c_1 \cdot b_1 + c_2 \cdot b_2 + c_3 \cdot b_3 + c_4 \cdot b_4)$
by (auto simp add: distrib-left power2-eq-square ac-simps)

finally show \( x \ dvd \ ?A1 \) by auto

from bc-def have
\[
?A2 = (b1 + c1 \cdot x) \cdot b2 - (b2 + c2 \cdot x) \cdot b1 - (b3 + c3 \cdot x) \cdot b4 + (b4 + c4 \cdot x) \cdot b3
\]
by simp

also have \( \ldots = x \cdot (c1 \cdot b2 - c2 \cdot b1 - c3 \cdot b4 + c4 \cdot b3) \)
by (auto simp add: distrib-left right-diff-distrib ac-simps)

finally show \( x \ dvd \ ?A2 \) by auto

from bc-def have
\[
?A3 = (b1 + c1 \cdot x) \cdot b3 + (b2 + c2 \cdot x) \cdot b4 - (b3 + c3 \cdot x) \cdot b1 - (b4 + c4 \cdot x) \cdot b2
\]
by simp

also have \( \ldots = x \cdot (c1 \cdot b3 + c2 \cdot b4 - c3 \cdot b1 - c4 \cdot b2) \)
by (auto simp add: distrib-left right-diff-distrib ac-simps)

finally show \( x \ dvd \ ?A3 \) by auto

from bc-def have
\[
?A4 = (b1 + c1 \cdot x) \cdot b4 - (b2 + c2 \cdot x) \cdot b3 + (b3 + c3 \cdot x) \cdot b2 - (b4 + c4 \cdot x) \cdot b1
\]
by simp

also have \( \ldots = x \cdot (c1 \cdot b4 - c2 \cdot b3 + c3 \cdot b2 - c4 \cdot b1) \)
by (auto simp add: distrib-left right-diff-distrib ac-simps)

finally show \( x \ dvd \ ?A4 \) by auto

qed
	hen then obtain \( d1 \ d2 \ d3 \ d4 \) where \( d \):

\( ?A1 = x \cdot d1 \land ?A2 = x \cdot d2 \land ?A3 = x \cdot d3 \land ?A4 = x \cdot d4 \)

by (auto simp add: ded-def)

let \( ?D = \text{sum}4\text{sq}(d1, d2, d3, d4) \)

from \( d \) have \( x \cdot 2 \cdot ?D = ?A \)
by (auto simp only: sum4sq-def power-mult-distrib distrib-left)

also have \( \ldots = \text{sum}4\text{sq}(a1, a2, a3, a4) \cdot \text{sum}4\text{sq}(b1, b2, b3, b4) \)
by (simp only: mult-sum4sq)

also with \( y \) ass have \( \ldots = (p \cdot x) \cdot (x \cdot y) \) by (auto simp add: sum4sq-def)
also have \( \ldots = x \cdot 2 \cdot (p \cdot y) \) by (simp only: power2-eq-square ac-simps)

finally have \( x \cdot 2 \cdot (?D - p \cdot y) = 0 \) by (auto simp add: right-diff-distrib)

with \( \text{ass have} \ ?D = p \cdot y \) by auto

moreover have \( y \cdot l \cdot x : y < x \)

proof
- have \( 4 \cdot b1 \cdot 2 = (2 \cdot |b1|) \cdot 2 \land 4 \cdot b2 \cdot 2 = (2 \cdot |b2|) \cdot 2 \land 4 \cdot b3 \cdot 2 = (2 \cdot |b3|) \cdot 2 \land 4 \cdot b4 \cdot 2 = (2 \cdot |b4|) \cdot 2 \)
  by (auto simp add: power-mul-to-power2 abs-mult power2-acts)

with bc-abs have \( 4 \cdot b1 \cdot 2 < x \cdot 2 \land 4 \cdot b2 \cdot 2 < x \cdot 2 \land 4 \cdot b3 \cdot 2 < x \cdot 2 \land 4 \cdot b4 \cdot 2 < x \cdot 2 \)
  using power-strict-mono [of 2 \cdot |b| x 2 for b]
  by auto

hence \( ?B < x \cdot 2 \) by auto

with \( y \) have \( x \cdot (x - y) > 0 \)
by (auto simp add: power2-eq-square right-diff-distrib)

moreover from \( \text{ass have} \ x > 0 \) by simp

ultimately show \( \text{thesis} \) by (auto dest: pos-mult-pos)

qed

moreover have \( y > 0 \)

proof
- have \( b2\text{pos}: b1 \cdot 2 \geq 0 \land b2 \cdot 2 \geq 0 \land b3 \cdot 2 \geq 0 \land b4 \cdot 2 \geq 0 \)
  by (auto simp add: zero-le-power2)

hence \( ?B = 0 \lor ?B > 0 \) by arith
moreover
{ assume ?B = 0
moreover from b2pos have
?B−b1ˆ2 ≥ 0 ∧ ?B−b2ˆ2 ≥ 0 ∧ ?B−b3ˆ2 ≥ 0 ∧ ?B−b4ˆ2 ≥ 0 by arith
ultimately have b1ˆ2 ≤ 0 ∧ b2ˆ2 ≤ 0 ∧ b3ˆ2 ≤ 0 ∧ b4ˆ2 ≤ 0 by auto
with b2pos have b1ˆ2 = 0 ∧ b2ˆ2 = 0 ∧ b3ˆ2 = 0 ∧ b4ˆ2 = 0 by arith
hence b1 = 0 ∧ b2 = 0 ∧ b3 = 0 ∧ b4 = 0 by auto
with bc-def have x dvd a1 ∧ x dvd a2 ∧ x dvd a3 ∧ x dvd a4
by auto
hence xˆ2 dvd a1ˆ2 ∧ xˆ2 dvd a2ˆ2 ∧ xˆ2 dvd a3ˆ2 ∧ xˆ2 dvd a4ˆ2
by (auto simp only: zpower-zdvd-mono)
ultimately have ?thesis by (auto dest: zdvd-mult-cancel)
moreover 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 have y < p by auto
ultimately show ?thesis by blast
qed
thus ∃ y. nat y − 1 < nat x − 1 ∧ ¬ ¬ ?Q y by blast
qed

thus is-sum4sq p by (auto simp add: is-sum4sq-def)

qed

theorem four-squares: (n::int) ≥ 0 ==> ∃ a b c d. aˆ2 + bˆ2 + cˆ2 + dˆ2 = 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=¬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 \( ?X = \text{int} (\prod ps) \)
assume primel ps \( \implies \) is-sum4sq \( ?X \) and primel \((p\#ps)\)
hence prime p and \( x: \text{is-sum4sq} \ ?X \) by (auto simp add: primel-hd-tl)
hence zprime \((\text{int} \ p)\) by (simp only: prime-impl-zprime-int)
with \( x \) have \( \text{is-sum4sq}(\text{int} \ p) \) by (simp add: is-mult-sum4sq)
thus is-sum4sq \((\text{int} \ (p\#\prod ps))\) by (auto simp only: int-mult)
qed
with ps npos have is-sum4sq n by auto 
ultimately have is-sum4sq n by auto 
thus \( ?\text{thesis} \) by (auto simp only: is-sum4sq-def sum4sq-def)
qed

end

References


