

Exponents 3 and 4 of Fermat's Last Theorem and the Parametrisation of Pythagorean Triples

Roelof Oosterhuis
University of Groningen

December 12, 2009

Abstract

This document gives a formal proof of the cases $n = 3$ and $n = 4$ (and all their multiples) of Fermat's Last Theorem: if $n > 2$ then for all integers x, y, z :

$$x^n + y^n = z^n \implies xyz = 0.$$

Both proofs only use facts about the integers and are developed along the lines of the standard proofs (see, for example, sections 1 and 2 of the book by Edwards [Edw77]).

First, the framework of 'infinite descent' is being formalised and in both proofs there is a central role for the lemma

$$\gcd(a, b) = 1 \wedge ab = c^n \implies \exists k : |a| = k^n.$$

Furthermore, the proof of the case $n = 4$ uses a parametrisation of the Pythagorean triples. The proof of the case $n = 3$ contains a study of the quadratic form $x^2 + 3y^2$. This study is completed with a result on which prime numbers can be written as $x^2 + 3y^2$.

The case $n = 4$ of FLT, in contrast to the case $n = 3$, has already been formalised (in the proof assistant Coq) [DM05]. The parametrisation of the Pythagorean Triples can be found as number 23 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). The author wants to thank Clemens Ballarin (TU München) and Freek Wiedijk (RU Nijmegen) for their support. For more information see [Oos07].

Contents

1	Powers, prime numbers and divisibility	3
1.1	Auxiliary results	3
1.2	Parity of integers	6
1.3	Powers of natural numbers	7
1.4	Powers of integers	10
1.5	Facts about small powers of integers	14
2	Pythagorean triples and Fermat's last theorem, case $n = 4$	16
2.1	Parametrisation of Pythagorean triples (over \mathbb{N} and \mathbb{Z})	17
2.2	Fermat's last theorem, case $n = 4$	22
3	The quadratic form $x^2 + Ny^2$	27
3.1	Definitions and auxiliary results	27
3.2	Basic facts if $N \geq 1$	28
3.3	Multiplication and division	29
3.4	Uniqueness ($N > 1$)	39
3.5	The case $N = 3$	43
3.6	Existence ($N = 3$)	54
4	Fermat's last theorem, case $n = 3$	57

1 Powers, prime numbers and divisibility

theory *IntNatAux*

imports

~~/src/HOL/Number-Theory/Factorization

~~/src/HOL/Number-Theory/EvenOdd

begin

Contains lemmas about divisibility and coprimality of powers, as well as some results about parities and small powers. Most lemmas are developed for the integers as well as for the natural numbers.

1.1 Auxiliary results

lemma *make-relprime*:

$(a \neq 0 \vee b \neq 0) \implies \exists c d. a = \text{gcd } a b * c \wedge b = \text{gcd } a b * d \wedge \text{gcd } c d = 1$

proof –

assume *ab0*: $a \neq 0 \vee b \neq 0$

let $?g = \text{gcd } a b$

have $?g \text{ dvd } a \wedge ?g \text{ dvd } b$ **by** *auto*

then obtain $c d$ **where** *abcd*: $a = ?g * c \wedge b = ?g * d$ **by** (*auto simp add: dvd-def*)

moreover have $\text{gcd } c d = 1$

proof –

from *abcd* **have** $?g * \text{gcd } c d = ?g$ **by** (*auto simp add: gcd-mult-distrib2*)

moreover with *ab0* **have** $?g \neq 0$ **by** (*simp add: gcd-zero*)

ultimately show *?thesis* **by** *simp*

qed

ultimately show *?thesis* **by** *auto*

qed

lemma *factor-exists-general*: $(a::\text{nat}) \neq 0 \implies (\exists ps. \text{primel } ps \wedge \text{prod } ps = a)$

proof –

assume *a0*: $a \neq 0$

show *?thesis*

proof (*case-tac a=1*)

assume $a=1$ **hence** $\text{primel } [] \wedge \text{prod } [] = a$ **by** (*auto simp add: primel-def*)

thus *?thesis* **by** *auto*

next

assume $a \neq 1$ **with** *a0* **have** $a > \text{Suc } 0$ **by** *auto*

thus *?thesis* **by** (*rule factor-exists*)

qed

qed

lemma *make-zrelprime*: $(a \neq 0 \vee b \neq 0)$

$\implies \exists c d. a = \text{zgcd } a b * c \wedge b = \text{zgcd } a b * d \wedge \text{zgcd } c d = 1$

proof –

assume *ab0*: $a \neq 0 \vee b \neq 0$

let $?g = \text{zgcd } a b$

have $?g \text{ dvd } a \wedge ?g \text{ dvd } b$ **by** *auto*

then obtain $c d$ **where** *abcd*: $a = ?g * c \wedge b = ?g * d$ **by** (*auto simp add: dvd-def*)

moreover have $\text{zgcd } c d = 1$

proof –

```

from abcd have  $?g * zgcd\ c\ d = ?g$ 
  by (auto simp add: zgcd-zmult-distrib2 zgcd-geq-zero)
moreover with ab0 have  $?g \neq 0$  by (auto simp add: zgcd-def gcd-zero)
ultimately show ?thesis by auto
qed
ultimately show ?thesis by auto
qed

```

```

lemma int-nat-abs-eq-abs:  $int(nat|x::int|) = |x|$ 
by simp

```

```

lemma prime-impl-zprime-int:  $prime\ (a::nat) \implies zprime\ (int\ a)$ 

```

```

proof -
  assume pra: prime a
  show zprime (int a)
  proof -
    from pra have agr1:  $1 < int\ a$  by (unfold prime-def, auto)
    moreover have  $!!m. m \geq 0 \wedge m\ dvd\ int\ a \wedge m \neq int\ a \implies m=1$ 
    proof -
      { fix m assume  $m: m \geq 0 \wedge m\ dvd\ int\ a \wedge m \neq int\ a$ 
        then obtain k::int where  $k: int\ a = m*k$  by (auto simp add: dvd-def)
        from m have  $int\ (nat\ m) = m$  by auto
        with k have  $int\ a = (int\ (nat\ m)) * k$  by simp
        hence  $nat\ (int\ a) = nat\ ((int\ (nat\ m)) * k)$  by simp
        hence  $a = nat\ ((int\ (nat\ m)) * k)$  by (simp only: nat-int)
        also have  $\dots = (nat\ m) * (nat\ k)$  by (simp add: nat-int nat-mult-distrib)
        finally have  $nat\ m\ dvd\ a$  by auto
        with pra have  $nat\ m = a \vee nat\ m = 1$  by (auto simp add: prime-def)
        moreover from m have  $nat\ m \neq a$  by auto
        ultimately have  $nat\ m = 1$  by auto
        hence  $m = 1$  by arith }
      thus  $!!m. m \geq 0 \wedge m\ dvd\ int\ a \wedge m \neq int\ a \implies m=1$  by auto
    }
  qed
  ultimately show ?thesis by (auto simp add: zprime-def)
qed

```

```

lemma zprime-factor-exists:  $(a::int) > 1 \implies \exists\ p. zprime\ p \wedge p\ dvd\ a$ 

```

```

proof -
  assume a1:  $a > 1$  hence  $a: int\ (nat\ a) = a$  by (auto simp add: int-nat-eq)
  with a1 have  $nat\ a > 1$  by auto
  hence  $\exists\ l. primel\ l \wedge prod\ l = nat\ a$  by (simp only: factor-exists)
  then obtain l where  $l: primel\ l \wedge prod\ l = nat\ a$  by (auto)
  show ?thesis
  proof (cases l)
    case Nil with l have  $nat\ a = 1$  by auto
    with a1 show ?thesis by arith
  next
    case (Cons p ps)
    with l have  $nat\ a = p*prod\ ps$  and  $p: prime\ p$  by (auto simp add: primel-def)
    hence  $int\ (nat\ a) = (int\ p)*int(prod\ ps)$ 
    by (auto simp add: int-mult)
  qed

```

```

with  $a\ p$  have  $zprime\ (int\ p) \wedge int\ p\ dvd\ a$ 
by  $(auto\ simp\ add:\ prime-impl-zprime-int)$ 
thus  $?thesis$  by  $blast$ 
qed
qed

```

```

lemma  $best-division-abs:\ (x::int) > 0 \implies \exists\ n.\ 2 * |y - n*x| \leq x$ 
proof -

```

```

  assume  $x0:\ x > 0$ 
  then obtain  $b$  where  $b \geq 0 \wedge b < x \wedge [y = b] (mod\ x)$ 
    by  $(blast\ dest:\ zcong-zless-unique)$ 
  hence  $x\ dvd\ (y-b)$  by  $(simp\ only:\ zcong-def)$ 
  then obtain  $m$  where  $y-b = x*m$  by  $(auto\ simp\ add:\ dvd-def)$ 
  hence  $m:\ b = y - m*x$  by  $(simp\ only:\ mult-ac)$ 
  show  $?thesis$ 
  proof  $(cases)$ 
    assume  $2*|b| \leq x$ 
    with  $m$  show  $?thesis$  by  $auto$ 
  next
    assume  $\neg 2*|b| \leq x$ 
    with  $b$  have  $bx:\ 2*b > x$  by  $auto$ 
    hence  $bx1:\ 2*(x-b) < x$  by  $auto$ 
    from  $b$  have  $bx2:\ b-x < 0$  by  $auto$ 
    obtain  $n$  where  $n = m+1$  by  $simp$ 
    hence  $y - n*x = y - m*x - x$  by  $(simp\ only:\ zadd-zmult-distrib\ zmult-1)$ 
    with  $m$  have  $n:\ y - n*x = b-x$  by  $simp$ 
    with  $bx2$  have  $pos:\ -y + n*x > 0$  by  $simp$ 
    moreover from  $n\ bx1$  have  $2*(-y + n*x) < x$  by  $simp$ 
    ultimately have  $2*|y - n*x| < x$  by  $simp$ 
    hence  $2*|y - n*x| \leq x$  by  $(unfold\ zabs-def,\ auto)$ 
    thus  $?thesis$  by  $auto$ 

```

```

qed
qed

```

```

lemma  $best-odd-division-abs:\ [\ (x::int) > 0;\ x \in zOdd ]$ 
   $\implies \exists\ n.\ 2 * |y - n*x| < x$ 

```

```

proof -
  assume  $x > 0$  and  $odd:\ x \in zOdd$ 
  then obtain  $n$  where  $n:\ 2 * |y - n*x| \leq x$  by  $(auto\ dest:\ best-division-abs)$ 
  moreover have  $x \neq 2 * |y - n*x|$ 
  proof  $(rule\ ccontr,\ clarsimp)$ 
    assume  $x = 2*|y - n*x|$ 
    hence  $x \in zEven$  by  $(unfold\ zEven-def,\ auto)$ 
    with  $odd$  show  $False$  by  $(auto\ simp\ only:\ odd-iff-not-even)$ 
  qed
  ultimately have  $2*|y - n*x| < x$  by  $simp$ 
  thus  $?thesis$  by  $auto$ 
qed

```

```

lemma  $zprime-2:\ zprime\ 2$ 
apply  $(auto\ simp\ add:\ zprime-def)$ 
apply  $(frule\ zdvd-imp-le,\ simp)$ 

```

apply (*auto simp add: order-le-less dvd-def*)
done

lemma *zgcd1-iff-no-common-primedivisor:*

$(zgcd\ a\ b=1) = (\neg(\exists\ p.\ zprime\ p \wedge p\ dvd\ a \wedge p\ dvd\ b))$

proof (*rule ccontr, auto*)

fix p **assume** $ab: zgcd\ a\ b=1$ **and** $p\ dvd\ a$ **and** $p\ dvd\ b$ **and** $p: zprime\ p$

hence $p\ dvd\ a \wedge p\ dvd\ b$ **by** *simp*

hence $p\ dvd\ zgcd\ a\ b$ **by** (*simp add: zgcd-greatest-iff*)

with $ab\ p$ **show** *False* **by** (*unfold zprime-def, auto*)

next

let $?g = zgcd\ a\ b$

assume $ps: \forall\ p.\ zprime\ p \longrightarrow p\ dvd\ a \longrightarrow \neg\ p\ dvd\ b$

assume $g1: ?g \neq 1$

moreover **have** $?g \neq 0$

proof (*rule ccontr, simp*)

assume $a = 0 \wedge b = 0$

hence $2\ dvd\ a \wedge 2\ dvd\ b$ **by** *simp*

with ps **show** *False* **by** (*auto simp add: zprime-2*)

qed

moreover **have** $?g \geq 0$ **by** (*rule zgcd-geq-zero*)

ultimately **have** $?g > 1$ **by** *arith*

then **obtain** p **where** $zprime\ p \wedge p\ dvd\ ?g$

by (*frule-tac a=?g in zprime-factor-exists, auto*)

hence $zprime\ p \wedge p\ dvd\ a \wedge p\ dvd\ b$ **by** (*simp add: zgcd-greatest-iff*)

with ps **show** *False* **by** *auto*

qed

lemma *pos-zmult-pos:* $a > (0::int) \Longrightarrow a*b > 0 \Longrightarrow b > 0$

apply (*case-tac b = 0, auto*)

apply (*rule ccontr, subgoal-tac b < 0, auto*)

apply (*subgoal-tac a*b < a*0, auto dest: zmult-zless-mono2*)

done

1.2 Parity of integers

lemma *power-preserves-even:* $n > 0 \Longrightarrow (x^n \in zEven) = (x \in zEven)$

apply (*induct n, auto simp add: even-times-either*)

apply (*case-tac n≠0, auto dest: even-product*)

done

lemma *power-preserves-odd:* $n > 0 \Longrightarrow (x^n \in zOdd) = (x \in zOdd)$

apply (*induct n, auto, rule odd-mult-odd-prop, auto*)

apply (*case-tac n≠0, auto dest: odd-times-odd*)

done

lemma *even-plus-odd:* $a \in zEven \Longrightarrow b \in zOdd \Longrightarrow a+b \in zOdd$

apply (*auto simp add: zEven-def zOdd-def*)

apply (*rule-tac x=k+ka in exI, auto*)

done

lemma *odd-plus-odd:* $a \in zOdd \Longrightarrow b \in zOdd \Longrightarrow a+b \in zEven$

```

apply (auto simp add: zEven-def zOdd-def)
apply (rule-tac x=1+k+ka in exI, auto)
done

```

```

lemma even-plus-odd-prop1: a+b ∈ zOdd ⇒ a ∈ zOdd ⇒ b ∈ zEven
by (subgoal-tac a+b - a ∈ zEven, auto dest: odd-minus-odd)

```

```

lemma even-plus-odd-prop2: a+b ∈ zOdd ⇒ a ∈ zEven ⇒ b ∈ zOdd
by (subgoal-tac a+b - a ∈ zOdd, auto dest: odd-minus-even)

```

1.3 Powers of natural numbers

```

lemma gcd-1-power-left-distrib: gcd a b = 1 ⇒ gcd (a^n) b = 1
by (induct n, auto simp add: gcd-mult-cancel)

```

NB: the next (identical) lemma is just added to illustrate the difference between Isar and Isabelle scripting.

```

lemma alternative-gcd-1-power-left-distrib: gcd a b = 1 ⇒ gcd(a^n) b = 1
proof -
  assume ab: gcd a b=1
  thus gcd (a^n) b = 1
  proof (induct n)
    case 0
    show gcd (a^0) b = 1 by auto
  next
    case (Suc n)
    hence gcd (a^n) b = 1 by simp
    with ab have gcd (a*a^n) b = 1 by (simp only: gcd-mult-cancel)
    thus gcd (a^Suc n) b = 1 by simp
  qed
qed

```

```

lemma gcd-1-power-distrib: gcd a b = 1 ⇒ gcd(a^n) (b^n) = 1
proof -
  assume gcd a b=1
  hence gcd (a^n) b = 1 by (rule gcd-1-power-left-distrib)
  hence gcd b (a^n) = 1 by (simp only: gcd-commute)
  hence gcd (b^n) (a^n) = 1 by (rule gcd-1-power-left-distrib)
  thus gcd (a^n) (b^n) = 1 by (simp only: gcd-commute)
qed

```

```

lemma gcd-power-distrib: gcd a b ^ n = gcd (a^n) (b^n)
proof cases
  assume a=0 ∧ b=0
  thus ?thesis by (auto simp add: power-0-left)
next
  let ?g = gcd a b
  assume ¬ (a=0 ∧ b=0)
  hence a ≠ 0 ∨ b ≠ 0 by simp
  then obtain c d where abcd: a = ?g*c ∧ b = ?g*d ∧ gcd c d=1
    by (frule-tac a=a in make-relprime, auto)
  moreover have (?g*c)^n = ?g^n * c^n ∧ (?g*d)^n = ?g^n * d^n

```

by (simp add: power-mult-distrib)
ultimately have $\gcd(a^n)(b^n) = ?g^n * \gcd(c^n)(d^n)$ by (simp only: gcd-mult-distrib2)
moreover from *abcd* have $\gcd(c^n)(d^n) = 1$ by (simp only: gcd-1-power-distrib)
ultimately show *?thesis* by simp
qed

Useful lemma: if prime $p|a^n$ then $p|a$.

lemma prime-dvd-power: $\llbracket \text{prime } p; p \text{ dvd } a^n \rrbracket \implies p \text{ dvd } a$
proof (induct n)
case 0 hence prime $p \wedge p = 1$ by auto
thus *?thesis* by auto
next case (Suc n) hence IH: prime $p \wedge p \text{ dvd } a^n \implies p \text{ dvd } a$ by auto
assume p : prime p and $p \text{ dvd } a^{Suc\ n}$
hence $p \text{ dvd } a * a^n$ by simp
with p have $p \text{ dvd } a \vee p \text{ dvd } a^n$ by (simp add: prime-dvd-mult)
with IH and p show $p \text{ dvd } a$ by auto
qed

lemma prime-power-dvd-cancel-right:
 $\llbracket \text{prime } p; \neg p \text{ dvd } b; p^n \text{ dvd } a * b \rrbracket \implies p^n \text{ dvd } a$
proof -
assume p : prime p and pb : $\neg p \text{ dvd } b$
hence $p1$: $p > 1$ by (simp add: prime-def)
have *!!a. $p^n \text{ dvd } a * b \longrightarrow p^n \text{ dvd } a$*
proof (induct n)
case 0 thus *?case* by auto
next
case (Suc n) hence IH: *!!a. $p^n \text{ dvd } a * b \longrightarrow p^n \text{ dvd } a$* .
fix a show $p^{Suc\ n} \text{ dvd } a * b \longrightarrow p^{Suc\ n} \text{ dvd } a$
proof (auto)
assume $ppnab$: $p * p^n \text{ dvd } a * b$
hence $p \text{ dvd } a * b$ by (rule dvd-mult-left)
with p have $p \text{ dvd } a \vee p \text{ dvd } b$ by (rule prime-dvd-mult)
with pb have $p \text{ dvd } a$ by simp
then obtain k where apk : $a = p * k$ by (auto simp add: dvd-def)
with $ppnab$ have $p * p^n \text{ dvd } p * (k * b)$ by (auto simp add: mult-ac)
with $p1$ have $p^n \text{ dvd } k * b$ by (auto dest: dvd-mult-cancel)
with IH have $p^n \text{ dvd } k$..
with apk show $p * p^n \text{ dvd } a$ by (simp add: mult-dvd-mono)
qed
qed
thus $p^n \text{ dvd } a * b \implies p^n \text{ dvd } a$..
qed

Helping lemma: if $n > 0$ then $a^n | b^n \iff a | b$.

lemma nat-power-dvd-mono: $n \neq 0 \implies (a^n \text{ dvd } b^n) = (a \text{ dvd } (b::nat))$
proof
assume $n \neq 0$
then obtain m where mn : $n = \text{Suc } m$
by (frule-tac $n=n$ in not0-implies-Suc, auto)
assume $a^n \text{ dvd } b^n$
then obtain k where k : $b^n = a^n * k$ by (auto simp add: dvd-def)

moreover have $\text{gcd } (a^n) ((a^n)*k) = (a^n) * \text{gcd } 1 k$ **by** (*simp add: gcd-mult-distrib2*)
ultimately have $\text{gcd } (a^n) (b^n) = a^n$ **by** (*auto simp add: gcd-commute gcd-1*)
hence $\text{gcd } a b^n = a^n$ **by** (*simp add: gcd-power-distrib*)
with mn **have** $a = \text{gcd } a b$ **by** (*rule-tac n=m in power-inject-base, auto*)
moreover have $\text{gcd } a b \text{ dvd } b$ **by** (*rule gcd-dvd2*)
ultimately show $a \text{ dvd } b$ **by** *simp*

next

assume $a \text{ dvd } b$
then obtain k **where** $b = a * k$ **by** (*auto simp add: dvd-def*)
hence $b^n = a^n * k^n$ **by** (*simp only: power-mult-distrib*)
thus $a^n \text{ dvd } b^n$ **by** *auto*

qed

Theorem: if $n > 0$ and $\text{gcd } ab = 1$ and $ab = c^n$ then $\exists k : a = k^n$. Proof uses induction on the number of prime factors of c .

theorem *nat-relprime-power-divisors*:

assumes $npos: n \neq 0$ **and** $abcn: a*b = c^n$ **and** $relprime: \text{gcd } a b = 1$
shows $\exists k. a = k^n$

proof –

from $npos$ **obtain** m **where** $mn: n = \text{Suc } m$
by (*frule-tac n=n in not0-implies-Suc, auto*)

show *?thesis*

proof (*case-tac c=0*)

assume $c=0$ **with** $abcn\ npos\ mn$ **have** $a*b = 0$ **by** (*auto simp only: power-0-Suc*)

hence $a=0 \vee b=0$ **by** *auto*

moreover

{ **assume** $a=0$ **with** $mn\ npos$ **have** $a = 0^n$ **by** (*auto simp only: power-0-Suc*)

hence *?thesis* **by** *blast* }

moreover

{ **assume** $b=0$ **with** $relprime$ **have** $a = 1^n$ **by** (*auto simp only: gcd-0 power-one*)

hence *?thesis* **by** *blast* }

ultimately show *?thesis* **by** *blast*

next

assume $c \neq 0$ **then obtain** xs **where** $xs: \text{primel } xs \wedge \text{prod } xs = c$

by (*frule-tac a=c in factor-exists-general, auto*)

moreover have

$!!a\ b. (\text{primel } xs \wedge a*b = (\text{prod } xs)^n \wedge \text{gcd } a\ b=1) \implies \exists k. a = k^n$

proof (*induct xs*)

case *Nil*

hence $a = 1^n$ **by** *simp*

thus $\exists k. a = k^n$..

next

case (*Cons p ps*)

hence $ass: \text{primel } ps \wedge \text{prime } p \wedge a*b = p^n * (\text{prod } ps)^n \wedge \text{gcd } a\ b=1$

and $IH: !!a\ b. \text{primel } ps \wedge a*b = (\text{prod } ps)^n \wedge \text{gcd } a\ b=1 \implies \exists k. a = k^n$

by (*auto simp add: primel-def power-mult-distrib*)

hence $pnab: p^n \text{ dvd } a*b$ **and** $pn0: p^n \neq 0$

by (*auto simp add: prime-def*)

moreover

{ **assume** $pa: p \text{ dvd } a$

have $\neg p \text{ dvd } b$

proof (*rule ccontr, simp*)

```

    assume p dvd b
    with pa have p dvd gcd a b by (simp add: gcd-greatest-iff)
    with ass show False by (auto simp add: prime-def)
qed
with ass pnab have p^n dvd a by (simp add: prime-power-dvd-cancel-right)
then obtain A where A: a = p^n * A by (auto simp add: dvd-def)
with ass pn0 have A*b = (prod ps)^n by auto
moreover have gcd A b=1
proof -
  let ?g = gcd A b
  have ?g dvd A ∧ ?g dvd b by (simp add: gcd-greatest)
  with A have ?g dvd a ∧ ?g dvd b by (simp add: dvd-mult)
  hence ?g dvd gcd a b by (simp only: gcd-greatest)
  with ass show ?g = 1 by auto
qed
moreover from IH ass have
  A*b = (prod ps)^n ∧ gcd A b=1 ⇒ ∃ k. A = k^n by auto
ultimately have ∃ k. A = k^n by auto
then obtain k where A = k^n by auto
with A have a = (p*k)^n by (auto simp add: power-mult-distrib)
hence ∃ k. a = k^n by blast }
moreover
{ assume ¬ p dvd a
  moreover from ass pnab have p^n dvd b*a ∧ prime p
    by (auto simp only: mult-ac)
  ultimately have p^n dvd b by (simp add: prime-power-dvd-cancel-right)
  then obtain B where B: b = p^n * B by (auto simp add: dvd-def)
  with ass pn0 have a*B = (prod ps)^n by auto
  moreover have gcd a B=1
  proof -
    let ?g = gcd a B
    have ?g dvd a ∧ ?g dvd B by (simp add: gcd-greatest)
    with B have ?g dvd a ∧ ?g dvd b by (simp add: dvd-mult)
    hence ?g dvd gcd a b by (simp only: gcd-greatest)
    with ass show ?g = 1 by auto
  qed
  moreover from IH ass have
    a*B = (prod ps)^n ∧ gcd a B=1 ⇒ ∃ k. a = k^n by auto
  ultimately have ∃ k. a = k^n by auto }
ultimately show ∃ k. a = k^n by auto
qed
moreover from abcn relprime have gcd a b=1 ∧ a*b=c^n by simp
ultimately show ?thesis by auto
qed
qed

```

1.4 Powers of integers

Now turn to the case of integers. This lemma is based on its equivalent for the natural numbers.

corollary *int-relprime-power-divisors*:

assumes $abcn: a*b = c^n$ **and** $n: n > 1$ **and** $relprime: zgcd\ a\ b = 1$

shows $\exists k. |a| = k^n$

proof –

let $?a1 = nat|a|$

let $?b1 = nat|b|$

let $?c1 = nat|c|$

from $relprime$ **have** $absrelprime: gcd\ ?a1\ ?b1 = 1$ **by** (*auto simp only: zgcd-def*)

have $|a*b| = |a|*|b|$ **by** (*simp add: abs-mult*)

with $abcn$ **have** $|c|^n = |a|*|b|$ **by** (*simp add: power-abs*)

hence $int(?c1^n) = int(?a1*?b1)$ **by** (*simp only: int-nat-abs-eq-abs int-mult int-power*)

hence $?a1*?b1 = ?c1^n$ **by** (*simp only: int-int-eq*)

with $absrelprime$ **and** n **have** $\exists k. ?a1 = k^n$ **by** (*simp only: nat-relprime-power-divisors*)

then obtain $k::nat$ **where** $alpha: ?a1 = k^n$ **by** *auto*

moreover have $int\ ?a1 = |a|$ **by** (*simp add: int-nat-eq*)

ultimately have $|a| = int(k^n)$ **by** *simp*

hence $|a| = int(k)^n$ **by** (*simp only: int-power*)

thus $?thesis$ **by** *auto*

qed

corollary *int-triple-relprime-power-divisors*:

$\llbracket a*b*c = d^n; n > 1; zgcd\ a\ b = 1; zgcd\ b\ c = 1; zgcd\ c\ a = 1 \rrbracket$

$\implies \exists k\ l\ m. |a| = k^n \wedge |b| = l^n \wedge |c| = m^n$

proof –

assume $abcd: a*b*c = d^n$ **and** $n1: n > 1$

and $ab: zgcd\ a\ b = 1$ **and** $bc: zgcd\ b\ c = 1$ **and** $ca: zgcd\ c\ a = 1$

hence $ba: zgcd\ b\ a = 1$ **and** $cb: zgcd\ c\ b = 1$ **and** $ac: zgcd\ a\ c = 1$

by (*auto simp only: zgcd-commute*)

from $ba\ ca$ **have** $zgcd\ (b*c)\ a = 1$ **by** (*simp only: zgcd-zmult-cancel*)

with $abcd$ **have** $a*(b*c) = d^n \wedge zgcd\ a\ (b*c) = 1$ **by** (*simp add: zgcd-commute*)

with $n1$ **have** $k: \exists k. |a| = k^n$ **by** (*auto dest: int-relprime-power-divisors*)

from $ab\ cb$ **have** $zgcd\ (a*c)\ b = 1$ **by** (*simp only: zgcd-zmult-cancel*)

with $abcd$ **have** $b*(a*c) = d^n \wedge zgcd\ b\ (a*c) = 1$

by (*simp add: zgcd-commute mult-ac*)

with $n1$ **have** $l: \exists l. |b| = l^n$ **by** (*auto dest: int-relprime-power-divisors*)

from $ac\ bc$ **have** $zgcd\ (a*b)\ c = 1$ **by** (*simp only: zgcd-zmult-cancel*)

with $abcd$ **have** $c*(a*b) = d^n \wedge zgcd\ c\ (a*b) = 1$

by (*simp add: zgcd-commute mult-ac*)

with $n1$ **have** $m: \exists m. |c| = m^n$ **by** (*auto dest: int-relprime-power-divisors*)

from $k\ l\ m$ **show** $?thesis$ **by** *auto*

qed

lemma *neg-odd-power*: $\llbracket x \in zOdd; x \geq 0 \rrbracket \implies (-a::int)^{(nat\ x)} = -(a^{(nat\ x)})$

proof –

assume $x \in zOdd$ **and** $0 \leq x$

hence $-(a^{(nat\ x)}) = (-1)^{(nat\ x)} * a^{(nat\ x)}$ **by** (*simp add: neg-one-odd-power*)

also have $\dots = (-1*a)^{(nat\ x)}$ **by** (*simp only: power-mult-distrib*)

finally show $?thesis$ **by** *simp*

qed

lemma *neg-even-power*: $\llbracket x \in zEven; x \geq 0 \rrbracket \implies (-a::int)^{(nat\ x)} = a^{(nat\ x)}$

proof –

assume $x \in zEven$ **and** $x \geq 0$

hence $a^{(\text{nat } x)} = (-1)^{(\text{nat } x)} * a^{(\text{nat } x)}$ by (*simp add: neg-one-even-power*)
 also have $\dots = (-1 * a)^{(\text{nat } x)}$ by (*simp only: power-mult-distrib*)
 finally show *?thesis* by *simp*

qed

corollary *int-relprime-odd-power-divisors*:

$\llbracket a * b = c^{(\text{nat } x)}; \text{nat } x > 1; x \in z\text{Odd}; z\text{gcd } a \ b = 1 \rrbracket \implies \exists k. a = k^{(\text{nat } x)}$

proof –

assume $a * b = c^{(\text{nat } x)}$ and $x1: \text{nat } x > 1$ and *odd*: $x \in z\text{Odd}$ and $z\text{gcd } a \ b = 1$

hence $\exists k. |a| = k^{(\text{nat } x)}$ by (*simp only: int-relprime-power-divisors*)

then obtain k where $k: |a| = k^{(\text{nat } x)}$ by *blast*

{ assume $a \neq k^{(\text{nat } x)}$

 with k have $a = -(k^{(\text{nat } x)})$ by *arith*

 with $x1$ *odd* have $a = (-k)^{(\text{nat } x)}$ by (*simp add: neg-odd-power*) }

thus *?thesis* by *blast*

qed

corollary *int-triple-relprime-odd-power-divisors*:

$\llbracket a * b * c = d^{(\text{nat } x)}; \text{nat } x > 1; x \in z\text{Odd}; z\text{gcd } a \ b = 1; z\text{gcd } b \ c = 1; z\text{gcd } c \ a = 1 \rrbracket$
 $\implies \exists k \ l \ m. a = k^{(\text{nat } x)} \wedge b = l^{(\text{nat } x)} \wedge c = m^{(\text{nat } x)}$

proof –

assume *abcd*: $a * b * c = d^{(\text{nat } x)}$ and $x1: \text{nat } x > 1$ and *odd*: $x \in z\text{Odd}$

 and *ab*: $z\text{gcd } a \ b = 1$ and *bc*: $z\text{gcd } b \ c = 1$ and *ca*: $z\text{gcd } c \ a = 1$

hence *ba*: $z\text{gcd } b \ a = 1$ and *cb*: $z\text{gcd } c \ b = 1$ and *ac*: $z\text{gcd } a \ c = 1$

 by (*auto simp only: zgcd-commute*)

{ from *ba ca* have $z\text{gcd } (b * c) \ a = 1$ by (*simp only: zgcd-zmult-cancel*)

 with *abcd* have $a * (b * c) = d^{(\text{nat } x)} \wedge z\text{gcd } a \ (b * c) = 1$

 by (*simp add: zgcd-commute*)

 with $x1$ *odd* have $\exists k. a = k^{(\text{nat } x)}$

 by (*auto dest: int-relprime-odd-power-divisors*) }

moreover

{ from *ab cb* have $z\text{gcd } (a * c) \ b = 1$ by (*simp only: zgcd-zmult-cancel*)

 with *abcd* have $b * (a * c) = d^{(\text{nat } x)} \wedge z\text{gcd } b \ (a * c) = 1$

 by (*simp add: zgcd-commute mult-ac*)

 with $x1$ *odd* have $\exists l. b = l^{(\text{nat } x)}$

 by (*auto dest: int-relprime-odd-power-divisors*) }

moreover

{ from *ac bc* have $z\text{gcd } (a * b) \ c = 1$ by (*simp only: zgcd-zmult-cancel*)

 with *abcd* have $c * (a * b) = d^{(\text{nat } x)} \wedge z\text{gcd } c \ (a * b) = 1$

 by (*simp add: zgcd-commute mult-ac*)

 with $x1$ *odd* have $m: \exists m. c = m^{(\text{nat } x)}$

 by (*auto dest: int-relprime-odd-power-divisors*) }

ultimately show *?thesis* by *auto*

qed

lemma *zgcd-1-power-left-distrib*: $z\text{gcd } a \ b = 1 \implies z\text{gcd } (a^n) \ b = 1$

 by (*induct n, auto simp add: zgcd-zmult-cancel*)

lemma *zgcd-1-power-distrib*: $z\text{gcd } a \ b = 1 \implies z\text{gcd } (a^n) \ (b^n) = 1$

proof –

 assume $z\text{gcd } a \ b = 1$

 hence $z\text{gcd } (a^n) \ b = 1$ by (*rule zgcd-1-power-left-distrib*)

hence $\text{zgcd } b (a^n) = 1$ **by** (*simp only: zgcd-commute*)
hence $\text{zgcd } (b^n) (a^n) = 1$ **by** (*rule zgcd-1-power-left-distrib*)
thus $\text{zgcd } (a^n) (b^n) = 1$ **by** (*simp only: zgcd-commute*)
qed

lemma *zgcd-power-distrib*: $\text{zgcd } a b^n = \text{zgcd } (a^n) (b^n)$

proof *cases*

assume $a=0 \wedge b=0$

thus *?thesis* **by** (*auto simp add: power-0-left*)

next

let $?g = \text{zgcd } a b$

assume $\neg (a=0 \wedge b=0)$

hence $ab \neq 0 \vee b \neq 0$ **by** *simp*

hence $\text{non0: zgcd } a b \neq 0 \text{ zgcd } (a^n) (b^n) \neq 0$

by (*auto simp add: zgcd-def gcd-zero power-eq-0-iff*)

moreover **have** $\text{zgcd } a b \geq 0 \text{ zgcd } (a^n) (b^n) \geq 0$ **by** (*simp-all add: zgcd-geq-zero*)

ultimately **have** $\text{zgcd } a b^n > 0 \text{ zgcd } (a^n) (b^n) > 0$

unfolding *less-le* **by** *simp-all*

moreover **from** $ab \neq 0$ **obtain** $c d$ **where** $abcd: a = ?g*c \wedge b = ?g*d \wedge \text{zgcd } c d = 1$

by (*frule-tac a=a in make-zrelprime, auto*)

moreover **have** $(?g*c)^n = ?g^n * c^n \wedge (?g*d)^n = ?g^n * d^n$

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

ultimately **have** $gabcdn: \text{zgcd } (a^n) (b^n) = ?g^n * \text{zgcd } (c^n) (d^n)$

by (*auto simp add: zgcd-zmult-distrib2*)

moreover **from** $abcd$ **have** $\text{zgcd } (c^n) (d^n) = 1$ **by** (*simp only: zgcd-1-power-distrib*)

ultimately **show** *?thesis* **by** *auto*

qed

lemma *zprime-zdvd-zmult-general*: $\llbracket \text{zprime } p; p \text{ dvd } m*n \rrbracket \implies p \text{ dvd } m \vee p \text{ dvd } n$

apply (*case-tac m ≥ 0, simp only: zprime-zdvd-zmult*)

apply (*subgoal-tac -m ≥ 0 ∧ p dvd (-m)*n, subgoal-tac p dvd -m ∨ p dvd n*)

prefer 2

apply (*rule-tac m=-m in zprime-zdvd-zmult, auto*)

done

lemma *zprime-zdvd-power*: $\llbracket \text{zprime } p; p \text{ dvd } a^n \rrbracket \implies p \text{ dvd } a$

apply (*induct n, auto*)

prefer 2

apply (*frule-tac m=a and n=a^n in zprime-zdvd-zmult-general*)

apply (*auto, simp add: zprime-def zdvd-not-zless*)

done

lemma *zpower-zdvd-mono*: $n \neq 0 \implies (a^n \text{ dvd } b^n) = (a \text{ dvd } (b::\text{int}))$

proof

assume $n \neq 0$

then **obtain** m **where** $mn: n = \text{Suc } m$

by (*frule-tac n=n in not0-implies-Suc, auto*)

assume $a^n \text{ dvd } b^n$

then **obtain** k **where** $k: b^n = a^n * k$ **by** (*auto simp add: dvd-def*)

moreover **have** $\text{zgcd } (a^n*1) (a^n*k) = |a^n| * \text{zgcd } 1 k$

by (*rule-tac k=a^n in zgcd-zmult-distrib2-abs*)

ultimately **have** $\text{zgcd } (a^n) (b^n) = |a^n|$

```

  by (auto simp add: zgcd-commute zgcd-1)
  hence zgcd a b ^ n = |a| ^ n ∧ zgcd a b ≥ 0 ∧ |a| ≥ 0
  by (simp add: zgcd-power-distrib power-abs zgcd-geq-zero)
  with mn have |a| = zgcd a b by (auto intro: power-inject-base [of - m])
  moreover have zgcd a b dvd b by (rule zgcd-zdvd2 [of a])
  ultimately have |a| dvd b by simp
  thus a dvd b by simp
next
  assume a dvd b
  then obtain k where k: b = a * k by (auto simp add: dvd-def)
  hence b ^ n = a ^ n * k ^ n by (simp only: power-mult-distrib)
  thus a ^ n dvd b ^ n by auto
qed

```

lemma *zprime-power-zdvd-cancel-right*:

```

[[ zprime p; ¬ p dvd b; p ^ n dvd a * b ]] ⇒ p ^ n dvd a
proof -
  assume p: zprime p and pb: ¬ p dvd b
  hence p1: p > 1 by (simp add: zprime-def)
  have !!a. p ^ n dvd a * b ⇒ p ^ n dvd a
  proof (induct n)
    case 0 thus ?case by auto
  next
    case (Suc n) hence IH: !!a. p ^ n dvd a * b ⇒ p ^ n dvd a .
    fix a show p ^ Suc n dvd a * b ⇒ p ^ Suc n dvd a
    proof (auto)
      assume ppnab: p * p ^ n dvd a * b
      hence p dvd a * b by (auto simp add: dvd-def mult-assoc)
      with p have p dvd a ∨ p dvd b by (rule zprime-zdvd-zmult-general)
      with pb have p dvd a by simp
      then obtain k where apk: a = p * k by (auto simp add: dvd-def)
      with ppnab have p * p ^ n dvd p * (k * b) by (auto simp add: mult-ac)
      with p1 have p ^ n dvd k * b by (auto dest: zdvd-mult-cancel)
      with IH have p ^ n dvd k ..
      with apk show p * p ^ n dvd a by (simp add: mult-dvd-mono)
    qed
  qed
  thus p ^ n dvd a * b ⇒ p ^ n dvd a ..
qed

```

lemma *zprime-power-zdvd-cancel-left*:

```

[[ zprime p; ¬ p dvd a; p ^ n dvd a * b ]] ⇒ p ^ n dvd b
apply (subgoal-tac p ^ n dvd b * a)
apply (auto dest: zprime-power-zdvd-cancel-right)
apply (simp add: mult-ac)
done

```

1.5 Facts about small powers of integers

```

lemma zadd-power2: ((a::int)+b)^2 = a^2 + 2*a*b + b^2
  by (simp add: nat-number ring-simps)

```

lemma *zdiff-power2*: $((a::int)-b)^2 = a^2 - 2*a*b + b^2$
by (*simp add: nat-number ring-simps*)

lemma *zspecial-product*: $((a::int) + b) * (a - b) = a^2 - b^2$
by (*simp add: nat-number ring-simps*)

lemma *abs-power2-distrib*: $|a^2| = |a::int|^2$
by (*simp add: power2-eq-square abs-mult*)

lemma *power2-eq-iff-abs-eq*: $((a::int)^2 = b^2) = (|a| = |b|)$

proof

assume $a^2 = b^2$
hence $(a+b)*(a-b) = 0$ **by** (*simp add: zspecial-product*)
hence $a=b \vee a=-b$ **by** *auto*
thus $|a| = |b|$ **by** *auto*

next

assume $|a| = |b|$
hence $|a|^2 = |b|^2$ **by** *simp*
thus $a^2 = b^2$ **by** (*simp only: power2-abs*)

qed

lemma *power2-eq1-iff*: $(a::int)^2 = 1 \implies |a|=1$
by (*auto simp add: zmult-eq-1-iff power2-eq-square abs-mult*)

lemma *zadd-power3*: $((a::int)+b)^3 = a^3 + 3*a^2*b + 3*a*b^2 + b^3$
by (*simp add: nat-number ring-simps*)

lemma *zdiff-power3*: $((a::int)-b)^3 = a^3 - 3*a^2*b + 3*a*b^2 - b^3$
by (*simp add: nat-number ring-simps*)

lemma *power3-minus*: $(-a::int)^3 = -(a^3)$

proof -

have $(3::int) \in zOdd \wedge (3::int) \geq 0$ **by** (*unfold zOdd-def, auto*)
hence $(-a)^{(nat 3)} = -(a^{(nat 3)})$ **by** (*simp only: neg-odd-power*)
thus *?thesis* **by** *simp*

qed

lemma *abs-power3-distrib*: $|(x::int)^3| = |x|^3$
by (*simp add: nat-number ring-simps abs-mult*)

lemma *cube-square*: $(a::int)*a^2 = a^3$
by (*simp add: nat-number ring-simps*)

lemma *quartic-square-square*: $(x^2)^2 = (x::int)^4$
by (*simp add: nat-number ring-simps*)

lemma *power2-ge-self*: $x^2 \geq (x::int)$

proof (*cases*)

assume *nonpos*: $x \leq 0$
have $0 \leq x^2$ **by** (*rule zero-le-power2*)
with *nonpos* **show** *?thesis* **by** (*rule zle-trans*)

next

```

assume  $\neg x \leq 0$  hence  $x1: x \geq 1$  by simp
thus ?thesis
proof (cases)
  assume  $x = 1$ 
  thus ?thesis by simp
next
  assume  $\neg x = 1$  with  $x1$  have  $x2: 1 < x$  by simp
  hence  $0 < x$  by simp
  with  $x2$  have  $x*1 < x*x$  by (rule zmult-zless-mono2)
  thus ?thesis by (simp only: power2-eq-square)
qed
qed
end

```

2 Pythagorean triples and Fermat's last theorem, case $n = 4$

```

theory Fermat4
imports IntNatAux Parity
begin

```

Proof of Fermat's last theorem for the case $n = 4$:

$$\forall x, y, z : x^4 + y^4 = z^4 \implies xyz = 0.$$

```

lemma even-eq-two-dvd: even (r::nat) = (2 dvd r) by presburger

```

```

lemma nat-power2-add: ((a::nat)+b)^2 = a^2 + b^2 + 2*a*b by algebra

```

```

lemma nat-power2-diff: a ≥ (b::nat) ⟹ (a-b)^2 = a^2 + b^2 - 2*a*b

```

```

proof -

```

```

  assume a-ge-b: a ≥ b

```

```

  hence a2-ge-b2: a^2 ≥ b^2 by (simp only: power-mono)

```

```

  from a-ge-b have ab-ge-b2: a*b ≥ b^2 by (simp add: power2-eq-square)

```

```

  have b*(a-b) + (a-b)^2 = a*(a-b) by (simp add: power2-eq-square diff-mult-distrib)

```

```

  also have ... = a*b + a^2 + (b^2 - b^2) - 2*a*b

```

```

    by (simp add: diff-mult-distrib2 power2-eq-square)

```

```

  also with a2-ge-b2 have ... = a*b + (a^2 - b^2) + b^2 - 2*a*b by simp

```

```

  also with ab-ge-b2 have ... = (a*b - b^2) + a^2 + b^2 - 2*a*b by auto

```

```

  also have ... = b*(a-b) + a^2 + b^2 - 2*a*b

```

```

    by (simp only: diff-mult-distrib2 power2-eq-square mult-commute)

```

```

  finally show ?thesis by arith

```

```

qed

```

```

lemma nat-power-le-imp-le-base: [ n ≠ 0; a^n ≤ b^n ] ⟹ (a::nat) ≤ b

```

```

proof -

```

```

  assume n ≠ 0 and ab: a^n ≤ b^n

```

```

  then obtain m where n = Suc m by (frule-tac n=n in not0-implies-Suc, auto)

```

```

  with ab have a ≥ 0 and a^Suc m ≤ b^Suc m and b ≥ 0 by auto

```

```

  thus ?thesis by (rule-tac n=m in power-le-imp-le-base)

```

qed

lemma *nat-power-inject-base*: $\llbracket n \neq 0; a^n = b^n \rrbracket \implies (a::nat) = b$

proof –

assume $n \neq 0$ **and** $ab: a^n = b^n$

then obtain m **where** $n = \text{Suc } m$ **by** (*frule-tac n=n in not0-implies-Suc, auto*)

with ab **have** $a^{\text{Suc } m} = b^{\text{Suc } m}$ **and** $a \geq 0$ **and** $b \geq 0$ **by** *auto*

thus *?thesis* **by** (*rule power-inject-base*)

qed

2.1 Parametrisation of Pythagorean triples (over \mathbb{N} and \mathbb{Z})

theorem *nat-euclid-pyth-triples*:

assumes $abc: a^2 + b^2 = c^2$ **and** $ab\text{-relprime}: \text{gcd } a \ b = 1$ **and** $a\text{odd}: \text{odd } a$
 shows $\exists p \ q. a = p^2 - q^2 \wedge b = 2*p*q \wedge c = p^2 + q^2 \wedge \text{gcd } p \ q = 1$

proof –

have $two0: (2::nat) \neq 0$ **by** *simp*

from abc **have** $a2cb: a^2 = c^2 - b^2$ **by** *arith*

 — factor a^2 in coprime factors $(c - b)$ and $(c + b)$; hence both are squares

have $a2factor: a^2 = (c-b)*(c+b)$

proof –

have $c*b - c*b = 0$ **by** *simp*

with $a2cb$ **have** $a^2 = c*c + c*b - c*b - b*b$ **by** (*simp add: power2-eq-square*)

also have $\dots = c*(c+b) - b*(c+b)$

by (*simp add: add-mult-distrib2 add-mult-distrib mult-commute*)

finally show *?thesis* **by** (*simp only: diff-mult-distrib*)

 qed

have $a\text{-nonzero}: a \neq 0$

proof (*rule ccontr*)

assume $\neg a \neq 0$ **hence** $a = 0$ **by** *simp*

with $a\text{odd}$ **have** $\text{odd } (0::nat)$ **by** *simp*

thus *False* **by** *simp*

 qed

have $b\text{-less-c}: b < c$

proof –

from abc **have** $b^2 \leq c^2$ **by** *auto*

with $two0$ **have** $b \leq c$ **by** (*rule-tac n=2 in nat-power-le-imp-le-base*)

moreover have $b \neq c$

proof

assume $b=c$ **with** $a2cb$ **have** $a^2 = 0$ **by** *simp*

with $a\text{-nonzero}$ **show** *False* **by** (*simp add: power2-eq-square*)

 qed

ultimately show *?thesis* **by** *auto*

qed

hence $b2\text{-le-c2}: b^2 \leq c^2$ **by** (*simp add: power-mono*)

have $bc\text{-relprime}: \text{gcd } b \ c = 1$

proof –

from $b2\text{-le-c2}$ **have** $\text{cancelb2}: c^2 - b^2 + b^2 = c^2$ **by** *auto*

let $?g = \text{gcd } b \ c$

have $?g^2 = \text{gcd } (b^2) \ (c^2)$ **by** (*simp only: gcd-power-distrib*)

with cancelb2 **have** $?g^2 = \text{gcd } (b^2) \ (c^2 - b^2 + b^2)$ **by** *simp*

hence $?g^2 = \text{gcd } (b^2) \ (c^2 - b^2)$ **by** *simp*

```

with a2cb have ?g^2 dvd a^2 by (simp only: gcd-dvd2)
hence ?g dvd a ∧ ?g dvd b by (simp add: nat-power-dvd-mono gcd-dvd1)
hence ?g dvd gcd a b by (simp only: gcd-greatest)
with ab-relprime show ?thesis by auto
qed
have p2: prime 2 by (rule two-is-prime)
have factors-odd: odd (c-b) ∧ odd (c+b)
proof (auto simp only: ccontr)
  assume even (c-b) hence 2 dvd c-b by (simp only: even-eq-two-dvd)
  with a2factor have 2 dvd a^2 by (simp only: dvd-mult2)
  with p2 have 2 dvd a by (rule prime-dvd-power)
  hence even a by (simp only: even-eq-two-dvd)
  with aodd show False by simp
next
  assume even (c+b) hence 2 dvd c+b by (simp only: even-eq-two-dvd)
  with a2factor have 2 dvd a^2 by (simp only: dvd-mult)
  with p2 have 2 dvd a by (rule prime-dvd-power)
  hence even a by (simp only: even-eq-two-dvd)
  with aodd show False by simp
qed
have cb1: c-b + (c+b) = 2*c
proof -
  have c-b + (c+b) = ((c-b)+b)+c by simp
  also with b-less-c have ... = (c+b-b)+c by (simp only: diff-add-assoc2)
  also have ... = c+c by simp
  finally show ?thesis by simp
qed
have cb2: 2*b + (c-b) = c+b
proof -
  have 2*b + (c-b) = b+b + (c - b) by auto
  also have ... = b + ((c-b)+b) by simp
  also with b-less-c have ... = b + (c+b-b) by (simp only: diff-add-assoc2)
  finally show ?thesis by simp
qed
have factors-relprime: gcd (c-b) (c+b) = 1
proof -
  let ?g = gcd (c-b) (c+b)
  have cb1: c-b + (c+b) = 2*c
  proof -
    have c-b + (c+b) = ((c-b)+b)+c by simp
    also with b-less-c have ... = (c+b-b)+c by (simp only: diff-add-assoc2)
    also have ... = c+c by simp
    finally show ?thesis by simp
  qed
  have ?g = gcd (c-b + (c+b)) (c+b) by simp
  with cb1 have ?g = gcd (2*c) (c+b) by (rule-tac a=c-b + (c+b) in back-subst)
  hence g2c: ?g dvd 2*c by (simp only: gcd-dvd1)
  have gcd (c-b) (2*b + (c-b)) = gcd (c-b) (2*b) by simp
  with cb2 have ?g = gcd (c-b) (2*b) by (rule-tac a=2*b + (c-b) in back-subst)
  hence g2b: ?g dvd 2*b by (simp only: gcd-dvd2)
  with g2c have ?g dvd 2 * gcd b c by (simp only: gcd-greatest gcd-mult-distrib2)
  with bc-relprime have ?g dvd 2 by simp

```

```

with p2 have g1or2: ?g = 2 ∨ ?g = 1 by (unfold prime-def, auto)
thus ?thesis
proof (auto)
  assume ?g = 2 hence 2 dvd ?g by simp
  hence 2 dvd c-b by (simp add: gcd-dvd1)
  with factors-odd show False by (simp add: even-eq-two-dvd)
qed
qed
from a2factor have (c-b)*(c+b) = a^2 and (2::nat) >1 by auto
with factors-relprime have ∃ k. c-b = k^2 by (simp only: nat-relprime-power-divisors)
then obtain r where r: c-b = r^2 by auto
from a2factor have (c+b)*(c-b) = a^2 and (2::nat) >1 by auto
with factors-relprime have ∃ k. c+b = k^2
  by (simp only: nat-relprime-power-divisors gcd-commute)
then obtain s where s: c+b = s^2 by auto
— now p := (s+r)/2 and q := (s-r)/2 is our solution
have rs-odd: odd r ∧ odd s
proof (auto dest: ccontr)
  assume even r hence 2 dvd r by presburger
  with r have 2 dvd (c-b) by (simp only: power2-eq-square dvd-mult)
  hence even (c-b) by (simp only: even-eq-two-dvd)
  with factors-odd show False by auto
next
  assume even s hence 2 dvd s by presburger
  with s have 2 dvd (c+b) by (simp only: power2-eq-square dvd-mult)
  hence even (c+b) by (simp only: even-eq-two-dvd)
  with factors-odd show False by auto
qed
obtain m where m: m = s-r by simp
from r s have r^2 ≤ s^2 by arith
with two0 have r ≤ s by (rule-tac n=2 in nat-power-le-imp-le-base)
with m have m2: s = r + m by simp
have even m
proof (rule ccontr)
  assume odd m with rs-odd and m2 show False by presburger
qed
hence 2 dvd m by (simp only: even-eq-two-dvd)
then obtain q where m = 2*q by (auto simp add: dvd-def)
with m2 have q: s = r + 2*q by simp
obtain p where p: p = r+q by simp
have c: c = p^2 + q^2
proof -
  from cb1 and r and s have 2*c = r^2 + s^2 by simp
  also with q have ... = 2*r^2 + (2*q)^2 + 2*r*(2*q)
    by (simp add: nat-power2-add)
  also have ... = 2*r^2 + 2^2*q^2 + 2*2*q*r by (simp add: power-mult-distrib)
  also have ... = 2*(r^2 + 2*q*r + q^2) + 2*q^2 by (simp add: power2-eq-square)
  also with p have ... = 2*p^2 + 2*q^2 by (simp add: nat-power2-add)
  finally show ?thesis by auto
qed
moreover have b: b = 2*p*q
proof -

```

```

from cb2 and r and s have  $2*b = s^2 - r^2$  by arith
also with q have  $\dots = (2*q)^2 + 2*r*(2*q)$  by (simp add: nat-power2-add)
also with p have  $\dots = 4*q*p$  by (simp add: power2-eq-square add-mult-distrib2)
finally show ?thesis by auto
qed
moreover have a:  $a = p^2 - q^2$ 
proof -
  from p have  $p \geq q$  by simp
  hence p2-ge-q2:  $p^2 \geq q^2$  by (simp only: power-mono)
  from a2cb and b and c have  $a^2 = (p^2 + q^2)^2 - (2*p*q)^2$  by simp
  also have  $\dots = (p^2)^2 + (q^2)^2 - 2*(p^2)*(q^2)$ 
    by (auto simp add: nat-power2-add power-mult-distrib mult-ac)
  also with p2-ge-q2 have  $\dots = (p^2 - q^2)^2$  by (simp only: nat-power2-diff)
  finally have  $a^2 = (p^2 - q^2)^2$  by simp
  with two0 show ?thesis by (rule-tac n=2 in nat-power-inject-base)
qed
moreover have gcd p q=1
proof -
  let ?k = gcd p q
  have ?k dvd p  $\wedge$  ?k dvd q by (simp add: gcd-dvd1 gcd-dvd2)
  with b and a have ?k dvd a  $\wedge$  ?k dvd b
    by (simp add: dvd-mult power2-eq-square dvd-diff)
  hence ?k dvd gcd a b by (simp only: gcd-greatest)
  with ab-relprime show ?thesis by auto
qed
ultimately show ?thesis by auto
qed

```

Now for the case of integers. Based on *nat-euclid-pyth-triples*.

```

corollary int-euclid-pyth-triples:  $\llbracket zgcd\ a\ b = 1; a \in zOdd; a^2 + b^2 = c^2 \rrbracket$ 
 $\implies \exists\ p\ q. a = p^2 - q^2 \wedge b = 2*p*q \wedge |c| = p^2 + q^2 \wedge zgcd\ p\ q = 1$ 
proof -
  assume ab-rel:  $zgcd\ a\ b = 1$  and aodd:  $a \in zOdd$  and abc:  $a^2 + b^2 = c^2$ 
  let ?a =  $nat|a|$ 
  let ?b =  $nat|b|$ 
  let ?c =  $nat|c|$ 
  have ab2-pos:  $a^2 \geq 0 \wedge b^2 \geq 0$  by (simp add: zero-le-power2)
  hence  $nat(a^2) + nat(b^2) = nat(a^2 + b^2)$  by (simp only: nat-add-distrib)
  with abc have  $nat(a^2) + nat(b^2) = nat(c^2)$ 
    by (auto simp add: power2-eq-square abs-power2-distrib)
  hence  $nat(|a|^2) + nat(|b|^2) = nat(|c|^2)$ 
    by (simp add: abs-power2-distrib)
  hence new-abc:  $?a^2 + ?b^2 = ?c^2$ 
    by (simp only: nat-mult-distrib power2-eq-square nat-add-distrib)
  moreover from ab-rel have new-ab-rel:  $gcd\ ?a\ ?b = 1$  by (simp add: zgcd-def)
  moreover have new-a-odd: odd ?a
proof -
  from aodd obtain k where k:  $a = 2*k+1$  by (unfold zOdd-def, auto)
  show ?thesis
proof (cases)
  assume apos:  $a \geq 0$  with k have  $k \geq 0$  by auto
  with k and apos have  $?a = 2*(nat\ k)+1$  by arith

```

```

thus ?thesis by simp
next
  assume  $\neg a \geq 0$  hence aneg:  $a < 0$  by simp
  with k have k2:  $2 * (-1 - k) \geq 0$  by simp
  have aux2:  $(2 :: \text{int}) \geq 0$  by simp
  have aux1:  $(1 :: \text{int}) \geq 0$  by simp
  from k and aneg have  $|a| = 2 * (-1 - k) + 1$  by simp
  with k2 aux1 have ?a = nat  $(2 * (-1 - k)) + \text{nat } 1$ 
    by (simp only: nat-add-distrib)
  with aux2 have ?a = (nat 2) * nat  $(-1 - k) + \text{nat } 1$ 
    by (simp only: nat-mult-distrib)
  thus ?thesis by simp
qed
ultimately have
   $\exists p q. ?a = p^2 - q^2 \wedge ?b = 2 * p * q \wedge ?c = p^2 + q^2 \wedge \text{gcd } p \ q = 1$ 
  by (rule-tac a=?a and b=?b and c=?c in nat-euclid-pyth-triples)
then obtain m and n where mn:
   $?a = m^2 - n^2 \wedge ?b = 2 * m * n \wedge ?c = m^2 + n^2 \wedge \text{gcd } m \ n = 1$  by auto
have  $n^2 \leq m^2$ 
proof (rule ccontr)
  assume  $\neg n^2 \leq m^2$  hence  $n^2 > m^2$  by simp
  with mn have ?a = 0 by simp
  with new-a-odd show False by simp
qed
moreover from mn have int ?a = int  $(m^2 - n^2)$  and int ?b = int  $(2 * m * n)$ 
  and int ?c = int  $(m^2 + n^2)$  by auto
ultimately have  $|a| = \text{int}(m^2) - \text{int}(n^2)$  and  $|b| = \text{int}(2 * m * n)$ 
  and  $|c| = \text{int}(m^2) + \text{int}(n^2)$  by (auto simp only: int-nat-abs-eq-abs zdiff-int)
hence absabc:  $|a| = (\text{int } m)^2 - (\text{int } n)^2 \wedge |b| = 2 * (\text{int } m) * \text{int } n$ 
   $\wedge |c| = (\text{int } m)^2 + (\text{int } n)^2$  by (simp add: power2-eq-square int-mult)
from mn have mn-rel: zgcd (int m) (int n) = 1 by (simp add: zgcd-def)
show  $\exists p q. a = p^2 - q^2 \wedge b = 2 * p * q \wedge |c| = p^2 + q^2 \wedge \text{zgcd } p \ q = 1$ 
  (is  $\exists p q. ?Q \ p \ q$ )
proof (cases)
  assume apos:  $a \geq 0$  then obtain p where p:  $p = \text{int } m$  by simp
  hence  $\exists q. ?Q \ p \ q$ 
  proof (cases)
    assume bpos:  $b \geq 0$  then obtain q where q = int n by simp
    with p apos bpos absabc mn-rel have ?Q p q by simp
    thus ?thesis by (rule exI)
  next
    assume  $\neg b \geq 0$  hence bneg:  $b < 0$  by simp
    then obtain q where q = - int n by simp
    with p apos bneg absabc mn-rel have ?Q p q by simp
    thus ?thesis by (rule exI)
  qed
thus ?thesis by (simp only: exI)
next
  assume  $\neg a \geq 0$  hence aneg:  $a < 0$  by simp
  then obtain p where p:  $p = \text{int } n$  by simp
  hence  $\exists q. ?Q \ p \ q$ 

```

proof (*cases*)
assume $bpos: b \geq 0$ **then obtain** q **where** $q = \text{int } m$ **by** *simp*
with $p \text{ aneg } bpos \text{ absabc } mn\text{-rel}$ **have** $?Q \ p \ q$
by (*simp add: zgcd-commute*)
thus $?thesis$ **by** (*rule exI*)
next
assume $\neg b \geq 0$ **hence** $bneg: b < 0$ **by** *simp*
then obtain q **where** $q = - \text{int } m$ **by** *simp*
with $p \text{ aneg } bneg \text{ absabc } mn\text{-rel}$ **have** $?Q \ p \ q$
by (*simp add: zgcd-commute mult-ac*)
thus $?thesis$ **by** (*rule exI*)
qed
thus $?thesis$ **by** (*simp only: exI*)
qed
qed

2.2 Fermat's last theorem, case $n = 4$

Core of the proof. Constructs a smaller solution over \mathbb{Z} of

$$a^4 + b^4 = c^2 \wedge \text{gcd } ab = 1 \wedge abc \neq 0 \wedge a \text{ odd.}$$

lemma *smaller-fermat4*:

assumes $abc: a^4 + b^4 = c^2$ **and** $abc0: a * b * c \neq 0$ **and** $aodd: a \in zOdd$
and $ab\text{-relprime}: \text{zgcd } a \ b = 1$

shows

$\exists \ p \ q \ r. (p^4 + q^4 = r^2 \wedge p * q * r \neq 0 \wedge p \in zOdd \wedge \text{zgcd } p \ q = 1 \wedge r^2 < c^2)$

proof –

— put equation in shape of a pythagorean triple and obtain u and v

from $ab\text{-relprime}$ **have** $a2b2\text{relprime}: \text{zgcd } (a^2) \ (b^2) = 1$

by (*simp only: zgcd-1-power-distrib*)

moreover from $aodd$ **have** $a^2 \in zOdd$ **by** (*simp only: power-preserves-odd*)

moreover from abc **have** $(a^2)^2 + (b^2)^2 = c^2$ **by** (*simp only: quartic-square-square*)

ultimately obtain u **and** v **where** $uabc$:

$a^2 = u^2 - v^2 \wedge b^2 = 2 * u * v \wedge |c| = u^2 + v^2 \wedge \text{zgcd } u \ v = 1$

by (*frule-tac a=a^2 in int-euclid-pyth-triples, auto*)

with $abc0$ **have** $uv0: u \neq 0 \wedge v \neq 0$ **by** *auto*

have $av\text{-relprime}: \text{zgcd } a \ v = 1$

proof –

have $\text{zgcd } a \ v \ \text{dvd} \ \text{zgcd } (a^2) \ v$

by (*simp only: zgcd-zdvd-zgcd-zmult power2-eq-square*)

moreover

from $uabc$ **have** $\text{zgcd } v \ (a^2) \ \text{dvd} \ \text{zgcd } (b^2) \ (a^2)$ **by** (*simp only: zgcd-zdvd-zgcd-zmult*)

with $a2b2\text{relprime}$ **have** $\text{zgcd } (a^2) \ v \ \text{dvd} \ (1::\text{int})$ **by** (*simp only: zgcd-commute*)

ultimately have $\text{zgcd } a \ v \ \text{dvd} \ 1$ **by** (*rule dvd-trans*)

hence $|\text{zgcd } a \ v| = 1$ **by** *auto*

thus $?thesis$ **by** (*simp add: zgcd-geq-zero*)

qed

— make again a pythagorean triple and obtain k and l

from $uabc$ **have** $a^2 + v^2 = u^2$ **by** *simp*

with $av\text{-relprime}$ **and** $aodd$ **obtain** $k \ l$ **where**

$klavu: a = k^2 - l^2 \wedge v = 2 * k * l \wedge |u| = k^2 + l^2$ **and** $kl\text{-rel}: \text{zgcd } k \ l = 1$

by (*frule-tac a=a in int-euclid-pyth-triples, auto*)
 — prove $b = 2m$ and $kl(k^2 + l^2) = m^2$, for coprime k, l and $k^2 + l^2$
 from *uwabc* have $b^2 \in zEven$ by (*unfold zEven-def, auto*)
 hence $b \in zEven$ by (*simp only: power-preserves-even*)
 then obtain m where $bm: b = 2*m$ by (*auto simp only: zEven-def*)
 have $|k|*|l|*|k^2+l^2| = m^2$
 proof —
 from *bm* have $4*m^2 = b^2$ by (*simp only: power2-eq-square mult-ac*)
 also have $\dots = |b^2|$ by *simp*
 also with *uwabc* have $\dots = 2*|v|*|u|$ by (*simp add: abs-mult*)
 also with *klavu* have $\dots = 2*|2*k*l|*|k^2+l^2|$ by *simp*
 also have $\dots = 4*|k|*|l|*|k^2+l^2|$ by (*auto simp add: abs-mult*)
 finally show *?thesis* by *simp*
 qed
 moreover have $(2::nat) > 1$ by *auto*
 moreover from *kl-rel* have $zgcd |k| |l| = 1$ by (*unfold zgcd-def, auto*)
 moreover have $zgcd |l| (|k^2+l^2|) = 1$
 proof —
 from *kl-rel* have $zgcd (k*k) l = 1$ by (*simp only: zgcd-zgcd-zmult*)
 hence $zgcd (k*k+l*l) l = 1$ by *simp*
 hence $zgcd l (k^2+l^2)=1$ by (*simp only: power2-eq-square zgcd-commute*)
 thus *?thesis* by (*unfold zgcd-def, auto*)
 qed
 moreover have $zgcd |k^2+l^2| |k|=1$
 proof —
 from *kl-rel* have $zgcd l k = 1$ by (*simp only: zgcd-commute*)
 hence $zgcd (l*l) k = 1$ by (*simp only: zgcd-zgcd-zmult*)
 hence $zgcd (l*l+k*k) k = 1$ by *simp*
 hence $zgcd (k^2+l^2) k = 1$ by (*simp only: add-ac power2-eq-square*)
 thus *?thesis* by (*unfold zgcd-def, auto*)
 qed
 ultimately have
 $\exists x y z. ||k|| = x^2 \wedge ||l|| = y^2 \wedge ||k^2+l^2|| = z^2$
 by (*rule int-triple-relprime-power-divisors*)
 then obtain $\alpha \beta \gamma$ where *albega*:
 $|k| = \alpha^2 \wedge |l| = \beta^2 \wedge |k^2+l^2| = \gamma^2$
 by *auto*
 — show this is a new solution
 have $k^2 = \alpha^4$
 proof —
 from *albega* have $|k|^2 = (\alpha^2)^2$ by *simp*
 thus *?thesis* by (*simp add: quartic-square-square abs-power2-distrib*)
 qed
 moreover have $l^2 = \beta^4$
 proof —
 from *albega* have $|l|^2 = (\beta^2)^2$ by *simp*
 thus *?thesis* by (*simp add: quartic-square-square abs-power2-distrib*)
 qed
 moreover have *gamma2*: $k^2 + l^2 = \gamma^2$
 proof —
 have $k^2 \geq 0 \wedge l^2 \geq 0$ by (*simp add: zero-le-power2*)
 with *albega* show *?thesis* by *auto*

```

qed
ultimately have newabc:  $\alpha^4 + \beta^4 = \gamma^2$  by auto
from w0 klavu albega have albega0:  $\alpha * \beta * \gamma \neq 0$  by auto
— show the coprimality
have alphabeta-relprime:  $\text{zgcd } \alpha \beta = 1$ 
proof (rule classical)
  let ?g =  $\text{zgcd } \alpha \beta$ 
  assume gnot1:  $?g \neq 1$ 
  have ?g > 1
  proof –
    have ?g  $\neq 0$ 
    proof
      assume ?g=0
      hence nat | $\alpha$ |=0 by (unfold zgcd-def, auto simp add: gcd-zero)
      hence  $\alpha=0$  by arith
      with albega0 show False by simp
    qed
    hence ?g>0 by (auto simp only: zgcd-geq-zero less-int-def)
    with gnot1 show ?thesis by simp
  qed
moreover have ?g dvd  $\text{zgcd } k l$ 
proof –
  have ?g dvd  $\alpha \wedge ?g \text{ dvd } \beta$  by auto
  with albega have ?g dvd  $|k| \wedge ?g \text{ dvd } |l|$ 
    by (simp add: power2-eq-square zmult-commute)
  hence ?g dvd  $k \wedge ?g \text{ dvd } l$  by simp
  thus ?thesis by (simp add: zgcd-greatest-iff)
qed
ultimately have  $\text{zgcd } k l \neq 1$  by auto
with kl-rel show ?thesis by auto
qed
— choose  $p$  and  $q$  in the right way
have  $\exists p q. p^4 + q^4 = \gamma^2 \wedge p * q * \gamma \neq 0 \wedge p \in \text{zOdd} \wedge \text{zgcd } p q = 1$ 
proof –
  have  $\alpha \in \text{zOdd} \vee \beta \in \text{zOdd}$ 
  proof (rule ccontr)
    assume  $\neg (\alpha \in \text{zOdd} \vee \beta \in \text{zOdd})$ 
    hence  $\alpha \in \text{zEven} \wedge \beta \in \text{zEven}$  by (auto simp add: not-odd-impl-even)
    then have  $2 \text{ dvd } \alpha \wedge 2 \text{ dvd } \beta$  by (auto simp add: zEven-def)
    then have  $2 \text{ dvd } \text{zgcd } \alpha \beta$  by (simp add: zgcd-greatest-iff)
    with alphabeta-relprime show False by auto
  qed
moreover
{ assume  $\alpha \in \text{zOdd}$ 
  with newabc albega0 alphabeta-relprime obtain  $p q$  where
     $p = \alpha \wedge q = \beta \wedge p^4 + q^4 = \gamma^2 \wedge p * q * \gamma \neq 0 \wedge p \in \text{zOdd} \wedge \text{zgcd } p q = 1$ 
    by auto
  hence ?thesis by auto }
moreover
{ assume  $\beta \in \text{zOdd}$ 
  with newabc albega0 alphabeta-relprime obtain  $p q$  where
     $q = \alpha \wedge p = \beta \wedge p^4 + q^4 = \gamma^2 \wedge p * q * \gamma \neq 0 \wedge p \in \text{zOdd} \wedge \text{zgcd } p q = 1$ 

```

```

    by (auto simp add: add-ac zgcd-commute)
    hence ?thesis by auto }
  ultimately show ?thesis by auto
qed
— show the solution is smaller
moreover have  $\gamma^2 < c^2$ 
proof —
  from gamma2 klavu have  $\gamma^2 \leq |u|$  by simp
  also have  $\dots \leq |u|^2$  by (rule power2-ge-self)
  also have  $\dots \leq u^2$  by (simp add: abs-power2-distrib)
  also have  $\dots < u^2 + v^2$ 
  proof —
    from uv0 have v2non0:  $0 \neq v^2$ 
    by (auto simp add: power2-eq-square zero-le-power2)
    have  $0 \leq v^2$  by (rule zero-le-power2)
    with v2non0 have  $0 < v^2$  by (auto simp add: less-int-def)
    thus ?thesis by auto
  qed
  also with uvabc have  $\dots \leq |c|$  by auto
  also have  $\dots \leq |c|^2$  by (rule power2-ge-self)
  also have  $\dots \leq c^2$  by (simp add: abs-power2-distrib)
  finally show ?thesis by simp
qed
ultimately show ?thesis by auto
qed

```

Show that no solution exists, by infinite descent of c^2 .

```

lemma no-rewritten-fermat4:
   $\neg (\exists a b. (a^4 + b^4 = c^2 \wedge a*b*c \neq 0 \wedge a \in zOdd \wedge zgcd a b=1))$ 
proof (induct c rule: infinite-descent0-measure[where V= $\lambda c. nat(c^2)$ ])
  case (0 x)
  have  $x^2 \geq 0$  by (rule zero-le-power2)
  with 0 have  $int(nat(x^2)) = 0$  by auto
  hence  $x = 0$  by auto
  thus ?case by auto
next
  case (smaller x)
  then obtain a b where  $a^4 + b^4 = x^2$  and  $a*b*x \neq 0$ 
    and  $a \in zOdd$  and  $zgcd a b=1$  by auto
  hence  $\exists p q r. (p^4 + q^4 = r^2 \wedge p*q*r \neq 0 \wedge p \in zOdd$ 
     $\wedge zgcd p q=1 \wedge r^2 < x^2)$  by (rule smaller-fermat4)
  then obtain p q r where pqr:  $p^4 + q^4 = r^2 \wedge p*q*r \neq 0 \wedge p \in zOdd$ 
     $\wedge zgcd p q=1 \wedge r^2 < x^2$  by auto
  have  $r^2 \geq 0$  and  $x^2 \geq 0$  by (auto simp only: zero-le-power2)
  hence  $int(nat(r^2)) = r^2 \wedge int(nat(x^2)) = x^2$  by auto
  with pqr have  $int(nat(r^2)) < int(nat(x^2))$  by auto
  hence  $nat(r^2) < nat(x^2)$  by (simp only: zless-int)
  with pqr show ?case by auto
qed

```

The theorem. Puts equation in requested shape.

theorem *fermat4*:

```

assumes ass:  $(x::int)^4 + y^4 = z^4$ 
shows  $x*y*z=0$ 
proof (rule ccontr)
  let  $?g = \text{zgcd } x \ y$ 
  let  $?c = (z \ \text{div } ?g)^2$ 
  assume xyz0:  $x*y*z \neq 0$ 
  — divide out the g.c.d.
  hence  $x \neq 0 \vee y \neq 0$  by simp
  then obtain a b where  $ab$ :  $x = ?g*a \wedge y = ?g*b \wedge \text{zgcd } a \ b=1$ 
    by (frule-tac a=x in make-zrelprime, auto)
  moreover have abc:  $a^4 + b^4 = ?c^2 \wedge a*b*?c \neq 0$ 
  proof —
    have zgab:  $z^4 = ?g^4 * (a^4 + b^4)$ 
    proof —
      from ab ass have  $z^4 = (?g*a)^4 + (?g*b)^4$  by simp
      thus ?thesis by (simp only: power-mult-distrib zadd-zmult-distrib2)
    qed
    have cgz:  $z^2 = ?c * ?g^2$ 
    proof —
      from zgab have  $?g^4 \ \text{dvd } z^4$  by simp
      hence  $?g \ \text{dvd } z$  by (simp only: zpower-zdvd-mono)
      hence  $(z \ \text{div } ?g)*?g = z$  by (simp only: mult-ac zdvd-mult-div-cancel)
      with ab show ?thesis by (auto simp only: power2-eq-square mult-ac)
    qed
    with xyz0 have c0:  $?c \neq 0$  by (auto simp add: power2-eq-square)
    from xyz0 have g0:  $?g \neq 0$  by (simp add: zgcd-def gcd-zero)
    have  $a^4 + b^4 = ?c^2$ 
    proof —
      have  $?c^2 * ?g^4 = (a^4 + b^4)*?g^4$ 
      proof —
        have  $?c^2 * ?g^4 = (?c*?g^2)^2$ 
        by (simp only: quartic-square-square power-mult-distrib)
        also with cgz have  $\dots = (z^2)^2$  by simp
        also have  $\dots = z^4$  by (rule quartic-square-square)
        also with zgab have  $\dots = ?g^4*(a^4 + b^4)$  by simp
        finally show ?thesis by simp
      qed
      with g0 show ?thesis by auto
    qed
    moreover from ab xyz0 c0 have  $a*b*?c \neq 0$  by auto
    ultimately show ?thesis by simp
  qed
  — choose the parity right
  have  $\exists \ p \ q. \ p^4 + q^4 = ?c^2 \wedge p*q*?c \neq 0 \wedge p \in \text{zOdd} \wedge \text{zgcd } p \ q=1$ 
  proof —
    have  $a \in \text{zOdd} \vee b \in \text{zOdd}$ 
    proof (rule ccontr)
      assume  $\neg(a \in \text{zOdd} \vee b \in \text{zOdd})$ 
      hence  $a \in \text{zEven} \wedge b \in \text{zEven}$  by (auto simp add: not-odd-impl-even)
      hence  $2 \ \text{dvd } a \wedge 2 \ \text{dvd } b$  by (auto simp add: zEven-def)
      hence  $2 \ \text{dvd } \text{zgcd } a \ b$  by (simp add: zgcd-greatest-iff)
      with ab show False by auto
    qed

```

```

qed
moreover
{ assume a ∈ zOdd
  then obtain p q where p = a and q = b and p ∈ zOdd by simp
  with ab abc have ?thesis by auto }
moreover
{ assume b ∈ zOdd
  then obtain p q where p = b and q = a and p ∈ zOdd by simp
  with ab abc have
    p^4 + q^4 = ?c^2 ∧ p*q*?c≠0 ∧ p ∈ zOdd ∧ zgcd p q=1
    by (auto simp add: zgcd-commute zmult-commute)
  hence ?thesis by auto }
ultimately show ?thesis by auto
qed
— show contradiction using the earlier result
thus False by (auto simp only: no-rewritten-fermat4)
qed

corollary fermat-mult4:
  assumes xyz: (x::int)^n + y^n = z^n and n: 4 dvd n
  shows x*y*z=0
proof —
  from n obtain m where n = m*4 by (auto simp only: mult-ac dvd-def)
  with xyz have (x^m)^4 + (y^m)^4 = (z^m)^4 by (simp only: power-mult)
  hence (x^m)*(y^m)*(z^m) = 0 by (rule fermat4)
  thus ?thesis by auto
qed

end

```

3 The quadratic form $x^2 + Ny^2$

```

theory QuadForm
imports
  ~~/src/HOL/Number-Theory/Quadratic-Reciprocity
  IntNatAux
begin

```

Shows some properties of the quadratic form $x^2 + Ny^2$, such as how to multiply and divide them. The second part focuses on the case $N = 3$ and is used in the proof of the case $n = 3$ of Fermat's last theorem. The last part – not used for FLT3 – shows which primes can be written as $x^2 + 3y^2$.

3.1 Definitions and auxiliary results

```

definition
  is-qn :: int ⇒ int ⇒ bool where
  is-qn A N ⟷ (∃ x y. A = x^2 + N*y^2)

```

```

definition
  is-cube-form :: int ⇒ int ⇒ bool where

```

is-cube-form $a\ b \longleftrightarrow (\exists\ p\ q.\ a = p^3 - 9*p*q^2 \wedge b = 3*p^2*q - 3*q^3)$

lemma *abs-eq-impl-unitfactor*: $|a::int| = |b| \implies \exists\ u.\ a = u*b \wedge |u|=1$

proof –

assume $|a| = |b|$

hence $a = 1*b \vee a = (-1)*b$ **by** *arith*

then obtain u **where** $a = u*b \wedge (u=1 \vee u=-1)$ **by** *blast*

thus *?thesis* **by** *auto*

qed

lemma *zprime-3*: *zprime 3*

proof (*auto simp add: zprime-def*)

fix $m::int$ **assume** $m0: m \geq 0$ **and** $mdvd3: m\ dvd\ 3$ **and** $mn3: m \neq 3$

hence $m \leq 3$ **by** (*auto simp only: zdvd-imp-le*)

with $mn3$ **have** $m < 3$ **by** *simp*

moreover from $mdvd3$ **have** $m \neq 0$ **by** *auto*

moreover with $m0$ **have** $m > 0$ **by** *simp*

ultimately have $m = 1 \vee m = 2$ **by** *auto*

moreover from $mdvd3$ **have** $m = 2 \implies False$ **by** *arith*

ultimately show $m = 1$ **by** *auto*

qed

3.2 Basic facts if $N \geq 1$

lemma *qfN-pos*: $\llbracket N \geq 1; is-qfN\ A\ N \rrbracket \implies A \geq 0$

proof –

assume $N: N \geq 1$ **and** *is-qfN A N*

then obtain $a\ b$ **where** $ab: A = a^2 + N*b^2$ **by** (*auto simp add: is-qfN-def*)

have $N*b^2 \geq 0$

proof (*cases*)

assume $b = 0$ **thus** *?thesis* **by** *auto*

next

assume $\neg b = 0$ **hence** $b^2 > 0$ **by** (*simp add: zero-less-power2*)

moreover from N **have** $N > 0$ **by** *simp*

ultimately have $N*b^2 > N*0$ **by** (*auto simp only: zmult-zless-mono2*)

thus *?thesis* **by** *auto*

qed

with ab **have** $A \geq a^2$ **by** *auto*

moreover have $a^2 \geq 0$ **by** (*rule zero-le-power2*)

ultimately show *?thesis* **by** *arith*

qed

lemma *qfN-zero*: $\llbracket (N::int) \geq 1; a^2 + N*b^2 = 0 \rrbracket \implies (a = 0 \wedge b = 0)$

proof –

assume $N: N \geq 1$ **and** $abN: a^2 + N*b^2 = 0$

show *?thesis*

proof (*rule ccontr, auto*)

assume $a \neq 0$ **hence** $a^2 > 0$ **by** (*simp add: zero-less-power2*)

moreover have $N*b^2 \geq 0$

proof (*cases*)

assume $b = 0$ **thus** *?thesis* **by** *auto*

next

assume $\neg b = 0$ hence $b^2 > 0$ by (simp add: zero-less-power2)
 moreover from N have $N > 0$ by simp
 ultimately have $N * b^2 > N * 0$ by (auto simp only: zmult-zless-mono2)
 thus ?thesis by auto
 qed
 ultimately have $a^2 + N * b^2 > 0$ by arith
 with abN show *False* by auto
 next
 assume $b \neq 0$ hence $b^2 > 0$ by (simp add: zero-less-power2)
 moreover from N have $N > 0$ by simp
 ultimately have $N * b^2 > N * 0$ by (auto simp only: zmult-zless-mono2)
 hence $N * b^2 > 0$ by simp
 moreover have $a^2 \geq 0$ by (rule zero-le-power2)
 ultimately have $a^2 + N * b^2 > 0$ by arith
 with abN show *False* by auto
 qed
 qed

3.3 Multiplication and division

lemma *qfN-mult1*: $((a::int)^2 + N * b^2) * (c^2 + N * d^2)$
 $= (a * c + N * b * d)^2 + N * (a * d - b * c)^2$
 by (simp add: nat-number ring-simps)

lemma *qfN-mult2*: $((a::int)^2 + N * b^2) * (c^2 + N * d^2)$
 $= (a * c - N * b * d)^2 + N * (a * d + b * c)^2$
 by (simp add: nat-number ring-simps)

corollary *is-qfN-mult*: $is\text{-}qfN\ A\ N \implies is\text{-}qfN\ B\ N \implies is\text{-}qfN\ (A * B)\ N$
 by (unfold is-qfN-def, auto, auto simp only: qfN-mult1)

corollary *is-qfN-power*: $(n::nat) > 0 \implies is\text{-}qfN\ A\ N \implies is\text{-}qfN\ (A^n)\ N$
 by (induct n, auto, case-tac n=0, auto simp add: is-qfN-mult)

lemma *qfN-div-prime*:

assumes *ass*: $zprime\ (p^2 + N * q^2) \wedge (p^2 + N * q^2)\ dvd\ (a^2 + N * b^2)$
 shows $\exists\ u\ v.\ a^2 + N * b^2 = (u^2 + N * v^2) * (p^2 + N * q^2)$
 $\wedge (\exists\ e.\ a = p * u + e * N * q * v \wedge b = p * v - e * q * u \wedge |e| = 1)$

proof –

let $?P = p^2 + N * q^2$

let $?A = a^2 + N * b^2$

from *ass* obtain U where $U: ?A = ?P * U$ by (auto simp only: dvd-def)

have $\exists\ e.\ ?P\ dvd\ b * p + e * a * q \wedge |e| = 1$

proof –

have $?P\ dvd\ (b * p + a * q) * (b * p - a * q)$

proof –

have $(b * p + a * q) * (b * p - a * q) = b^2 * ?P - q^2 * ?A$

by (simp add: nat-number ring-simps)

also from U have $\dots = (b^2 - q^2 * U) * ?P$ by (simp add: ring-simps)

finally show ?thesis by simp

qed

with *ass* have $?P\ dvd\ (b * p + a * q) \vee ?P\ dvd\ (b * p - a * q)$

by (*simp only: zprime-zdvd-zmult-general*)
moreover
 { **assume** $?P \text{ dvd } b*p + a*q$
 hence $?P \text{ dvd } b*p + 1*a*q \wedge |1| = (1::int)$ **by** *simp* }
moreover
 { **assume** $?P \text{ dvd } b*p - a*q$
 hence $?P \text{ dvd } b*p + (-1)*a*q \wedge |-1| = (1::int)$ **by** *simp* }
ultimately show *?thesis* **by** *blast*
qed
then obtain $v \ e$ **where** $v: b*p + e*a*q = ?P*v$ **and** $e: |e| = 1$
 by (*auto simp only: dvd-def*)
have $?P \text{ dvd } a*p - e*N*b*q$
proof (*cases*)
 assume $e1: e = 1$
 from U **have** $U * ?P^2 = ?A * ?P$ **by** (*simp add: power2-eq-square*)
 also with $e1$ **have** $\dots = (a*p - e*N*b*q)^2 + N*(b*p + e*a*q)^2$
 by (*simp only: qfN-mult2 add-commute zmult-1*)
 also with v **have** $\dots = (a*p - e*N*b*q)^2 + N*v^2*?P^2$
 by (*simp only: power-mult-distrib mult-ac*)
 finally have $(a*p - e*N*b*q)^2 = ?P^2*(U - N*v^2)$
 by (*simp add: mult-ac zdiff-zmult-distrib*)
 hence $?P^2 \text{ dvd } (a*p - e*N*b*q)^2$ **by** (*rule dvdI*)
 thus *?thesis* **by** (*simp only: zpower-zdvd-mono*)
next
 assume $\neg e=1$ **with** e **have** $e1: e=-1$ **by** *auto*
 from U **have** $U * ?P^2 = ?A * ?P$ **by** (*simp add: power2-eq-square*)
 also with $e1$ **have** $\dots = (a*p - e*N*b*q)^2 + N*(-(b*p + e*a*q))^2$
 by (*simp add: qfN-mult1*)
 also have $\dots = (a*p - e*N*b*q)^2 + N*(b*p + e*a*q)^2$
 by (*simp only: power2-minus*)
 also with v **have** $\dots = (a*p - e*N*b*q)^2 + N*v^2*?P^2$
 by (*simp only: power-mult-distrib mult-ac*)
 finally have $(a*p - e*N*b*q)^2 = ?P^2*(U - N*v^2)$
 by (*simp add: mult-ac zdiff-zmult-distrib*)
 hence $?P^2 \text{ dvd } (a*p - e*N*b*q)^2$ **by** (*rule dvdI*)
 thus *?thesis* **by** (*simp only: zpower-zdvd-mono*)
qed
then obtain u **where** $u: a*p - e*N*b*q = ?P*u$ **by** (*auto simp only: dvd-def*)
from e **have** $e2-1: e*e = 1$ **by** *auto*
have $a: a = p*u + e*N*q*v$
proof –
 have $(p*u + e*N*q*v)*?P = p*(?P*u) + (e*N*q)*(?P*v)$
 by (*simp only: zadd-zmult-distrib mult-ac*)
 also with v **have** $\dots = p*(a*p - e*N*b*q) + (e*N*q)*(b*p + e*a*q)$
 by *simp*
 also have $\dots = a*(p^2 + e*e*N*q^2)$
 by (*simp add: power2-eq-square zadd-zmult-distrib2 mult-ac zdiff-zmult-distrib2*)
 also with $e2-1$ **have** $\dots = a*?P$ **by** *simp*
 finally have $(a - (p*u + e*N*q*v))*?P = 0$ **by** *auto*
 moreover from *ass* **have** $?P \neq 0$ **by** (*unfold zprime-def, auto*)
 ultimately show *?thesis* **by** *simp*
qed

moreover have $b: b = p*v - e*q*u$
proof –
have $(p*v - e*q*u)*?P = p*(?P*v) - (e*q)*(?P*u)$
by (*simp only: zdiff-zmult-distrib mult-ac*)
also with $v u$ **have** $\dots = p*(b*p + e*a*q) - e*q*(a*p - e*N*b*q)$ **by** *simp*
also have $\dots = b*(p^2 + e*e*N*q^2)$
by (*simp add: power2-eq-square zadd-zmult-distrib2 mult-ac zdiff-zmult-distrib2*)
also with $e2-1$ **have** $\dots = b * ?P$ **by** *simp*
finally have $(b - (p*v - e*q*u))*?P = 0$ **by** *auto*
moreover from *ass* **have** $?P \neq 0$ **by** (*unfold zprime-def, auto*)
ultimately show *?thesis* **by** *simp*
qed
moreover have $?A = (u^2 + N*v^2)*?P$
proof (*cases*)
assume $e=1$
with a **and** b **show** *?thesis* **by** (*simp add: qfN-mult1 zmult-1 mult-ac*)
next
assume $\neg e=1$ **with** e **have** $e=-1$ **by** *simp*
with a **and** b **show** *?thesis* **by** (*simp add: qfN-mult2 zmult-1 mult-ac*)
qed
moreover from e **have** $|e| = 1$.
ultimately show *?thesis* **by** *blast*
qed

corollary *qfN-div-prime-weak*:
 $\llbracket \text{zprime } (p^2 + N*q^2); (p^2 + N*q^2) \text{ dvd } (a^2 + N*b^2) \rrbracket$
 $\implies \exists u v. a^2 + N*b^2 = (u^2 + N*v^2)*(p^2 + N*q^2)$
apply (*subgoal-tac* $\exists u v. a^2 + N*b^2 = (u^2 + N*v^2)*(p^2 + N*q^2)$)
 $\wedge (\exists e. a = p*u + e*N*q*v \wedge b = p*v - e*q*u \wedge |e|=1)$, *blast*)
apply (*rule qfN-div-prime, auto*)
done

corollary *qfN-div-prime-general*: $\llbracket \text{zprime } P; P \text{ dvd } A; \text{is-qfN } A \ N; \text{is-qfN } P \ N \rrbracket$
 $\implies \exists Q. A = Q*P \wedge \text{is-qfN } Q \ N$
apply (*subgoal-tac* $\exists u v. A = (u^2 + N*v^2)*P$)
apply (*unfold is-qfN-def, auto*)
apply (*simp only: qfN-div-prime-weak*)
done

lemma *qfN-power-div-prime*:
assumes *ass*: $\text{zprime } P \wedge P \in \text{zOdd} \wedge P \text{ dvd } A \wedge P^n = p^2 + N*q^2$
 $\wedge A^n = a^2 + N*b^2 \wedge \text{zgcd } a \ b=1 \wedge \text{zgcd } p \ (N*q) = 1 \wedge n>0$
shows $\exists u v. a^2 + N*b^2 = (u^2 + N*v^2)*(p^2 + N*q^2) \wedge \text{zgcd } u \ v=1$
 $\wedge (\exists e. a = p*u + e*N*q*v \wedge b = p*v - e*q*u \wedge |e| = 1)$

proof –
from *ass* **have** $P \text{ dvd } A \wedge n>0$ **by** *simp*
hence $P^n \text{ dvd } A^n$ **by** (*simp add: zpower-zdvd-mono*)
then obtain U **where** $U: A^n = U*P^n$ **by** (*auto simp only: dvd-def mult-ac*)
have $\exists e. P^n \text{ dvd } b*p + e*a*q \wedge |e| = 1$
proof –
have $P^n \text{-dvd-prod: } P^n \text{ dvd } (b*p + a*q)*(b*p - a*q)$
proof –

have $(b*p + a*q)*(b*p - a*q) = (b*p)^2 - (a*q)^2$ **by** (*rule zspecial-product*)
also have $\dots = b^2 * p^2 + b^2 * N * q^2 - b^2 * N * q^2 - a^2 * q^2$
by (*simp add: power-mult-distrib*)
also with *ass* **have** $\dots = b^2 * P^n - q^2 * A^n$
by (*simp only: mult-ac zadd-zmult-distrib zadd-zmult-distrib2*)
also with *U* **have** $\dots = (b^2 - q^2 * U) * P^n$ **by** (*simp only: zdiff-zmult-distrib*)
finally show *?thesis* **by** (*simp add: mult-ac*)

qed

have $P^n \text{ dvd } (b*p + a*q) \vee P^n \text{ dvd } (b*p - a*q)$

proof –

have *PdvdPn*: $P \text{ dvd } P^n$

proof –

from *ass* **have** $\exists m. n = \text{Suc } m$ **by** (*simp add: not0-implies-Suc*)

then obtain *m* **where** $n = \text{Suc } m$ **by** *auto*

hence $P^n = P * (P^m)$ **by** *auto*

thus *?thesis* **by** *auto*

qed

have $\neg P \text{ dvd } b*p + a*q \vee \neg P \text{ dvd } b*p - a*q$

proof (*rule ccontr, simp*)

assume $P \text{ dvd } b*p + a*q \wedge P \text{ dvd } b*p - a*q$

hence $P \text{ dvd } (b*p + a*q) + (b*p - a*q) \wedge P \text{ dvd } (b*p + a*q) - (b*p - a*q)$

by (*simp only: dvd-add, simp only: dvd-diff*)

hence $P \text{ dvd } 2*(b*p) \wedge P \text{ dvd } 2*(a*q)$ **by** (*simp only: mult-2, auto*)

with *ass* **have** $(P \text{ dvd } 2 \vee P \text{ dvd } b*p) \wedge (P \text{ dvd } 2 \vee P \text{ dvd } a*q)$

by (*simp add: zprime-zdvd-zmult-general*)

hence $P \text{ dvd } 2 \vee (P \text{ dvd } b*p \wedge P \text{ dvd } a*q)$ **by** *auto*

moreover have $\neg P \text{ dvd } 2$

proof (*rule ccontr, simp*)

assume *pdvd2*: $P \text{ dvd } 2$

have $P \leq 2$

proof (*rule ccontr*)

assume $\neg P \leq 2$ **hence** *Pl2*: $P > 2$ **by** *simp*

with *pdvd2* **show** *False* **by** (*simp add: zdvd-not-zless*)

qed

moreover from *ass* **have** $P > 1$ **by** (*simp only: zprime-def*)

ultimately have $P = 2$ **by** *auto*

with *ass* **have** $2 \in \text{zOdd}$ **by** *simp*

moreover have $2 \in \text{zEven}$ **by** (*simp add: zEven-def*)

ultimately show *False* **by** (*simp add: odd-iff-not-even*)

qed

ultimately have $P \text{ dvd } b*p \wedge P \text{ dvd } a*q$ **by** *auto*

with *ass* **have** $(P \text{ dvd } b \vee P \text{ dvd } p) \wedge (P \text{ dvd } a \vee P \text{ dvd } q)$

by (*auto simp only: zprime-zdvd-zmult-general*)

moreover have $\neg P \text{ dvd } p \wedge \neg P \text{ dvd } q$

proof (*auto dest: ccontr*)

assume *Pdvdp*: $P \text{ dvd } p$

hence $P \text{ dvd } p^2$ **by** (*simp only: dvd-mult power2-eq-square*)

with *PdvdPn* **have** $P \text{ dvd } P^n - p^2$ **by** (*simp only: dvd-diff*)

with *ass* **have** $P \text{ dvd } N*(q*q)$ **by** (*simp add: power2-eq-square*)

with *ass* **have** $P \text{ dvd } N \vee P \text{ dvd } q$ **by** (*auto dest: zprime-zdvd-zmult-general*)

hence $P \text{ dvd } N*q$ **by** *auto*

with *Pdvdp* **have** $P \text{ dvd } \text{zgcd } p (N*q)$ **by** (*simp add: zgcd-greatest-iff*)

```

    with ass show False by (auto simp add: zprime-def)
  next
    assume P dvd q
    hence PdvdNq: P dvd N*q by simp
    hence P dvd N*q*q by simp
    hence P dvd N*q^2 by (simp add: power2-eq-square mult-ac)
    with PdvdPn have P dvd P^n - N*q^2 by (simp only: dvd-diff)
    with ass have P dvd p*p by (simp add: power2-eq-square)
    with ass have P dvd p by (auto dest: zprime-zdvd-zmult-general)
    with PdvdNq have P dvd zgcd p (N*q) by (simp add: zgcd-greatest-iff)
    with ass show False by (auto simp add: zprime-def)
  qed
  ultimately have P dvd a ∧ P dvd b by auto
  hence P dvd zgcd a b by (simp add: zgcd-greatest-iff)
  with ass show False by (auto simp add: zprime-def)
  qed
  moreover
  { assume ¬ P dvd b*p+a*q
    with Pn-dvd-prod and ass have P^n dvd b*p-a*q
      by (rule-tac a=b*p+a*q in zprime-power-zdvd-cancel-left, simp) }
  moreover
  { assume ¬ P dvd b*p-a*q
    with Pn-dvd-prod and ass have P^n dvd b*p+a*q
      by (rule-tac a=b*p+a*q in zprime-power-zdvd-cancel-right, simp) }
  ultimately show ?thesis by auto
  qed
  moreover
  { assume P^n dvd b*p + a*q
    hence P^n dvd b*p + 1*a*q ∧ |1| = (1::int) by simp }
  moreover
  { assume P^n dvd b*p - a*q
    hence P^n dvd b*p + (-1)*a*q ∧ |-1| = (1::int) by simp }
  ultimately show ?thesis by blast
  qed
  then obtain v e where v: b*p + e*a*q = P^n*v and e: |e| = 1
    by (auto simp only: dvd-def)
  have P^n dvd a*p - e*N*b*q
  proof (cases)
    assume e1: e = 1
    from U have  $(P^n)^2 * U = A^n * P^n$  by (simp add: power2-eq-square mult-ac)
    also with e1 ass have  $\dots = (a*p - e*N*b*q)^2 + N*(b*p + e*a*q)^2$ 
      by (simp only: qfN-mult2 add-commute zmult-1)
    also with v have  $\dots = (a*p - e*N*b*q)^2 + (P^n)^2 * (N*v^2)$ 
      by (simp only: power-mult-distrib mult-ac)
    finally have  $(a*p - e*N*b*q)^2 = (P^n)^2 * U - (P^n)^2 * N*v^2$  by simp
    also have  $\dots = (P^n)^2 * (U - N*v^2)$  by (simp only: zdiff-zmult-distrib2)
    finally have  $(P^n)^2 dvd (a*p - e*N*b*q)^2$  by (rule dvdI)
    thus ?thesis by (simp only: zpower-zdvd-mono)
  next
    assume ¬ e=1 with e have e1: e=-1 by auto
    from U have  $(P^n)^2 * U = A^n * P^n$  by (simp add: power2-eq-square)
    also with e1 ass have  $\dots = (a*p - e*N*b*q)^2 + N*(-(b*p + e*a*q))^2$ 

```

by (simp add: qfN-mult1)
 also have $\dots = (a*p - e*N*b*q)^2 + N*(b*p + e*a*q)^2$
 by (simp only: power2-minus)
 also with v and ass have $\dots = (a*p - e*N*b*q)^2 + N*v^2*(P^n)^2$
 by (simp only: power-mult-distrib mult-ac)
 finally have $(a*p - e*N*b*q)^2 = (P^n)^2*U - (P^n)^2*N*v^2$ by simp
 also have $\dots = (P^n)^2 * (U - N*v^2)$ by (simp only: zdiff-zmult-distrib2)
 finally have $(P^n)^2 dvd (a*p - e*N*b*q)^2$ by (rule dvdI)
 thus ?thesis by (simp only: zpower-zdvd-mono)

qed

then obtain u where $u: a*p - e*N*b*q = P^n*u$ by (auto simp only: dvd-def)

from e have $e2-1: e*e = 1$ by auto

have $a: a = p*u + e*N*q*v$

proof -

from ass have $(p*u + e*N*q*v)*P^n = p*(P^n*u) + (e*N*q)*(P^n*v)$
 by (simp only: zadd-zmult-distrib mult-ac)

also with v and u have $\dots = p*(a*p - e*N*b*q) + (e*N*q)*(b*p + e*a*q)$
 by simp

also have $\dots = a*(p^2 + e*e*N*q^2)$
 by (simp add: power2-eq-square zadd-zmult-distrib2 mult-ac zdiff-zmult-distrib2)

also with $e2-1$ and ass have $\dots = a*P^n$ by simp

finally have $(a - (p*u + e*N*q*v))*P^n = 0$ by auto

moreover from ass have $P^n \neq 0$
 by (unfold zprime-def, auto simp add: power-eq-0-iff)

ultimately show ?thesis by auto

qed

moreover have $b: b = p*v - e*q*u$

proof -

from ass have $(p*v - e*q*u)*P^n = p*(P^n*v) - (e*q)*(P^n*u)$
 by (simp only: zdiff-zmult-distrib mult-ac)

also with v and u have $\dots = p*(b*p + e*a*q) - e*q*(a*p - e*N*b*q)$ by simp

also have $\dots = b*(p^2 + e*e*N*q^2)$
 by (simp add: power2-eq-square zadd-zmult-distrib2 mult-ac zdiff-zmult-distrib2)

also with $e2-1$ and ass have $\dots = b * P^n$ by simp

finally have $(b - (p*v - e*q*u))*P^n = 0$ by auto

moreover from ass have $P^n \neq 0$
 by (unfold zprime-def, auto simp add: power-eq-0-iff)

ultimately show ?thesis by auto

qed

moreover have $A^n = (u^2 + N*v^2)*P^n$

proof (cases)

assume $e=1$

with a and b and ass show ?thesis by (simp add: qfN-mult1 zmult-1 mult-ac)

next

assume $\neg e=1$ with e have $e=-1$ by simp

with a and b and ass show ?thesis by (simp add: qfN-mult2 zmult-1 mult-ac)

qed

moreover have $zgcd\ u\ v=1$

proof -

let $?g = zgcd\ u\ v$

have $?g\ dvd\ u \wedge ?g\ dvd\ v$ by auto

hence $?g\ dvd\ u*p + v*(e*N*q) \wedge ?g\ dvd\ v*p - u*(e*q)$ by simp

with a and b have $?g \text{ dvd } a \wedge ?g \text{ dvd } b$ by (auto simp only: mult-ac)
hence $?g \text{ dvd } \text{zgcd } a \ b$ by (simp add: zgcd-greatest-iff)
with ass have $?g = 1 \vee ?g = -1$ by simp
moreover have $?g \geq 0$ by (rule zgcd-geq-zero)
ultimately show $?thesis$ by auto
qed
moreover from e and ass have
 $|e| = 1 \wedge A \hat{n} = a^{\wedge 2} + N * b^{\wedge 2} \wedge P \hat{n} = p^{\wedge 2} + N * q^{\wedge 2}$ **by simp**
ultimately show $?thesis$ by auto
qed

lemma $qfN\text{-primedivisor-not}$:
assumes ass : $zprime \ P \wedge Q > 0 \wedge is\text{-}qfN \ (P * Q) \ N \wedge \neg is\text{-}qfN \ P \ N$
shows $\exists R. (zprime \ R \wedge R \text{ dvd } Q \wedge \neg is\text{-}qfN \ R \ N)$
proof (rule ccontr, auto)
assume $ass2$: $\forall R. R \text{ dvd } Q \longrightarrow zprime \ R \longrightarrow is\text{-}qfN \ R \ N$
have $\exists ps. primel \ ps \wedge int \ (prod \ ps) = Q$
proof -
from ass have $Q=1 \vee nat(Q) > Suc \ 0$ by auto
moreover
{ assume $Q=1$ hence $primel \ [] \wedge int \ (prod \ []) = Q$ by (simp add: primel-def)
hence $?thesis$ by auto }
moreover
{ assume $nat(Q) > Suc \ 0$
then have $\exists ps. primel \ ps \wedge prod \ ps = nat(Q)$ by (simp only: factor-exists)
with ass have $?thesis$ by auto }
ultimately show $?thesis$ by blast
qed
then obtain ps where ps : $primel \ ps \wedge int(prod \ ps) = Q$ by auto
have ps -lemma: $(primel \ ps \wedge is\text{-}qfN \ (P * int(prod \ ps)) \ N$
 $\wedge (\forall R. (zprime \ R \wedge R \text{ dvd } int(prod \ ps)) \longrightarrow is\text{-}qfN \ R \ N) \implies False$
(is $?B \ ps \implies False$)
proof (induct ps)
case Nil hence $is\text{-}qfN \ P \ N$ by simp
with ass show $False$ by simp
next
case (Cons $p \ ps$)
hence $ass3$: $?B \ ps \implies False$
and IH : $?B \ (p \# ps)$ by simp-all
hence p : $zprime \ (int \ p)$ and $int \ p \text{ dvd } int(prod(p \# ps))$
by (auto simp add: primel-def prime-impl-zprime-int int-mult)
moreover with IH have qfN : $is\text{-}qfN \ (int \ p) \ N$
and $int \ p \text{ dvd } P * int(prod(p \# ps))$ and $is\text{-}qfN \ (P * int(prod(p \# ps))) \ N$
by auto
ultimately obtain S where S : $P * int(prod(p \# ps)) = S * (int \ p) \wedge is\text{-}qfN \ S \ N$
by (auto dest: qfN-div-prime-general simp del: dvd-mult)
hence $(int \ p) * (P * int(prod \ ps) - S) = 0$ by (auto simp add: int-mult)
with $p \ S$ have $is\text{-}qfN \ (P * int(prod \ ps)) \ N$ by (auto simp add: zprime-def)
moreover from IH have $primel \ ps$ by (simp add: primel-def)
moreover from IH have $\forall R. zprime \ R \wedge R \text{ dvd } int(prod \ ps) \longrightarrow is\text{-}qfN \ R \ N$
by (auto simp add: int-mult)
ultimately have $?B \ ps$ by simp

```

  with ass3 show False by simp
qed
with ps ass2 ass show False by auto
qed

```

lemma *qfN-oddprime-cube*:

```

[[ zprime ( $p^2 + N*q^2$ ); ( $p^2 + N*q^2$ ) ∈ zOdd;  $p \neq 0$ ;  $N \geq 1$  ]
⇒ ∃ a b. ( $p^2 + N*q^2$ )3 =  $a^2 + N*b^2$  ∧ zgcd a ( $N*b$ )=1

```

proof –

```

let ?P =  $p^2 + N*q^2$ 
assume P: zprime ?P and Podd: ?P ∈ zOdd and p0:  $p \neq 0$  and N1:  $N \geq 1$ 
have suc23:  $3 = \text{Suc } 2$  by simp
let ?a =  $p*(p^2 - 3*N*q^2)$ 
let ?b =  $q*(3*p^2 - N*q^2)$ 
have abP: ?P3 = ?a2 +  $N*?b^2$  by (simp add: nat-number ring-simps)
have zgcd ?b ?a ≠ 1 ⇒ ?P dvd p

```

proof –

```

let ?h = zgcd ?b ?a
assume h1: ?h ≠ 1
have ?h ≥ 0 by (rule zgcd-geq-zero)
hence ?h = 0 ∨ ?h = 1 ∨ ?h > 1 by arith
with h1 have ?h = 0 ∨ ?h > 1 by auto

```

moreover

```

{ assume ?h = 0 hence  $\text{nat}|?b| = 0 \wedge \text{nat}|?a| = 0$ 
  by (unfold zgcd-def, auto simp add: gcd-zero)
  hence ?a = 0 ∧ ?b = 0 by arith
  with abP have ?P3 = 0 by auto
  with P have False by (unfold zprime-def, auto)
  hence ?thesis by simp }

```

moreover

```

{ assume ?h > 1 hence ∃ g. zprime g ∧ g dvd ?h by (rule zprime-factor-exists)
  then obtain g where g: zprime g ∧ g dvd ?h by blast
  hence g dvd ?b ∧ g dvd ?a by (simp add: zgcd-greatest-iff)
  with g have g1: g dvd  $q \vee g \text{ dvd } 3*p^2 - N*q^2$ 
    and g2: g dvd  $p \vee g \text{ dvd } p^2 - 3*N*q^2$ 
    by (auto simp add: zprime-zdvd-zmult-general)
  from g have gpos:  $g \geq 0$  by (auto simp only: zprime-def)
  have g dvd ?P

```

proof (*cases*)

```

  assume g dvd q
  hence gNq: g dvd  $N*q^2$  by (auto simp add: dvd-def power2-eq-square)
  show ?thesis

```

proof (*cases*)

```

  assume gp: g dvd p
  hence g dvd  $p^2$  by (auto simp add: dvd-def power2-eq-square)
  with gNq show ?thesis by auto

```

next

```

  assume ¬ g dvd p with g2 have g dvd  $p^2 - 3*N*q^2$  by auto
  moreover from gNq have g dvd  $4*(N*q^2)$  by (rule dvd-mult)
  ultimately have g dvd  $p^2 - 3*(N*q^2) + 4*(N*q^2)$ 
    by (simp only: mult-ac dvd-add)

```

```

  moreover have  $p^2 - 3*(N*q^2) + 4*(N*q^2) = p^2 + N*q^2$  by arith

```

```

    ultimately show ?thesis by simp
  qed
next
assume  $\neg g \text{ dvd } q$  with  $g1$  have  $gpq: g \text{ dvd } 3*p^2 - N*q^2$  by simp
show ?thesis
proof (cases)
  assume  $g \text{ dvd } p$ 
  hence  $g \text{ dvd } 4*p^2$  by (auto simp add: dvd-def power2-eq-square)
  with  $gpq$  have  $g \text{ dvd } 4*p^2 - (3*p^2 - N*q^2)$  by (simp only: dvd-diff)
  moreover have  $4*p^2 - (3*p^2 - N*q^2) = p^2 + N*q^2$  by arith
  ultimately show ?thesis by simp
next
assume  $\neg g \text{ dvd } p$  with  $g2$  have  $g \text{ dvd } p^2 - 3*N*q^2$  by auto
with  $gpq$  have  $g \text{ dvd } 3*p^2 - N*q^2 - (p^2 - 3*N*q^2)$ 
  by (simp only: dvd-diff)
moreover have  $3*p^2 - N*q^2 - (p^2 - 3*N*q^2) = 2*?P$  by auto
ultimately have  $g \text{ dvd } 2*?P$  by simp
with  $g$  have  $g \text{ dvd } 2 \vee g \text{ dvd } ?P$  by (simp only: zprime-zdvd-zmult)
moreover have  $\neg g \text{ dvd } 2$ 
proof (rule ccontr, simp)
  assume  $gdvd2: g \text{ dvd } 2$ 
  have  $g \leq 2$ 
  proof (rule ccontr)
    assume  $\neg g \leq 2$  hence  $g > 2$  by simp
    moreover have  $(0::int) < 2$  by auto
    ultimately have  $\neg g \text{ dvd } 2$  by (auto simp only: zdvd-not-zless)
    with  $gdvd2$  show False by simp
  qed
moreover from  $g$  have  $g \geq 2$  by (simp add: zprime-def)
ultimately have  $g = 2$  by auto
with  $g$  have  $2 \text{ dvd } ?a \wedge 2 \text{ dvd } ?b$  by (auto simp add: zgcd-greatest-iff)
hence  $2 \text{ dvd } ?a^2 \wedge 2 \text{ dvd } N*?b^2$ 
  by (simp add: power2-eq-square)
with  $abP$  have  $2 \text{ dvd } ?P^3$  by (simp only: dvd-add)
hence  $?P^3 \in zEven$  by (auto simp add: dvd-def zEven-def)
moreover have  $?P^3 \in zOdd$ 
proof -
  from  $Podd$  have  $?P*?P^2 \in zOdd$ 
  by (simp only: odd-times-odd power2-eq-square)
  thus ?thesis by (simp only: cube-square)
qed
ultimately show False by (auto simp only: odd-iff-not-even)
qed
ultimately show ?thesis by simp
qed
qed
with  $P$   $gpos$  have  $g = 1 \vee g = ?P$  by (auto simp only: zprime-def)
with  $g$  have  $g = ?P$  by (simp add: zprime-def)
with  $g$  have  $Pab: ?P \text{ dvd } ?a \wedge ?P \text{ dvd } ?b$  by (auto simp add: zgcd-greatest-iff)
have ?thesis
proof -
  from  $Pab$   $P$  have  $?P \text{ dvd } p \vee ?P \text{ dvd } p^2 - 3*N*q^2$ 

```

```

    by (auto simp add: zprime-zdvd-zmult-general)
  moreover
  { assume ?P dvd p^2 - 3*N*q^2
    moreover have ?P dvd 3*(p^2 + N*q^2)
      by (auto simp only: dvd-refl dvd-mult)
    ultimately have ?P dvd p^2 - 3*N*q^2 + 3*(p^2 + N*q^2)
      by (simp only: dvd-add)
    hence ?P dvd 4*p^2 by auto
    with P have ?P dvd 4 ∨ ?P dvd p^2
      by (simp only: zprime-zdvd-zmult-general)
    moreover have ¬ ?P dvd 4
    proof (rule ccontr, simp)
      assume P dvd 4: ?P dvd 4
      have ?P ≤ 4
      proof (rule ccontr)
        assume ¬ ?P ≤ 4 hence ?P > 4 by simp
        moreover have (0::int) < 4 by auto
        ultimately have ¬ ?P dvd 4 by (auto simp only: zdvd-not-zless)
        with P dvd 4 show False by simp
      qed
    moreover from P have ?P ≥ 2 by (auto simp add: zprime-def)
    moreover have ?P ≠ 2 ∧ ?P ≠ 4
    proof (rule ccontr, simp)
      assume ?P = 2 ∨ ?P = 4 hence ?P ∈ zEven
        by (auto simp add: zEven-def)
      with P odd show False by (simp add: odd-iff-not-even)
    qed
    ultimately have ?P = 3 by auto
    with P dvd 4 have (3::int) dvd 4 by simp
    thus False by arith
  qed
  ultimately have ?P dvd p*p by (simp add: power2-eq-square)
  with P have ?thesis by (auto dest: zprime-zdvd-zmult-general) }
  ultimately show ?thesis by auto
qed }
ultimately show ?thesis by blast
qed
moreover have zgcd N ?a ≠ 1 ⇒ ?P dvd p
proof -
  let ?h = zgcd N ?a
  assume h1: ?h ≠ 1
  have ?h ≥ 0 by (rule zgcd-geq-zero)
  hence ?h = 0 ∨ ?h = 1 ∨ ?h > 1 by arith
  with h1 have ?h = 0 ∨ ?h > 1 by auto
  moreover
  { assume ?h = 0 hence nat|N| = 0 ∧ nat|?a| = 0
    by (unfold zgcd-def, auto simp add: gcd-zero)
    hence N = 0 by arith
    with N1 have False by auto
    hence ?thesis by simp }
  moreover
  { assume ?h > 1 hence ∃ g. zprime g ∧ g dvd ?h by (rule zprime-factor-exists)

```

```

then obtain g where g: zprime g ∧ g dvd ?h by blast
hence gN: g dvd N and g dvd ?a by (auto simp add: zgcd-greatest-iff)
hence g dvd p*p^2 - N*(3*p*q^2)
  by (auto simp only: zdiff-zmult-distrib2 mult-ac)
with gN have g dvd p*p^2 - N*(3*p*q^2) + N*(3*p*q^2)
  by (simp only: dvd-add dvd-mult2)
hence g dvd p*p^2 by simp
with g have g dvd p ∨ g dvd p*p
  by (simp add: zprime-zdvd-zmult-general power2-eq-square)
with g have gp: g dvd p by (auto dest: zprime-zdvd-zmult-general)
hence g dvd p^2 by (simp add: power2-eq-square)
with gN have gP: g dvd ?P by auto
from g have g ≥ 0 by (simp add: zprime-def)
with gP P have g = 1 ∨ g = ?P by (auto simp only: zprime-def)
with g have g = ?P by (auto simp only: zprime-def)
with gp have ?thesis by simp }
ultimately show ?thesis by auto
qed
moreover have ¬ ?P dvd p
proof (rule ccontr, clarsimp)
  assume Pdvdp: ?P dvd p
  have p^2 ≥ ?P^2
  proof (rule ccontr)
    assume ¬ p^2 ≥ ?P^2 hence pP: p^2 < ?P^2 by simp
    moreover with p0 have p^2 > 0 by (simp add: zero-less-power2)
    ultimately have ¬ ?P^2 dvd p^2 by (simp add: zdvd-not-zless)
    with Pdvdp show False by (simp add: zpower-zdvd-mono)
  qed
  moreover with P have ?P*1 < ?P*?P
    by (unfold zprime-def, auto simp only: zmult-zless-mono2)
  ultimately have p^2 > ?P by (auto simp add: power2-eq-square)
  hence neg: N*q^2 < 0 by auto
  show False
  proof -
    have is-qn (0^2 + N*q^2) N by (auto simp only: is-qn-def)
    with N1 have 0^2 + N*q^2 ≥ 0 by (rule qn-pos)
    with neg show False by simp
  qed
qed
ultimately have zgcd ?b ?a = 1 ∧ zgcd N ?a = 1 by auto
hence zgcd (N*?b) ?a = 1 by (simp only: zgcd-zmult-cancel)
with abP show ?thesis by (auto simp only: zgcd-commute)
qed

```

3.4 Uniqueness ($N > 1$)

lemma *qn-prime-unique*:

```

[[ zprime (a^2+N*b^2); N > 1; a^2+N*b^2 = c^2+N*d^2 ]]
⇒ (|a| = |c| ∧ |b| = |d|)

```

proof -

```

let ?P = a^2+N*b^2

```

```

assume P: zprime ?P and N: N > 1 and abcdN: ?P = c^2 + N*d^2

```

```

have mult: (a*d+b*c)*(a*d-b*c) = ?P*(d^2-b^2)
proof -
  have (a*d+b*c)*(a*d-b*c) = (a^2 + N*b^2)*d^2 - b^2*(c^2 + N*d^2)
    by (simp add: nat-number ring-simps)
  with abcdN show ?thesis by (simp add: ring-simps)
qed
have ?P dvd a*d+b*c ∨ ?P dvd a*d-b*c
proof -
  from mult have ?P dvd (a*d+b*c)*(a*d-b*c) by simp
  with P show ?thesis by (simp add: zprime-zdvd-zmult-general)
qed
moreover
{ assume ?P dvd a*d+b*c
  then obtain Q where Q: a*d+b*c = ?P*Q by (auto simp add: dvd-def)
  from abcdN have ?P^2 = (a^2 + N*b^2) * (c^2 + N*d^2)
    by (simp add: power2-eq-square)
  also have ... = (a*c-N*b*d)^2 + N*(a*d+b*c)^2 by (rule qfN-mult2)
  also with Q have ... = (a*c-N*b*d)^2 + N*Q^2*?P^2
    by (simp add: mult-ac power-mult-distrib)
  also have ... ≥ N*Q^2*?P^2 by (simp add: zero-le-power2)
  finally have pos: ?P^2 ≥ ?P^2*(Q^2*N) by (simp add: mult-ac)
  have b^2 = d^2
  proof (rule ccontr)
    assume b^2 ≠ d^2
    with P mult Q have Q ≠ 0 by (unfold zprime-def, auto)
    hence Q^2 > 0 by (simp add: zero-less-power2)
    moreover with N have Q^2*N > Q^2*1 by (simp only: zmult-zless-mono2)
    ultimately have Q^2*N > 1 by arith
    moreover with P have ?P^2 > 0 by (simp add: zprime-def zero-less-power2)
    ultimately have ?P^2*1 < ?P^2*(Q^2*N) by (simp only: zmult-zless-mono2)
    with pos show False by simp
  qed }
moreover
{ assume ?P dvd a*d-b*c
  then obtain Q where Q: a*d-b*c = ?P*Q by (auto simp add: dvd-def)
  from abcdN have ?P^2 = (a^2 + N*b^2) * (c^2 + N*d^2)
    by (simp add: power2-eq-square)
  also have ... = (a*c+N*b*d)^2 + N*(a*d-b*c)^2 by (rule qfN-mult1)
  also with Q have ... = (a*c+N*b*d)^2 + N*Q^2*?P^2
    by (simp add: mult-ac power-mult-distrib)
  also have ... ≥ N*Q^2*?P^2 by (simp add: zero-le-power2)
  finally have pos: ?P^2 ≥ ?P^2*(Q^2*N) by (simp add: mult-ac)
  have b^2 = d^2
  proof (rule ccontr)
    assume b^2 ≠ d^2
    with P mult Q have Q ≠ 0 by (unfold zprime-def, auto)
    hence Q^2 > 0 by (simp add: zero-less-power2)
    moreover with N have Q^2*N > Q^2*1 by (simp only: zmult-zless-mono2)
    ultimately have Q^2*N > 1 by arith
    moreover with P have ?P^2 > 0 by (simp add: zprime-def zero-less-power2)
    ultimately have ?P^2*1 < ?P^2 * (Q^2*N) by (simp only: zmult-zless-mono2)
    with pos show False by simp
  }
}

```

```

    qed }
  ultimately have  $bd: b^2 = d^2$  by blast
  moreover with  $abcdN$  have  $a^2 = c^2$  by auto
  ultimately show ?thesis by (auto simp only: power2-eq-iff-abs-eq)
qed

lemma qfN-square-prime:
  assumes ass:
     $zprime (p^2 + N * q^2) \wedge N > 1 \wedge (p^2 + N * q^2)^2 = r^2 + N * s^2 \wedge zgcd\ r\ s = 1$ 
  shows  $|r| = |p^2 - N * q^2| \wedge |s| = |2 * p * q|$ 
proof -
  let  $?P = p^2 + N * q^2$ 
  let  $?A = r^2 + N * s^2$ 
  from ass have  $P1: ?P > 1$  by (simp add: zprime-def)
  from ass have  $APP: ?A = ?P * ?P$  by (simp only: power2-eq-square)
  with ass have  $zprime\ ?P \wedge ?P\ dvd\ ?A$  by (simp add: dvdI)
  then obtain  $u\ v\ e$  where uve:
     $?A = (u^2 + N * v^2) * ?P \wedge r = p * u + e * N * q * v \wedge s = p * v - e * q * u \wedge |e| = 1$ 
    by (frule-tac p=p in qfN-div-prime, auto)
  with  $APP\ P1\ ass$  have  $zprime (u^2 + N * v^2) \wedge N > 1 \wedge u^2 + N * v^2 = ?P$ 
    by auto
  hence  $|u| = |p| \wedge |v| = |q|$  by (auto dest: qfN-prime-unique)
  then obtain  $f\ g$  where  $f: u = f * p \wedge |f| = 1$  and  $g: v = g * q \wedge |g| = 1$ 
    by (blast dest: abs-eq-impl-unitfactor)
  with uve have  $r = f * p * p + (e * g) * N * q * q \wedge s = g * p * q - (e * f) * p * q$  by simp
  hence  $rs: r = f * p^2 + (e * g) * N * q^2 \wedge s = (g - e * f) * p * q$ 
    by (auto simp only: power2-eq-square zdiff-zmult-distrib)
  moreover have  $s \neq 0$ 
  proof (rule ccontr, simp)
    assume  $s0: s = 0$ 
    hence  $zgcd\ r\ s = |r|$  by (simp only: zgcd-0)
    with ass have  $|r| = 1$  by simp
    hence  $r^2 = 1$  by (auto simp add: abs-power2-distrib)
    with  $s0$  have  $?A = 1$  by simp
    moreover have  $?P^2 > 1$ 
    proof -
      from  $P1$  have  $1 < ?P \wedge (0::int) \leq 1 \wedge (0::nat) < 2$  by auto
      hence  $?P^2 > 1^2$  by (simp only: power-strict-mono)
      thus ?thesis by auto
    qed
  qed
  moreover from ass have  $?A = ?P^2$  by simp
  ultimately show False by auto
qed

ultimately have  $g \neq e * f$  by auto
moreover from  $f\ g\ uve$  have  $|g| = |e * f|$  by auto
ultimately have  $g = -(e * f)$  by arith
with  $rs\ uve$  have  $r = f * (p^2 - N * q^2) \wedge s = -(e * f) * 2 * p * q$ 
  by (auto simp add: power2-eq-square zdiff-zmult-distrib2)
hence  $|r| = |f| * |p^2 - N * q^2|$ 
   $\wedge |s| = |e| * |f| * |2 * p * q|$ 
  by (auto simp add: abs-mult)
with uve f g show ?thesis by (auto simp only: zmult-1)

```

qed

lemma *qfN-cube-prime*:

assumes *ass*: $zprime (p^2 + N*q^2) \wedge N > 1$
 $\wedge (p^2 + N*q^2)^3 = a^2 + N*b^2 \wedge zgcd a b = 1$
 shows $|a| = |p^3 - 3*N*p*q^2| \wedge |b| = |3*p^2*q - N*q^3|$

proof –

let $?P = p^2 + N*q^2$
 let $?A = a^2 + N*b^2$
 from *ass* have $P1: ?P > 1$ by (*simp add: zprime-def*)
 with *ass* have $APP: ?A = ?P*?P^2$ by (*auto simp only: cube-square*)
 with *ass* have $zprime ?P \wedge ?P \text{ dvd } ?A$ by (*simp add: dvdI*)
 then obtain $u v e$ where *uve*:
 $?A = (u^2 + N*v^2)*?P \wedge a = p*u + e*N*q*v \wedge b = p*v - e*q*u \wedge |e| = 1$
 by (*frule-tac p=p in qfN-div-prime, auto*)
 have $zgcd u v = 1$

proof –

let $?g = zgcd u v$
 have $?g \text{ dvd } u \wedge ?g \text{ dvd } v$ by (*auto simp add: zgcd-greatest-iff*)
 with *uve* have $?g \text{ dvd } a \wedge ?g \text{ dvd } b$ by *auto*
 hence $?g \text{ dvd } zgcd a b$ by (*auto simp add: zgcd-greatest-iff*)
 with *ass* have $?g \text{ dvd } 1$ by *simp*
 moreover have $?g \geq 0$ by (*rule zgcd-geq-zero*)
 ultimately show *thesis* by *auto*

qed

with $P1$ *uve* APP *ass* have $zprime ?P \wedge N > 1 \wedge ?P^2 = u^2 + N*v^2$
 $\wedge zgcd u v = 1$ by (*auto simp add: mult-ac*)

hence $|u| = |p^2 - N*q^2| \wedge |v| = |2*p*q|$ by (*rule qfN-square-prime*)

then obtain $f g$ where $f: u = f*(p^2 - N*q^2) \wedge |f| = 1$

and $g: v = g*(2*p*q) \wedge |g| = 1$ by (*blast dest: abs-eq-impl-unitfactor*)

with *uve* have $a = p*f*(p^2 - N*q^2) + e*N*q*g*2*p*q$

$\wedge b = p*g*2*p*q - e*q*f*(p^2 - N*q^2)$ by *auto*

hence $ab: a = f*p*p^2 + -f*N*p*q^2 + 2*e*g*N*p*q^2$

$\wedge b = 2*g*p^2*q - e*f*p^2*q + e*f*N*q*q^2$

by (*auto simp add: mult-ac zdiff-zmult-distrib2 power2-eq-square*)

from f have $f2: f^2 = 1$ by (*auto simp add: abs-power2-distrib*)

from g have $g2: g^2 = 1$ by (*auto simp add: abs-power2-distrib*)

have $e \neq f*g$

proof (*rule ccontr, simp*)

assume $efg: e = f*g$

with $ab g$ have $a = f*p*p^2 + f*N*p*q^2$ by (*auto simp add: power2-eq-square*)

hence $a = (f*p)*?P$ by (*auto simp add: zadd-zmult-distrib2 mult-ac*)

hence $Pa: ?P \text{ dvd } a$ by *auto*

from $efg f ab$ have $b = g*p^2*q + g*N*q*q^2$ by (*auto simp add: power2-eq-square*)

hence $b = (g*q)*?P$ by (*auto simp add: zadd-zmult-distrib2 mult-ac*)

hence $?P \text{ dvd } b$ by *auto*

with Pa have $?P \text{ dvd } zgcd a b$ by (*simp add: zgcd-greatest-iff*)

with *ass* have $?P \text{ dvd } 1$ by *auto*

with $P1$ show *False* by *auto*

qed

moreover from $f g$ *uve* have $|e| = |f*g|$ by *auto*

ultimately have $e = -(f*g)$ by *arith*

with $abfg$ have $a = f * p * p^2 - 3 * f * N * p * q^2 \wedge b = 3 * g * p^2 * q - g * N * q * q^2$
 by (auto simp add: power2-eq-square)
 hence $a = f * (p^3 - 3 * N * p * q^2) \wedge b = g * (3 * p^2 * q - N * q^3)$
 by (auto simp only: zdiff-zmult-distrib2 mult-ac cube-square)
 with fg show ?thesis by (auto simp add: zmult-1 abs-mult)
 qed

3.5 The case $N = 3$

lemma $qf3$ -even: $a^2 + 3 * b^2 \in zEven \implies \exists B. a^2 + 3 * b^2 = 4 * B \wedge is_qfN B 3$

proof –

let $?A = a^2 + 3 * b^2$

assume even: $?A \in zEven$

have $(a \in zOdd \wedge b \in zOdd) \vee (a \in zEven \wedge b \in zEven)$

proof (rule ccontr, auto dest: not-odd-impl-even)

assume $a \notin zOdd$ and $b \notin zEven$

hence $a \in zEven \wedge b \in zOdd$ by (auto simp only: odd-iff-not-even)

hence $a^2 \in zEven \wedge b^2 \in zOdd$

by (auto simp only: power2-eq-square odd-times-odd even-times-either)

moreover have $3 \in zOdd$ by (unfold zOdd-def, auto)

ultimately have $?A \in zOdd$ by (auto simp add: odd-times-odd even-plus-odd)

with even show False by (simp add: odd-iff-not-even)

next

assume $a \notin zEven$ and $b \notin zOdd$

hence $a \in zOdd \wedge b \in zEven$ by (auto simp only: odd-iff-not-even)

hence $a^2 \in zOdd \wedge b^2 \in zEven$

by (auto simp only: power2-eq-square odd-times-odd even-times-either)

moreover hence $b^2 * 3 \in zEven$ by (simp only: even-times-either)

ultimately have $b^2 * 3 + a^2 \in zOdd$ by (auto simp add: even-plus-odd)

hence $?A \in zOdd$ by (simp only: mult-ac add-ac)

with even show False by (simp add: odd-iff-not-even)

qed

moreover

{ assume $a \in zEven \wedge b \in zEven$

then obtain cd where $abcd: a = 2 * c \wedge b = 2 * d$ by (unfold zEven-def, auto)

hence $?A = 4 * (c^2 + 3 * d^2)$ by (simp add: power-mult-distrib)

moreover have $is_qfN (c^2 + 3 * d^2) 3$ by (unfold is-qfN-def, auto)

ultimately have ?thesis by blast }

moreover

{ assume $a \in zOdd \wedge b \in zOdd$

then obtain cd where $abcd: a = 2 * c + 1 \wedge b = 2 * d + 1$

by (unfold zOdd-def, auto)

have $c - d \in zOdd \vee c - d \in zEven$ by (rule-tac $x = c - d$ in even-odd-disj)

moreover

{ assume $c - d \in zEven$

then obtain e where $c - d = 2 * e$ by (auto simp add: zEven-def)

with $abcd$ have $e1: a - b = 4 * e$ by arith

hence $e2: a + 3 * b = 4 * (e + b)$ by auto

have $4 * ?A = (a + 3 * b)^2 + 3 * (a - b)^2$

by (simp add: nat-number ring-simps)

also with $e1 e2$ have $\dots = (4 * (e + b))^2 + 3 * (4 * e)^2$ by (simp(no-asm-simp))

finally have $?A = 4 * ((e + b)^2 + 3 * e^2)$ by (simp add: nat-number ring-simps)

moreover have $is\text{-}qfN ((e+b)^2 + 3*e^2) \exists$ **by** (*unfold is-qfN-def, auto*)
ultimately have *?thesis* **by** *blast* }
moreover
{ **assume** $c-d \in zOdd$
then obtain e **where** $c-d = 2*e+1$ **by** (*auto simp add: zOdd-def*)
with $abcd$ **have** $e1: a+b = 4*(e+d+1)$ **by** *auto*
hence $e2: a-3*b = 4*(e+d-b+1)$ **by** *auto*
have $4*A = (a-3*b)^2 + 3*(a+b)^2$
by (*simp add: nat-number ring-simps*)
also with $e1 e2$ **have** $\dots = (4*(e+d-b+1))^2 + 3*(4*(e+d+1))^2$
by (*simp (no-asm-simp)*)
finally have $?A = 4*((e+d-b+1)^2 + 3*(e+d+1)^2)$
by (*simp add: nat-number ring-simps*)
moreover have $is\text{-}qfN ((e+d-b+1)^2 + 3*(e+d+1)^2) \exists$
by (*unfold is-qfN-def, auto*)
ultimately have *?thesis* **by** *blast* }
ultimately have *?thesis* **by** *auto* }
ultimately show *?thesis* **by** *auto*
qed

lemma *qf3-even-general*: [$is\text{-}qfN A \exists; A \in zEven$]
 $\implies \exists B. A = 4*B \wedge is\text{-}qfN B \exists$

proof –
assume $A \in zEven$ **and** $is\text{-}qfN A \exists$
then obtain $a b$ **where** $A = a^2 + 3*b^2$
and $a^2 + 3*b^2 \in zEven$ **by** (*unfold is-qfN-def, auto*)
thus *?thesis* **by** (*auto simp add: qf3-even*)
qed

lemma *qf3-oddprimedivisor-not*:

assumes $ass: zprime P \wedge P \in zOdd \wedge Q > 0 \wedge is\text{-}qfN (P*Q) \exists \wedge \neg is\text{-}qfN P \exists$
shows $\exists R. zprime R \wedge R \in zOdd \wedge R \text{ dvd } Q \wedge \neg is\text{-}qfN R \exists$

proof (*rule ccontr, simp*)

assume $ass2: \forall R. R \text{ dvd } Q \implies R \in zOdd \implies zprime R \implies is\text{-}qfN R \exists$
(is ?A Q)

obtain $n::nat$ **where** $n = nat Q$ **by** *auto*

with ass **have** $n: Q = int n$ **by** *auto*

have $(n > 0 \wedge is\text{-}qfN (P*int n) \exists \wedge ?A(int n)) \implies False$ (**is** $?B n \implies False$)

proof (*induct n rule: less-induct*)

case (*less n*)

hence $IH: \forall m. m < n \wedge ?B m \implies False$

and $Bn: ?B n$ **by** *auto*

show $False$

proof (*cases*)

assume $odd: (int n) \in zOdd$

from Bn ass **have** $zprime P \wedge int n > 0 \wedge is\text{-}qfN (P*int n) \exists \wedge \neg is\text{-}qfN P \exists$

by *simp*

hence $\exists R. zprime R \wedge R \text{ dvd } int n \wedge \neg is\text{-}qfN R \exists$

by (*rule qfN-primedivisor-not*)

then obtain R **where** $R: zprime R \wedge R \text{ dvd } int n \wedge \neg is\text{-}qfN R \exists$ **by** *auto*

moreover with odd **have** $R \in zOdd$

proof –

```

    from R obtain U where int n = R*U by (auto simp add: dvd-def)
    with odd show ?thesis by (auto dest: odd-mult-odd-prop)
  qed
  moreover from Bn have ?A (int n) by simp
  ultimately show False by auto
next
  assume ¬ (int n) ∈ zOdd
  hence even: int n ∈ zEven by (rule not-odd-impl-even)
  hence (int n)*P ∈ zEven by (rule even-times-either)
  with Bn have P*int n ∈ zEven ∧ is-qn (P*int n) 3 by (simp add: mult-ac)
  hence ∃ B. P*(int n) = 4*B ∧ is-qn B 3 by (simp only: qf3-even-general)
  then obtain B where B: P*(int n) = 4*B ∧ is-qn B 3 by auto
  hence 2^2 dvd (int n)*P by (simp add: mult-ac)
  moreover have ¬ 2 dvd P
  proof (rule ccontr, simp)
    assume 2 dvd P
    with ass have P ∈ zOdd ∧ P ∈ zEven by (simp add: dvd-def zEven-def)
    thus False by (simp only: even-odd-conj)
  qed
  moreover have zprime 2 by (rule zprime-2)
  ultimately have 2^2 dvd int n
    by (rule-tac p=2 in zprime-power-zdvd-cancel-right)
  then obtain im::int where int n = 4*im by (auto simp add: dvd-def)
  moreover obtain m::nat where m = nat im by auto
  ultimately have m: n = 4*m by arith
  with B have is-qn (P*int m) 3 by (auto simp add: int-mult)
  moreover from m Bn have m > 0 by auto
  moreover from m Bn have ?A (int m)
    by (auto simp add: int-mult)
  ultimately have Bm: ?B m by simp
  from Bn m have m < n by arith
  with IH Bm show False by auto
  qed
  qed
  with ass ass2 n show False by auto
  qed

```

lemma *qf3-oddprimedivisor*:

```

[[ zprime P; P ∈ zOdd; zgcd a b=1; P dvd (a^2+3*b^2) ]]
⇒ is-qn P 3

```

proof(*induct P arbitrary:a b rule:infinite-descent0-measure[where V=λP. nat|P|]*)

case (0 x)

moreover hence $x = 0$ by *arith*

ultimately show ?case by (simp add: zprime-def)

next

case (*smaller x*)

then obtain a b where $abx: zprime x \wedge x \in zOdd \wedge zgcd a b=1$

$\wedge x \text{ dvd } (a^2+3*b^2) \wedge \neg is-qn x 3$ by *auto*

then obtain M where $M: a^2+3*b^2 = x*M$ by (auto simp add: dvd-def)

let ?A = $a^2 + 3*b^2$

from abx have $x0: x > 0 \wedge x \in zOdd$ by (simp add: zprime-def)

then obtain m where $2*|a-m*x| < x$ by (auto dest: best-odd-division-abs)

```

then obtain c where cm: c = a - m*x ∧ 2*|c| < x by auto
from x0 obtain n where 2*|b - n*x| < x by (auto dest: best-odd-division-abs)
then obtain d where dn: d = b - n*x ∧ 2*|d| < x by auto
let ?C = c^2 + 3*d^2
have C3: is-qn ?C 3 by (unfold is-qn-def, auto)
have C0: ?C > 0
proof -
  have hlp: (3::int) ≥ 1 by simp
  with C3 have ?C ≥ 0 by (simp only: qn-pos)
  hence ?C = 0 ∨ ?C > 0 by auto
  moreover
  { assume ?C = 0
    with hlp have c=0 ∧ d=0 by (rule qn-zero)
    with cm dn have a = m*x ∧ b = n*x by simp
    hence x dvd a ∧ x dvd b by simp
    hence x dvd zgcd a b by (simp add: zgcd-greatest-iff)
    with abx have False by (auto simp add: zprime-def) }
  ultimately show ?thesis by blast
qed
have x dvd ?C
proof
  have ?C = |c|^2 + 3*|d|^2 by (simp only: power2-abs)
  also with cm dn have ... = (a - m*x)^2 + 3*(b - n*x)^2 by simp
  also have ... =
    a^2 - 2*a*(m*x) + (m*x)^2 + 3*(b^2 - 2*b*(n*x) + (n*x)^2)
    by (simp only: zdiff-power2)
  also with abx M have ... =
    x*M - x*(2*a*m + 3*2*b*n) + x^2*(m^2 + 3*n^2)
    by (simp only: power-mult-distrib zadd-zmult-distrib2 mult-ac, auto)
  finally show ?C = x*(M - (2*a*m + 3*2*b*n) + x*(m^2 + 3*n^2))
    by (simp add: power2-eq-square zadd-zmult-distrib2 zdiff-zmult-distrib2)
qed
then obtain y where y: ?C = x*y by (auto simp add: dvd-def)
have yx: y < x
proof (rule ccontr)
  assume ¬ y < x hence xy: x - y ≤ 0 by simp
  have hlp: 2*|c| ≥ 0 ∧ 2*|d| ≥ 0 ∧ (3::nat) > 0 by simp
  from y have 4*x*y = 2^2*c^2 + 3*2^2*d^2 by simp
  hence 4*x*y = (2*|c|)^2 + 3*(2*|d|)^2
    by (auto simp add: power2-abs power-mult-distrib)
  with cm dn hlp have 4*x*y < x^2 + 3*(2*|d|)^2
    and (3::int) > 0 ∧ (2*|d|)^2 < x^2
    by (auto simp add: power-strict-mono)
  hence x*4*y < x^2 + 3*x^2 by (auto)
  also have ... = x*4*x by (simp add: power2-eq-square)
  finally have contr: (x - y)*(4*x) > 0 by (auto simp add: zdiff-zmult-distrib2)
  show False
proof (cases)
  assume x - y = 0 with contr show False by auto
next
  assume ¬ x - y = 0 with xy have x - y < 0 by simp
  moreover from x0 have 4*x > 0 by simp

```

```

    ultimately have  $4*x*(x-y) < 4*x*0$  by (simp only: zmult-zless-mono2)
    with contr show False by auto
  qed
qed
have  $y0: y > 0$ 
proof (rule ccontr)
  assume  $\neg y > 0$ 
  hence  $y \leq 0$  by simp
  moreover have  $y \neq 0$ 
  proof (rule ccontr)
    assume  $\neg y \neq 0$  hence  $y=0$  by simp
    with  $y$  and  $C0$  show False by auto
  qed
  ultimately have  $y < 0$  by simp
  with  $x0$  have  $x*y < x*0$  by (simp only: zmult-zless-mono2)
  with  $C0$   $y$  show False by simp
qed
let  $?g = \text{zgcd } c \ d$ 
have  $c \neq 0 \vee d \neq 0$ 
proof (rule ccontr)
  assume  $\neg (c \neq 0 \vee d \neq 0)$  hence  $c=0 \wedge d=0$  by simp
  with  $C0$  show False by simp
qed
then obtain  $e \ f$  where  $ef: c = ?g*e \wedge d = ?g * f \wedge \text{zgcd } e \ f = 1$ 
  by (frule-tac  $a=c$  in make-zrelprime, auto)
have  $g2\text{nonzero}: ?g^2 \neq 0$ 
proof (rule ccontr, simp)
  assume  $c = 0 \wedge d = 0$ 
  with  $C0$  show False by simp
qed
let  $?E = e^2 + 3*f^2$ 
have  $E3: \text{is-qn } ?E \ 3$  by (unfold is-qn-def, auto)
have  $CgE: ?C = ?g^2 * ?E$ 
proof -
  have  $?g^2 * ?E = (?g*e)^2 + 3*(?g*f)^2$ 
  by (simp add: zadd-zmult-distrib2 power-mult-distrib)
  with  $ef$  show ?thesis by simp
qed
hence  $?g^2 \text{ dvd } ?C$  by (simp add: dvd-def)
with  $y$  have  $g2\text{dvdxy}: ?g^2 \text{ dvd } y*x$  by (simp add: mult-ac)
moreover have  $\text{zgcd } x \ (?g^2) = 1$ 
proof -
  let  $?h = \text{zgcd } ?g \ x$ 
  have  $?h \text{ dvd } ?g$  and  $?g \text{ dvd } c$  by auto
  hence  $?h \text{ dvd } c$  by (rule dvd-trans)
  have  $?h \text{ dvd } ?g$  and  $?g \text{ dvd } d$  by auto
  hence  $?h \text{ dvd } d$  by (rule dvd-trans)
  have  $?h \text{ dvd } x$  by simp
  hence  $?h \text{ dvd } m*x$  by (rule dvd-mult)
  with  $\langle ?h \text{ dvd } c \rangle$  have  $?h \text{ dvd } c+m*x$  by (rule dvd-add)
  with  $cm$  have  $?h \text{ dvd } a$  by simp
  from  $\langle ?h \text{ dvd } x \rangle$  have  $?h \text{ dvd } n*x$  by (rule dvd-mult)

```

```

with ⟨?h dvd d⟩ have ?h dvd d+n*x by (rule dvd-add)
with dn have ?h dvd b by simp
with ⟨?h dvd a⟩ have ?h dvd zgcd a b by (simp add: zgcd-greatest-iff)
with abx have ?h dvd 1 by simp
hence ?h = 1 by (simp add: zgcd-geq-zero)
hence zgcd (?g^2) x = 1 by (rule zgcd-1-power-left-distrib)
thus ?thesis by (simp only: zgcd-commute)
qed
ultimately have ?g^2 dvd y by (auto dest: zrelprime-zdvd-zmult)
then obtain w where w: y = ?g^2 * w by (auto simp add: dvd-def)
with CgE y g2nonzero have Ewx: ?E = x*w by auto
have w>0
proof (rule ccontr)
  assume ¬ w>0 hence w ≤ 0 by auto
  hence w=0 ∨ w<0 by auto
  moreover
  { assume w=0 with w y0 have False by auto }
  moreover
  { assume wneg: w<0
    have ?g^2 ≥ 0 by (rule zero-le-power2)
    with g2nonzero have ?g^2 > 0 by arith
    with wneg have ?g^2*w < ?g^2*0 by (simp only: zmult-zless-mono2)
    with w y0 have False by auto }
  ultimately show False by blast
qed
have w-le-y: w ≤ y
proof (rule ccontr)
  assume ¬ w ≤ y
  hence wy: w > y by simp
  have ?g^2 = 1 ∨ ?g^2 > 1
  proof -
    have ?g^2 ≥ 0 by (rule zero-le-power2)
    hence ?g^2 = 0 ∨ ?g^2 > 0 by auto
    with g2nonzero show ?thesis by arith
  qed
  moreover
  { assume ?g^2 = 1 with w wy have False by simp }
  moreover
  { assume g1: ?g^2 > 1
    with ⟨w>0⟩ have w*1 < w*?g^2 by (auto dest: zmult-zless-mono2)
    with w have w < y by (simp add: zmult-1 mult-ac)
    with wy have False by auto }
  ultimately show False by blast
qed
from Ewx E3 abx ⟨w>0⟩ have
  zprime x ∧ x ∈ zOdd ∧ w > 0 ∧ is-qn (x*w) 3 ∧ ¬ is-qn x 3 by simp
then obtain z where z: zprime z ∧ z ∈ zOdd ∧ z dvd w ∧ ¬ is-qn z 3
  by (frule-tac P=x in qf3-oddprimedivisor-not, auto)
from Ewx have w dvd ?E by simp
with z have z dvd ?E by (auto dest: dvd-trans)
with z ef have zprime z ∧ z ∈ zOdd ∧ zgcd e f = 1 ∧ z dvd ?E ∧ ¬ is-qn z 3
  by auto

```

```

moreover have  $\text{nat}|z| < \text{nat}|x|$ 
proof -
  have  $z \leq w$ 
  proof (rule ccontr)
    assume  $\neg z \leq w$  hence  $w < z$  by auto
    with  $\langle w > 0 \rangle$  have  $\neg z \text{ dvd } w$  by (rule zdvd-not-zless)
    with  $z$  show False by simp
  qed
  with w-le-y yx have  $z < x$  by simp
  with  $z$  have  $|z| < |x|$  by (simp add: zprime-def)
  thus ?thesis by auto
qed
ultimately show ?case by auto
qed

lemma qf3-cube-prime-impl-cube-form:
  assumes ab-relprime:  $\text{zgcd } a \ b = 1$  and abP:  $P^3 = a^2 + 3*b^2$ 
  and P: zprime  $P \wedge P \in \text{zOdd}$ 
  shows is-cube-form  $a \ b$ 
proof -
  from abP have qfP3: is-qfN  $(P^3) \ 3$  by (auto simp only: is-qfN-def)
  have PvdP3:  $P \text{ dvd } P^3$  by (simp add: nat-number)
  with abP ab-relprime  $P$  have qfP: is-qfN  $P \ 3$  by (simp only: qf3-oddprimedivisor)
  then obtain  $p \ q$  where pq:  $P = p^2 + 3*q^2$  by (auto simp only: is-qfN-def)
  with  $P \ abP \ ab-relprime$  have zprime  $(p^2 + 3*q^2) \wedge (3::\text{int}) > 1$ 
     $\wedge (p^2 + 3*q^2)^3 = a^2 + 3*b^2 \wedge \text{zgcd } a \ b = 1$  by auto
  hence ab:  $|a| = |p^3 - 3*3*p*q^2| \wedge |b| = |3*p^2*q - 3*q^3|$ 
    by (rule qfN-cube-prime)
  hence a:  $a = p^3 - 9*p*q^2 \vee a = -(p^3) + 9*p*q^2$  by arith
  from ab have b:  $b = 3*p^2*q - 3*q^3 \vee b = -(3*p^2*q) + 3*q^3$  by arith
  obtain  $r \ s$  where r:  $r = -p$  and s:  $s = -q$  by simp
  show ?thesis
proof (cases)
  assume a1:  $a = p^3 - 9*p*q^2$ 
  show ?thesis
  proof (cases)
  assume b1:  $b = 3*p^2*q - 3*q^3$ 
  with a1 show ?thesis by (unfold is-cube-form-def, auto)
  next
  assume  $\neg b = 3*p^2*q - 3*q^3$ 
  with  $b$  have  $b = -3*p^2*q + 3*q^3$  by simp
  with  $s$  have  $b = 3*p^2*s - 3*s^3$  by (simp add: power3-minus)
  moreover from a1  $s$  have  $a = p^3 - 9*p*s^2$  by (simp add: power2-minus)
  ultimately show ?thesis by (unfold is-cube-form-def, auto)
  qed
next
  assume  $\neg a = p^3 - 9*p*q^2$ 
  with  $a$  have  $a = -(p^3) + 9*p*q^2$  by simp
  with  $r$  have ar:  $a = r^3 - 9*r*q^2$  by (simp add: power3-minus)
  show ?thesis
proof (cases)
  assume b1:  $b = 3*p^2*q - 3*q^3$ 

```

with r have $b = 3*r^2*q - 3*q^3$ by (simp add: power2-minus)
with ar show ?thesis by (unfold is-cube-form-def, auto)
next
assume $\neg b = 3*p^2*q - 3*q^3$
with b have $b = -3*p^2*q + 3*q^3$ by simp
with $r s$ have $b = 3*r^2*s - 3*s^3$
by (simp add: power2-minus power3-minus)
moreover from $ar s$ have $a = r^3 - 9*r*s^2$ by (simp add: power2-minus)
ultimately show ?thesis by (unfold is-cube-form-def, auto)
qed
qed
qed

**lemma cube-form-mult: \llbracket is-cube-form $a b$; is-cube-form $c d$; $|e| = 1 \rrbracket$
 \implies is-cube-form $(a*c + e*3*b*d) (a*d - e*b*c)$**

proof -

assume ab : is-cube-form $a b$ and $c-d$: is-cube-form $c d$ and e : $|e| = 1$
from ab obtain $p q$ where pq : $a = p^3 - 9*p*q^2 \wedge b = 3*p^2*q - 3*q^3$
by (auto simp only: is-cube-form-def)
from $c-d$ obtain $r s$ where rs : $c = r^3 - 9*r*s^2 \wedge d = 3*r^2*s - 3*s^3$
by (auto simp only: is-cube-form-def)

let $?t = p*r + e*3*q*s$

let $?u = p*s - e*r*q$

have $e2$: $e^2 = 1$

proof -

from e have $e = 1 \vee e = -1$ by simp

moreover

{ assume $e = 1$ hence ?thesis by auto }

moreover

{ assume $e = -1$ hence ?thesis by (simp add: power2-minus) }

ultimately show ?thesis by blast

qed

hence $e*e^2 = e$ by simp

hence $e3$: $e*1 = e^3$ by (simp only: cube-square)

have $a*c + e*3*b*d = ?t^3 - 9*?t*?u^2$

proof -

**have $?t^3 - 9*?t*?u^2 = p^3*r^3 + e*9*p^2*q*r^2*s + e^2*27*p*q^2*r*s^2$
 $+ e^3*27*q^3*s^3 - 9*p*p^2*r*s^2 + e*18*p^2*q*r^2*s - e^2*9*p*q^2*(r*r^2)$
 $- e*27*p^2*q*(s*s^2) + e^2*54*p*q^2*r*s^2 - e*e^2*27*(q*q^2)*r^2*s$**

by (simp add: nat-number ring-simps)

also with $e2 e3$ have ... =

$p^3*r^3 + e*27*p^2*q*r^2*s + 81*p*q^2*r*s^2 + e*27*q^3*s^3$
 $- 9*p^3*r*s^2 - 9*p*q^2*r^3 - e*27*p^2*q*s^3 - e*27*q^3*r^2*s$

by (simp add: cube-square zmult-1)

also with $pq rs$ have ... = $a*c + e*3*b*d$

by (simp only: zdiff-zmult-distrib zdiff-zmult-distrib2 mult-ac)

finally show ?thesis by auto

qed

moreover have $a*d - e*b*c = 3*?t^2*?u - 3*?u^3$

proof -

have $3*?t^2*?u - 3*?u^3 =$

$3*(p*p^2)*r^2*s - e*3*p^2*q*(r*r^2) + e*18*p^2*q*r*s^2$

```

    - e^2*18*p*q^2*r^2*s + e^2*27*p*q^2*(s*s^2) - e*e^2*27*(q*q^2)*r*s^2
    - 3*p^3*s^3 + e*9*p^2*q*r*s^2 - e^2*9*p*q^2*r^2*s + e^3*3*r^3*q^3
    by (simp add: nat-number ring-simps)
  also with e2 e3 have ... = 3*p^3*r^2*s - e*3*p^2*q*r^3 + e*18*p^2*q*r*s^2
    - 18*p*q^2*r^2*s + 27*p*q^2*s^3 - e*27*q^3*r*s^2 - 3*p^3*s^3
    + e*9*p^2*q*r*s^2 - 9*p*q^2*r^2*s + e*3*r^3*q^3
    by (simp add: cube-square zmult-1)
  also with pq rs have ... = a*d - e*b*c
    by (simp only: zdiff-zmult-distrib zdiff-zmult-distrib2 mult-ac)
  finally show ?thesis by auto
qed
ultimately show ?thesis by (auto simp only: is-cube-form-def)
qed

```

lemma *qf3-cube-primelist-impl-cube-form*: $\llbracket \text{primel } ps; \text{int } (\text{prod } ps) \in \text{zOdd} \rrbracket \implies$
 $(!! a b. \text{zgcd } a b = 1 \implies a^2 + 3*b^2 = (\text{int } (\text{prod } ps))^3 \implies \text{is-cube-form } a b)$

proof (*induct ps*)

case Nil hence *ab1*: $a^2 + 3*b^2 = 1$ by *simp*

have *b0*: $b=0$

proof (*rule ccontr*)

assume $b \neq 0$

hence $b^2 > 0$ by (*simp add: zero-less-power2*)

hence $3*b^2 > 1$ by *arith*

with *ab1* have $a^2 < 0$ by *arith*

moreover have $a^2 \geq 0$ by (*rule zero-le-power2*)

ultimately show *False* by *auto*

qed

with *ab1* have *a1*: $(a=1 \vee a=-1)$ by (*auto simp add: power2-eq-square zmult-eq-1-iff*)

then obtain *p* and *q* where $p=a$ and $q=(0::\text{int})$ by *simp*

with *a1* and *b0* have $a = p^3 - 9*p*q^2 \wedge b = 3*p^2*q - 3*q^3$ by *auto*

thus *is-cube-form a b* by (*auto simp only: is-cube-form-def*)

next

case (Cons p ps) hence *ass*: $\text{zgcd } a b = 1 \wedge \text{int } (\text{prod } (p\#ps)) \in \text{zOdd}$

$\wedge a^2 + 3*b^2 = \text{int } (\text{prod } (p\#ps))^3 \wedge \text{primel } ps \wedge \text{zprime } (\text{int } p)$

and *IH*: $!! u v. \text{zgcd } u v = 1 \wedge u^2 + 3*v^2 = \text{int } (\text{prod } ps)^3$

$\wedge \text{int } (\text{prod } ps) \in \text{zOdd} \implies \text{is-cube-form } u v$

by (*auto simp add: primel-def prime-impl-zprime-int*)

let *?w* = $\text{int } (\text{prod } (p\#ps))$

let *?X* = $\text{int } (\text{prod } ps)$

let *?p* = $\text{int } p$

have *ge3-1*: $(3::\text{int}) \geq 1$ by *auto*

have *pw*: $?w = ?p * ?X \wedge ?p \in \text{zOdd} \wedge ?X \in \text{zOdd}$

proof (*safe*)

have $\text{prod } (p\#ps) = p * \text{prod } ps$ by *simp*

thus *wpx*: $?w = ?p * ?X$ by (*auto simp only: zmult-int*)

with *ass* show $?p \in \text{zOdd}$ by (*auto dest: odd-mult-odd-prop*)

from *wpx* have $?w = ?X * ?p$ by *simp*

with *ass* show $?X \in \text{zOdd}$ by (*auto dest: odd-mult-odd-prop*)

qed

have *is-qfN ?p 3*

proof -

from *ass* have $a^2 + 3*b^2 = (?p * ?X)^3$ by (*simp add: zmult-int*)

hence $?p \text{ dvd } a^2 + 3*b^2$ by (simp add: nat-number ring-simps)
 moreover from *ass* have $zprime ?p$ and $zgcd a b = 1$ by *simp-all*
 moreover from *pw* have $?p \in zOdd$ by *simp*
 ultimately show *?thesis* by (simp only: qf3-oddprimedivisor)
 qed
 then obtain $\alpha \beta$ where *alphabet*: $?p = \alpha^2 + 3*\beta^2$
 by (auto simp add: is-qfN-def)
 have $\alpha \neq 0$
 proof (rule *ccontr*, *simp*)
 assume $\alpha = 0$ with *alphabet* have $3 \text{ dvd } ?p$ by *auto*
 with *pw* have $w3: 3 \text{ dvd } ?w$ by (simp only: dvd-mult2)
 then obtain v where $?w = 3*v$ by (auto simp add: dvd-def)
 with *ass* have *vab*: $27*v^3 = a^2 + 3*b^2$ by (simp add: power-mult-distrib)
 hence $a^2 = 3*(9*v^3 - b^2)$ by *auto*
 hence $3 \text{ dvd } a^2$ by (unfold dvd-def, blast)
 moreover have $zprime 3$ by (rule *zprime-3*)
 ultimately have *a3*: $3 \text{ dvd } a$ by (rule-tac $p=3$ in *zprime-zdvd-power*)
 then obtain c where $c: a = 3*c$ by (auto simp add: dvd-def)
 with *vab* have $27*v^3 = 9*c^2 + 3*b^2$ by (simp add: power-mult-distrib)
 hence $b^2 = 3*(3*v^3 - c^2)$ by *auto*
 hence $3 \text{ dvd } b^2$ by (unfold dvd-def, blast)
 moreover have $zprime 3$ by (rule *zprime-3*)
 ultimately have $3 \text{ dvd } b$ by (rule-tac $p=3$ in *zprime-zdvd-power*)
 with *a3* have $3 \text{ dvd } zgcd a b$ by (simp add: zgcd-greatest-iff)
 with *ass* show *False* by *simp*
 qed
 moreover from *alphabet* *pw* *ass* have
 $zprime (\alpha^2 + 3*\beta^2) \wedge \alpha^2 + 3*\beta^2 \in zOdd \wedge (3::int) \geq 1$ by *auto*
 ultimately obtain $c d$ where *cdp*:
 $(\alpha^2 + 3*\beta^2)^3 = c^2 + 3*d^2 \wedge zgcd c (3*d) = 1$
 by (blast dest: qfN-oddprime-cube)
 with *ass* *pw* *alphabet* have $\exists u v. a^2 + 3*b^2 = (u^2 + 3*v^2)*(c^2 + 3*d^2)$
 $\wedge zgcd u v = 1 \wedge (\exists e. a = c*u + e*3*d*v \wedge b = c*v - e*d*u \wedge |e| = 1)$
 by (rule-tac $A=?w$ and $n=3$ in *qfN-power-div-prime*, *auto*)
 then obtain $u v e$ where *uve*: $a^2 + 3*b^2 = (u^2 + 3*v^2)*(c^2 + 3*d^2)$
 $\wedge zgcd u v = 1 \wedge a = c*u + e*3*d*v \wedge b = c*v - e*d*u \wedge |e| = 1$ by *blast*
 moreover have *is-cube-form* $u v$
 proof -
 have *uvX*: $u^2 + 3*v^2 = ?X^3$
 proof -
 from *ass* have $p0: ?p \neq 0$ by (simp add: *zprime-def*)
 from *pw* have $?p^3 * ?X^3 = ?w^3$ by (simp add: power-mult-distrib)
 also with *ass* have $\dots = a^2 + 3*b^2$ by *simp*
 also with *uve* have $\dots = (u^2 + 3*v^2)*(c^2 + 3*d^2)$ by *auto*
 also with *cdp* *alphabet* have $\dots = ?p^3 * (u^2 + 3*v^2)$ by (simp only: *mult-ac*)
 finally have $?p^3 * (u^2 + 3*v^2 - ?X^3) = 0$ by *auto*
 with $p0$ show *?thesis* by *auto*
 qed
 with *pw* *IH* *uve* show *?thesis* by *simp*
 qed
 moreover have *is-cube-form* $c d$
 proof -

```

have zgcd c d = 1
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix h::int assume h dvd c and h dvd d and h: zprime h
  hence h dvd c*u + d*(e*3*v)  $\wedge$  h dvd c*v - d*(e*u) by simp
  with wve have h dvd a  $\wedge$  h dvd b by (auto simp only: mult-ac)
  with ass h show False by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed
with pw cdp ass alphabeta show ?thesis
  by (rule-tac P=?p in qf3-cube-prime-impl-cube-form, auto)
qed
ultimately show is-cube-form a b by (simp only: cube-form-mult)
qed

```

lemma qf3-cube-impl-cube-form:

```

assumes ass: zgcd a b=1  $\wedge$  a2 + 3*b2 = w3  $\wedge$  w  $\in$  zOdd
shows is-cube-form a b

```

```

proof -
have  $\exists$  ps. primel ps  $\wedge$  int (prod ps) = w
proof -
  have wpos: w  $\geq$  1
  proof -
    have b2  $\geq$  0 by (rule zero-le-power2)
    hence 3*b2  $\geq$  0 by arith
    moreover have a2  $\geq$  0 by (rule zero-le-power2)
    ultimately have a2 + 3*b2  $\geq$  0 by arith
    with ass have w3pos: w3  $\geq$  0 by simp
    have w $\geq$ 0
    proof (rule ccontr)
      assume  $\neg$  w $\geq$ 0 hence -w > 0 by auto
      hence (-1 * w)3 > 0 by (auto simp only: zero-less-power)
      hence (-1)3 * (w3) > 0 by (simp only: power-mult-distrib)
      hence w3 < 0 by (simp add: neg-one-odd-power)
      with w3pos show False by auto
    qed
    moreover have w  $\neq$  0
    proof (rule ccontr)
      assume  $\neg$ w $\neq$ 0 with ass have 0  $\in$  zOdd by simp
      moreover have 0  $\in$  zEven by (simp add: zEven-def)
      ultimately show False by (auto simp add: odd-iff-not-even)
    qed
    ultimately show ?thesis by (auto)
  qed
hence w=1  $\vee$  Suc 0 < nat w by auto
moreover
{ assume w=1
  hence primel []  $\wedge$  int (prod []) = w by (auto simp add: primel-def)
  hence ?thesis by (simp only: exI) }
moreover
{ assume Suc 0 < nat w
  hence  $\exists$  l. primel l  $\wedge$  prod l = nat w by (rule factor-exists)
  then obtain ps where ps: primel ps  $\wedge$  prod ps = nat w by auto
  with wpos have ?thesis by auto }

```

ultimately show *?thesis* by *blast*
qed
 with *ass* show *?thesis* by (auto dest: *qf3-cube-primelist-impl-cube-form*)
qed

3.6 Existence ($N = 3$)

This part contains the proof that all prime numbers $\equiv 1 \pmod{6}$ can be written as $x^2 + 3y^2$.

First show $\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right)$, where p is an odd prime.

lemma *Legendre-zmult*: $\llbracket p > 2; \text{zprime } p \rrbracket$
 $\implies (\text{Legendre } (a*b) \text{ } p) = (\text{Legendre } a \text{ } p) * (\text{Legendre } b \text{ } p)$
proof –
 assume *p2*: $p > 2$ and *prp*: *zprime* p
 let *?p12* = $\text{nat}(((p) - 1) \text{ div } 2)$
 let *?Labp* = *Legendre* $(a*b) \text{ } p$
 let *?Lap* = *Legendre* $a \text{ } p$
 let *?Lbp* = *Legendre* $b \text{ } p$
 from *p2 prp* have $[?Labp = (a*b) ^{?p12}] \pmod{p}$
 by (*simp only*: *Euler-Criterion*)
 hence $[a ^{?p12} * b ^{?p12} = ?Labp] \pmod{p}$
 by (*simp only*: *power-mult-distrib zcong-sym*)
 moreover from *p2 prp* have $[?Lap * ?Lbp = a ^{?p12} * b ^{?p12}] \pmod{p}$
 by (*simp only*: *Euler-Criterion zcong-zmult*)
 ultimately have $[?Lap * ?Lbp = ?Labp] \pmod{p}$
 by (*rule-tac* $b = a ^{?p12} * b ^{?p12}$ in *zcong-trans*)
 then obtain *k* where *k*: $?Labp = (?Lap * ?Lbp) + p * k$
 by (*auto simp add*: *zcong-iff-lin*)
 have $k = 0$
proof (*rule ccontr*)
 assume $k \neq 0$ hence $|k| = 1 \vee |k| > 1$ by *arith*
 moreover
 { assume $|k| = 1$
 with *p2* have $|k| * p > 2$ by *auto* }
 moreover
 { assume *k1*: $|k| > 1$
 with *p2* have $|k| * 2 < |k| * p$
 by (*simp only*: *zmult-zless-mono2*)
 with *k1* have $|k| * p > 2$ by *auto* }
 ultimately have $|k| * p > 2$ by *auto*
 moreover from *p2* have $|p| = p$ by *auto*
 ultimately have $|k * p| > 2$ by (*auto simp only*: *abs-mult*)
 moreover from *k* have $?Labp - ?Lap * ?Lbp = k * p$ by *auto*
 ultimately have $|?Labp - ?Lap * ?Lbp| > 2$ by *auto*
 moreover have $?Labp = 1 \vee ?Labp = 0 \vee ?Labp = -1$
 by (*simp add*: *Legendre-def*)
 moreover have $?Lap * ?Lbp = 1 \vee ?Lap * ?Lbp = 0 \vee ?Lap * ?Lbp = -1$
 by (*auto simp add*: *Legendre-def*)
 ultimately show *False* by *auto*
qed
 with *k* show *?thesis* by *auto*

qed

Now show $\left(\frac{-3}{p}\right) = +1$ for primes $p \equiv 1 \pmod{6}$.

lemma *Legendre-1mod6*: $zprime (6*m+1) \implies Legendre (-3) (6*m+1) = 1$

proof –

let $?p = 6*m+1$

let $?L = Legendre (-3) ?p$

let $?L1 = Legendre (-1) ?p$

let $?L3 = Legendre 3 ?p$

assume p : $zprime ?p$

have *neg1cube*: $(-1::int)^3 = -1$ by (*simp add: power3-minus*)

have *m1*: $m \geq 1$

proof (*rule ccontr*)

assume $\neg m \geq 1$ hence $m \leq 0$ by *simp*

with p show *False* by (*auto simp add: zprime-def*)

qed

hence *pn3*: $?p \neq 3$ and *p2*: $?p > 2$ by *auto*

with p have $?L = (Legendre (-1) ?p) * (Legendre 3 ?p)$

by (*frule-tac a=-1 and b=3 in Legendre-zmult, auto*)

moreover have $[Legendre (-1) ?p = (-1)^{\text{nat } m}] \pmod{?p}$

proof –

from p *p2* have $[?L1 = (-1)^{\text{nat}((?p) - 1 \text{ div } 2)}] \pmod{?p}$

by (*simp only: Euler-Criterion*)

moreover have $\text{nat}((?p - 1) \text{ div } 2) = 3 * \text{nat } m$

proof –

have $(?p - 1) \text{ div } 2 = 3 * m$ by *auto*

hence $\text{nat}((?p - 1) \text{ div } 2) = \text{nat} (3 * m)$ by *simp*

moreover have $(3::int) \geq 0$ by *simp*

ultimately show *?thesis* by (*simp add: nat-mult-distrib*)

qed

moreover with *neg1cube* have $(-1::int)^{(3*\text{nat } m)} = (-1)^{\text{nat } m}$

by (*simp only: power-mult*)

ultimately show *?thesis* by *auto*

qed

moreover have $?L3 = (-1)^{\text{nat } m}$

proof –

have $?L3 * (Legendre ?p 3) = (-1)^{\text{nat } m}$

proof –

have $3 \in zOdd \wedge ?p \in zOdd$ by (*unfold zOdd-def, auto*)

with p *pn3* have $?L3 * (Legendre ?p 3) = (-1::int)^{(3*\text{nat } m)}$

by (*simp add: zprime-3 Quadratic-Reciprocity nat-mult-distrib*)

with *neg1cube* show *?thesis* by (*simp add: power-mult*)

qed

moreover have $Legendre ?p 3 = 1$

proof –

have $[1^2 = ?p] \pmod{3}$ by (*unfold zcong-def dvd-def, auto*)

hence $QuadRes 3 ?p$ by (*unfold QuadRes-def, blast*)

moreover have $\neg [?p = 0] \pmod{3}$

proof (*rule ccontr, simp*)

assume $[?p = 0] \pmod{3}$

hence $3 \text{ dvd } ?p$ by (*simp add: zcong-def*)

moreover have $3 \text{ dvd } 6*m$ by (*auto simp add: dvd-def*)

```

    ultimately have  $3 \text{ dvd } ?p - 6 * m$  by (simp only: dvd-diff)
    hence  $(3 :: \text{int}) \text{ dvd } 1$  by simp
    thus False by auto
  qed
  ultimately show ?thesis by (unfold Legendre-def, auto)
  qed
  ultimately show ?thesis by auto
  qed
  ultimately have  $[?L = (-1) ^{(\text{nat } m) * (-1) ^{(\text{nat } m)}}] \pmod{?p}$ 
    by (auto dest: zcong-scalar)
  hence  $[?L = (-1) ^{(\text{nat } m) + (\text{nat } m)}] \pmod{?p}$  by (simp only: power-add)
  moreover have  $(\text{nat } m) + (\text{nat } m) = 2 * (\text{nat } m)$  by auto
  ultimately have  $[?L = (-1) ^{(2 * (\text{nat } m))}] \pmod{?p}$  by simp
  hence  $[?L = ((-1) ^2) ^{(\text{nat } m)}] \pmod{?p}$  by (simp only: power-mult)
  hence  $[1 = ?L] \pmod{?p}$  by (auto simp add: zcong-sym)
  hence  $?p \text{ dvd } 1 - ?L$  by (simp only: zcong-def)
  moreover have  $?L = -1 \vee ?L = 0 \vee ?L = 1$  by (simp add: Legendre-def)
  ultimately have  $?p \text{ dvd } 2 \vee ?p \text{ dvd } 1 \vee ?L = 1$  by auto
  moreover
  { assume  $?p \text{ dvd } 2 \vee ?p \text{ dvd } 1$ 
    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 $x^2 + 3y^2$.

```

lemma qf3-prime-exists:  $zprime (6 * m + 1) \implies \exists x y. 6 * m + 1 = x^2 + 3 * y^2$ 
proof -
  let  $?p = 6 * m + 1$ 
  assume  $p: zprime ?p$ 
  hence Legendre  $(-3) ?p = 1$  by (rule Legendre-1mod6)
  moreover
  { assume  $\neg QuadRes ?p (-3)$ 
    hence Legendre  $(-3) ?p \neq 1$  by (unfold Legendre-def, auto) }
  ultimately have  $QuadRes ?p (-3)$  by auto
  then obtain  $s$  where  $s: [s^2 = -3] \pmod{?p}$  by (auto simp add: QuadRes-def)
  hence  $?p \text{ dvd } s^2 - (-3 :: \text{int})$  by (unfold zcong-def, simp)
  moreover have  $s^2 - (-3 :: \text{int}) = s^2 + 3$  by arith
  ultimately have  $?p \text{ dvd } s^2 + 3 * 1^2$  by auto
  moreover have  $zgcd s 1 = 1$  by (unfold zgcd-def, auto)
  moreover have  $?p \in zOdd$ 
  proof -
    have  $?p = 2 * (3 * m) + 1$  by simp
    thus ?thesis by (unfold zOdd-def, blast)
  qed
  moreover from  $p$  have  $zprime ?p$  by simp
  ultimately have  $is\text{-}qfN ?p 3$  by (simp only: qf3-oddprimedivisor)
  thus ?thesis by (unfold is-qfN-def, auto)
  qed
end

```

4 Fermat's last theorem, case $n = 3$

```
theory Fermat3
imports QuadForm
begin
```

Proof of Fermat's last theorem for the case $n = 3$:

$$\forall x, y, z : x^3 + y^3 = z^3 \implies xyz = 0.$$

```
lemma factor-sum-cubes: (x::int)^3 + y^3 = (x+y)*(x^2 - x*y + y^2)
  by (simp add: nat-number ring-simps)
```

```
lemma two-not-abs-cube: |x^3| = (2::int) \implies False
```

```
proof -
```

```
  assume |x^3| = 2
```

```
  hence x32: |x|^3 = 2 by (simp only: abs-power3-distrib)
```

```
  have |x| ≥ 0 by simp
```

```
  moreover
```

```
  { assume |x| = 0 ∨ |x| = 1 ∨ |x| = 2
```

```
    with x32 have False by (auto simp add: power-0-left) }
```

```
  moreover
```

```
  { assume |x| > 2
```

```
    moreover have (0::int) ≤ 2 and (0::nat) < 3 by auto
```

```
    ultimately have |x|^3 > 2^3 by (simp only: power-strict-mono)
```

```
    with x32 have False by simp }
```

```
  ultimately show False by arith
```

```
qed
```

Shows there exists no solution $v^3 + w^3 = x^3$ with $vwx \neq 0$ and $\gcd(v, w) = 1$ and x even, by constructing a solution with a smaller $|x^3|$.

```
lemma no-rewritten-fermat3:
```

```
  ¬ (∃ v w. v^3 + w^3 = x^3 ∧ v*w*x ≠ 0 ∧ x ∈ zEven ∧ zgcd v w = 1)
```

```
proof (induct x rule: infinite-descent0-measure[where V = λx. nat|x^3|])
```

```
  case (0 x) hence x^3 = 0 by arith
```

```
  hence x=0 by auto
```

```
  thus ?case by auto
```

```
next
```

```
  case (smaller x)
```

```
  then obtain v w where vwx:
```

```
    v^3 + w^3 = x^3 ∧ v*w*x ≠ 0 ∧ x ∈ zEven ∧ zgcd v w = 1 (is ?P v w x)
```

```
  by auto
```

```
  have ∃ α β γ. ?P α β γ ∧ nat|γ^3| < nat|x^3|
```

```
  proof -
```

```
    — obtain coprime p and q such that v = p + q and w = p - q
```

```
    have vwOdd: v ∈ zOdd ∧ w ∈ zOdd
```

```
    proof (rule ccontr, case-tac v ∈ zOdd, simp-all)
```

```
      assume v ∉ zOdd hence ve: v ∈ zEven by (rule not-odd-impl-even)
```

```
      hence v^3 ∈ zEven by (simp only: power-preserves-even)
```

```
      moreover from vwx have x^3 ∈ zEven by (simp only: power-preserves-even)
```

```
      ultimately have (x^3 - v^3) ∈ zEven by (simp only: even-minus-even)
```

```
      moreover from vwx have x^3 - v^3 = w^3 by simp
```

ultimately have $w^3 \in zEven$ by *simp*
 hence $w \in zEven$ by (*simp only: power-preserves-even*)
 with *ve* have $2 \mid v \wedge 2 \mid w$ by (*auto simp add: zEven-def*)
 hence $2 \mid \text{zgcd } v \ w$ by (*simp add: zgcd-greatest-iff*)
 with *vwx* show *False* by *simp*
 next
 assume *vo*: $v \in zOdd$ and $w \notin zOdd$
 hence $w \in zEven$ by (*simp add: not-odd-impl-even*)
 with *vo* have $v^3 \in zOdd$ and $w^3 \in zEven$
 by (*auto simp only: power-preserves-even power-preserves-odd*)
 hence $w^3 + v^3 \in zOdd$ by (*simp only: even-plus-odd*)
 with *vwx* have $x^3 \in zOdd$ by (*simp add: zadd-commute*)
 hence $x \in zOdd$ by (*simp only: power-preserves-odd*)
 with *vwx* show *False* by (*auto simp add: odd-iff-not-even*)
 qed
 hence $v+w \in zEven \wedge v-w \in zEven$ by (*simp add: odd-minus-odd odd-plus-odd*)
 then obtain *p q* where $pq: v+w = 2*p \wedge v-w = 2*q$
 by (*auto simp add: zEven-def*)
 hence *vw*: $v = p+q \wedge w = p-q$ by *auto*
 — show that $x^3 = (2p)(p^2 + 3q^2)$ and that these factors are
 — either coprime (first case), or have 3 as g.c.d. (second case)
 have *vwpq*: $v^3 + w^3 = (2*p)*(p^2 + 3*q^2)$
 proof —
 have $2*(v^3 + w^3) = 2*(v+w)*(v^2 - v*w + w^2)$
 by (*simp only: factor-sum-cubes*)
 also from *pq* have $\dots = 4*p*(v^2 - v*w + w^2)$ by *auto*
 also have $\dots = p*((v+w)^2 + 3*(v-w)^2)$
 by (*simp add: nat-number ring-simps*)
 also with *pq* have $\dots = p*((2*p)^2 + 3*(2*q)^2)$ by *simp*
 also have $\dots = 2*(2*p)*(p^2 + 3*q^2)$ by (*simp add: power-mult-distrib*)
 finally show *?thesis* by *simp*
 qed
 let *?g* = $\text{zgcd } (2*p) (p^2 + 3*q^2)$
 have *g1*: $?g \geq 1$
 proof (*rule ccontr*)
 assume $\neg ?g \geq 1$
 then have $?g < 0 \vee ?g = 0$ unfolding *not-le* by *arith*
 moreover have $?g \geq 0$ by (*rule zgcd-geq-zero*)
 ultimately have $?g = 0$ by *arith*
 hence $p = 0$ by *simp*
 with *vwpq vwx* ($0 < \text{nat}|x^3|$) show *False* by *auto*
 qed
 have *gOdd*: $\neg 2 \mid ?g$
 proof (*rule ccontr, simp*)
 assume $2 \mid ?g$
 hence $2 \mid p^2 + 3*q^2$ by (*simp only: zgcd-greatest-iff*)
 then obtain *k* where $k: p^2 + 3*q^2 = 2*k$ by (*auto simp add: dvd-def*)
 hence $2*(k - 2*q^2) = p^2 - q^2$ by *auto*
 with *vw* have $v*w = 2*(k - 2*q^2)$ by (*simp add: zspecial-product*)
 hence $v*w \in zEven$ by (*auto simp only: zEven-def*)
 hence $v \in zEven \vee w \in zEven$ by (*simp add: even-product*)
 with *vwOdd* show *False* by (*auto simp add: odd-iff-not-even*)

```

qed
— first case:  $p$  is not a multiple of 3; hence  $2p$  and  $p^2 + 3q^2$ 
— are coprime; hence both are cubes
{ assume  $p3: \neg 3 \text{ dvd } p$ 
  have  $g3: \neg 3 \text{ dvd } ?g$ 
  proof (rule ccontr, simp)
    assume  $3 \text{ dvd } ?g$  hence  $3 \text{ dvd } 2*p$  by (simp add: zgcd-greatest-iff)
    hence ( $3::\text{int}$ )  $\text{dvd } 2 \vee 3 \text{ dvd } p$ 
    by (auto simp only: zprime-3 zprime-zdvd-zmult-general)
    with  $p3$  show False by arith
qed
have  $pq\text{-relprime}: \text{zgcd } p \ q = 1$ 
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix  $z$  assume  $z: \text{zprime } z$  and  $zp: z \text{ dvd } p$  and  $zq: z \text{ dvd } q$ 
  hence  $z \text{ dvd } p+q \wedge z \text{ dvd } p-q$  by (auto simp only: dvd-add dvd-diff)
  with  $vw$  have  $z \text{ dvd } v \wedge z \text{ dvd } w$  by simp
  with  $z \text{ vwx}$  show False
  by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed
have  $\text{factors-relprime}: ?g = 1$ 
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix  $z$  assume  $z: \text{zprime } z$  and  $z2p: z \text{ dvd } 2*p$  and  $zpq: z \text{ dvd } p^2 + 3*q^2$ 
  hence  $zg: z \text{ dvd } ?g$  by (simp add: zgcd-greatest-iff)
  with  $g\text{Odd}$  have  $z \neq 2$  by auto
  with  $z$  have  $zg2: z > 2$  by (auto simp add: zprime-def)
  from  $z \ z2p$  have  $z \text{ dvd } 2 \vee z \text{ dvd } p$  by (simp only: zprime-zdvd-zmult-general)
  moreover
  { assume  $z \text{ dvd } 2$ 
    hence  $z \leq 2$  by (auto simp add: zdvd-imp-le)
    with  $zg2$  have False by simp }
  ultimately have  $zp: z \text{ dvd } p$  by auto
  hence  $z \text{ dvd } p^2$  by (auto simp add: power2-eq-square)
  with  $zpq$  have  $z \text{ dvd } p^2 + 3*q^2 - p^2$  by (simp only: dvd-diff)
  hence  $z \text{ dvd } 3*q^2$  by auto
  with  $z$  have  $z \text{ dvd } 3 \vee z \text{ dvd } q^2$  by (simp only: zprime-zdvd-zmult-general)
  moreover
  { assume  $z \text{ dvd } 3$ 
    hence  $z \leq 3$  by (auto simp add: zdvd-imp-le)
    with  $zg2$  have  $z = 3$  by auto
    with  $zg \ g3$  have False by auto }
  ultimately have  $z \text{ dvd } q^2$  by auto
  with  $z$  have  $z \text{ dvd } q$  by (rule zprime-zdvd-power)
  with  $zp \ z \ pq\text{-relprime}$  show False
  by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed
moreover from  $vw \ vwpq$  have  $pqx: (2*p)*(p^2 + 3*q^2) = x^3$  by auto
moreover have  $\text{triv3}: 3 = \text{nat } 3 \wedge \text{nat } 3 > 1 \wedge 3 \in \text{zOdd}$ 
  by (unfold zOdd-def, auto)
ultimately have  $\exists c. 2*p = c^3$ 
  by (simp only: int-relprime-odd-power-divisors)
then obtain  $c$  where  $c: c^3 = 2*p$  by auto
from  $pqx \ \text{factors-relprime}$  have  $\text{zgcd } (p^2 + 3*q^2) \ (2*p) = 1$ 

```

and $(p^2 + 3q^2)(2p) = x^3$ **by** (*auto simp add: zgcd-commute mult-ac*)
with *triv3* **have** $\exists d. p^2 + 3q^2 = d^3$
by (*simp only: int-relprime-odd-power-divisors*)
then obtain d **where** $d: p^2 + 3q^2 = d^3$ **by** *auto*
have $d \in zOdd$
proof (*rule ccontr*)
assume $d \notin zOdd$ **hence** $d \in zEven$ **by** (*rule not-odd-impl-even*)
hence $d^3 \in zEven$ **by** (*simp only: power-preserves-even*)
hence $2 \text{ dvd } d^3$ **by** (*simp add: zEven-def dvd-def*)
moreover have $2 \text{ dvd } 2p$ **by** (*rule dvd-triv-left*)
ultimately have $2 \text{ dvd } zgcd(2p)(d^3)$ **by** (*simp add: zgcd-greatest-iff*)
with *d factors-relprime* **show** *False* **by** *simp*
qed
with *d pq-relprime* **have** $zgcd p q = 1 \wedge p^2 + 3q^2 = d^3 \wedge d \in zOdd$
by *simp*
hence *is-cube-form p q* **by** (*rule qf3-cube-impl-cube-form*)
then obtain $a b$ **where** $p = a^3 - 9a^2b^2 \wedge q = 3a^2b - 3b^3$
by (*unfold is-cube-form-def, auto*)
hence $ab: p = a(a+3b)(a-3b) \wedge q = b(a+b)(a-b) \cdot 3$
by (*simp add: nat-number ring-simps*)
with c **have** $abc: (2a)(a+3b)(a-3b) = c^3$ **by** *auto*
have *ab-relprime: zgcd a b = 1*
proof (*simp only: zgcd1-iff-no-common-primedivisor, clarify*)
fix z **assume** $z: zprime z$ **and** $za: z \text{ dvd } a$ **and** $zb: z \text{ dvd } b$
with ab **have** $z \text{ dvd } p \wedge z \text{ dvd } q$ **by** *simp*
with z *pq-relprime* **show**
False **by** (*auto simp add: zgcd1-iff-no-common-primedivisor*)
qed
have $ab1: zgcd(2a)(a+3b) = 1$
proof (*simp only: zgcd1-iff-no-common-primedivisor, clarify*)
fix z **assume** $z: zprime z$ **and** $z \text{ dvd } 2a$ **and** $zab: z \text{ dvd } a+3b$
hence $z \text{ dvd } 2 \vee z \text{ dvd } a$ **by** (*simp add: zprime-zdvd-zmult*)
moreover have $zn2: \neg z \text{ dvd } 2$
proof (*rule ccontr, simp*)
assume $z2: z \text{ dvd } 2$
hence $z \leq 2$ **by** (*simp only: zdvd-imp-le*)
with z **have** $z = 2$ **by** (*unfold zprime-def, auto*)
with zab **have** $ab2: 2 \text{ dvd } a+3b$ **by** *simp*
moreover have $2 \text{ dvd } 2b$ **by** (*rule dvd-triv-left*)
ultimately have $2 \text{ dvd } a+3b-2b$ **by** (*rule dvd-diff*)
hence $2 \text{ dvd } a+b$ **by** *arith*
hence $2 \text{ dvd } (a+b)((a-b)b \cdot 3)$ **by** (*rule dvd-mult2*)
with ab **have** $qEven: 2 \text{ dvd } q$ **by** (*simp only: mult-ac*)
from $ab2$ **have** $2 \text{ dvd } (a+3b)((a-3b)a)$ **by** (*rule dvd-mult2*)
with ab **have** $2 \text{ dvd } p$ **by** (*simp only: mult-ac*)
with $qEven$ **have** $2 \text{ dvd } zgcd p q$ **by** (*simp add: zgcd-greatest-iff*)
with *pq-relprime* **show** *False* **by** *auto*
qed
ultimately have $za: z \text{ dvd } a$ **by** *auto*
with zab **have** $z \text{ dvd } a+3b-a$ **by** (*simp only: dvd-diff*)
hence $z \text{ dvd } 3b$ **by** *simp*
with z **have** $z \text{ dvd } 3 \vee z \text{ dvd } b$ **by** (*simp only: zprime-zdvd-zmult-general*)

```

moreover
{ assume  $z \text{ dvd } 3$ 
  with  $z$  have  $z \leq 3$  by (auto simp add: zdvd-imp-le)
  moreover from  $zn2$  have  $z \neq 2$  by auto
  moreover from  $z$  have  $z > 1$  by (simp add: zprime-def)
  ultimately have  $z=3$  by auto
  with  $za$  have  $3 \text{ dvd } a$  by simp
  with  $ab$  have  $3 \text{ dvd } p$  by auto
  with  $p3$  have False by auto }
ultimately have  $z \text{ dvd } b$  by auto
with  $za$   $z$  ab-relprime show False
  by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed
have  $ab2: zgcd(a+3*b)(a-3*b) = 1$ 
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix  $z$  assume  $z: zprime\ z$  and  $zab1: z \text{ dvd } a+3*b$  and  $zab2: z \text{ dvd } a-3*b$ 
  hence  $z \text{ dvd } (a+3*b)+(a-3*b)$  by (simp only: dvd-add)
  hence  $z \text{ dvd } 2*a$  by simp
  with  $zab1$   $z$  ab1 show False
  by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed
have  $zgcd(a-3*b)(2*a) = 1$ 
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix  $z$  assume  $z: zprime\ z$  and  $z2a: z \text{ dvd } 2*a$  and  $zab: z \text{ dvd } a-3*b$ 
  hence  $z \text{ dvd } 2*a-(a-3*b)$  by (simp only: dvd-diff)
  moreover have  $2*a-(a-3*b) = a+3*b$  by simp
  ultimately have  $z \text{ dvd } a+3*b$  by simp
  with  $z2a$   $z$  ab1 show False
  by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed
with abc ab1 ab2 triv3 have  $\exists k\ l\ m. 2*a=k^3 \wedge a+3*b=l^3 \wedge a-3*b=m^3$ 
  by (simp only: int-triple-relprime-odd-power-divisors)
then obtain  $\alpha\ \beta\ \gamma$  where albeqa:
   $2*a = \gamma^3 \wedge a - 3*b = \alpha^3 \wedge a+3*b = \beta^3$  by auto
— show this is a (smaller) solution
hence  $\alpha^3 + \beta^3 = \gamma^3$  by auto
moreover have  $\alpha*\beta*\gamma \neq 0$ 
proof (rule ccontr, safe)
  assume  $\alpha * \beta * \gamma = 0$ 
  with albeqa ab have  $p=0$  by (auto simp add: power-0-left)
  with vwpq vwx show False by auto
qed
moreover have  $\gamma \in zEven$ 
proof —
  have  $2*a \in zEven$  by (simp add: zEven-def)
  with albeqa have  $\gamma^3 \in zEven$  by simp
  thus thesis by (simp only: power-preserves-even)
qed
moreover have  $zgcd\ \alpha\ \beta=1$ 
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix  $z$  assume  $z: zprime\ z$  and  $za: z \text{ dvd } \alpha$  and  $zb: z \text{ dvd } \beta$ 
  hence  $z \text{ dvd } \alpha * \alpha^2 \wedge z \text{ dvd } \beta * \beta^2$  by simp

```

hence $z \text{ dvd } \alpha^{\wedge 2} \wedge z \text{ dvd } \beta^{\wedge 2}$ by (auto simp only: power-Suc)
 with $ab2$ have $z \text{ dvd } a - 3*b \wedge z \text{ dvd } a + 3*b$ by auto
 with $ab2$ show *False*
 by (auto simp add: zgcd1-iff-no-common-primedivisor)

qed

moreover have $\text{nat}|x^{\wedge 3}| < \text{nat}|y^{\wedge 3}|$

proof -

let $?A = p^{\wedge 2} + 3*q^{\wedge 2}$
 from vwx $vwpq$ have $x^{\wedge 3} = 2*p*?A$ by auto
 also with ab have $\dots = 2*a*((a+3*b)*(a-3*b)*?A)$ by auto
 also with $abega$ have $\dots = \gamma^{\wedge 3} * ((a+3*b)*(a-3*b)*?A)$ by auto
 finally have $eq: |x^{\wedge 3}| = |\gamma^{\wedge 3}| * |(a+3*b)*(a-3*b)*?A|$
 by (auto simp add: abs-mult)
 with $\langle 0 < \text{nat}|x^{\wedge 3}| \rangle$ have $|(a+3*b)*(a-3*b)*?A| > 0$ by auto
 hence $eqpos: |(a+3*b)*(a-3*b)| > 0$ by auto
 moreover have $Ag1: |?A| > 1$

proof -

have $Aqf3: \text{is-}qfN\ ?A\ 3$ by (auto simp add: is- qfN -def)
 moreover have $triv3b: (3::int) \geq 1$ by simp
 ultimately have $?A \geq 0$ by (simp only: qfN -pos)
 hence $?A > 1 \vee ?A = 0 \vee ?A = 1$ by auto
 moreover
 { assume $?A = 0$ with $triv3b$ have $p = 0 \wedge q = 0$ by (rule qfN -zero)
 with $vwpq$ vwx have *False* by auto }

moreover

{ assume $A1: ?A = 1$
 have $q=0$
 proof (rule ccontr)
 assume $q \neq 0$
 hence $q^{\wedge 2} > 0$ by (simp add: zero-less-power2)
 hence $3*q^{\wedge 2} > 1$ by arith
 moreover have $p^{\wedge 2} \geq 0$ by (rule zero-le-power2)
 ultimately have $?A > 1$ by arith
 with $A1$ show *False* by simp

qed

with $A1$ have $p21: p^{\wedge 2} = 1$ by simp
 hence $|p| = 1$ by (rule power2-eq1-iff)
 with $vwpq$ vwx $A1$ have $|x^{\wedge 3}| = 2$ by auto
 hence *False* by (rule two-not-abs-cube) }

ultimately show *?thesis* by auto

qed

ultimately have

$|(a+3*b)*(a-3*b)|*1 < |(a+3*b)*(a-3*b)|*?A$
 by (simp only: zmult-zless-mono2)

with $eqpos$ have $|(a+3*b)*(a-3*b)|*?A > 1$ by arith
 hence $|(a+3*b)*(a-3*b)*?A| > 1$ by (auto simp add: abs-mult)
 moreover have $|\gamma^{\wedge 3}| > 0$

proof -

from eq have $|\gamma^{\wedge 3}| = 0 \implies |x^{\wedge 3}| = 0$ by auto
 with $\langle 0 < \text{nat}|x^{\wedge 3}| \rangle$ show *?thesis* by auto

qed

ultimately have $|\gamma^{\wedge 3}| * 1 < |\gamma^{\wedge 3}| * |(a+3*b)*(a-3*b)*?A|$

by (rule zmult-zless-mono2)
 with eq have $|x^3| > |\gamma^3|$ by auto
 thus ?thesis by arith
 qed
 ultimately have ?thesis by auto }
 moreover
 — second case: $p = 3r$ and hence $x^3 = (18r)(q^2 + 3r^2)$ and these
 — factors are coprime; hence both are cubes
 { assume p3: 3 dvd p
 then obtain r where r: $p = 3*r$ by (auto simp add: dvd-def)
 moreover have 3 dvd $3*(3*r^2 + q^2)$ by (rule dvd-triv-left)
 ultimately have pq3: 3 dvd $p^2 + 3*q^2$ by (simp add: power-mult-distrib)
 moreover from p3 have 3 dvd $2*p$ by (rule dvd-mult)
 ultimately have g3: 3 dvd ?g by (simp add: zgcd-greatest-iff)
 have qr-relprime: $\text{zgcd } q \ r = 1$
 proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
 fix z assume z: zprime z and zq: z dvd q and z dvd r
 with r have z dvd p by simp
 with zq have z dvd $p+q \wedge z \text{ dvd } p-q$ by simp
 with vw have z dvd zgcd v w by (simp add: zgcd-greatest-iff)
 with vwx z show False by (auto simp add: zprime-def)
 qed
 have factors-relprime: $\text{zgcd } (18*r) \ (q^2 + 3*r^2) = 1$
 proof —
 from g3 obtain k where k: ?g = $3*k$ by (auto simp add: dvd-def)
 have k = 1
 proof (rule ccontr)
 assume k $\neq 1$
 with g1 k have k > 1 by auto
 then obtain h where h: zprime h $\wedge h \text{ dvd } k$
 by (frule-tac a=k in zprime-factor-exists, blast)
 with k have hg: $3*h \text{ dvd } ?g$ by (auto simp add: mult-dvd-mono)
 hence $3*h \text{ dvd } p^2 + 3*q^2$ and hp: $3*h \text{ dvd } 2*p$
 by (auto simp only: zgcd-greatest-iff)
 then obtain s where s: $p^2 + 3*q^2 = (3*h)*s$
 by (auto simp add: dvd-def)
 with r have rqh: $3*r^2 + q^2 = h*s$ by (simp add: power-mult-distrib)
 from hp r have $3*h \text{ dvd } 3*(2*r)$ by simp
 moreover have $(3::\text{int}) \neq 0$ by simp
 ultimately have h dvd $2*r$ by (rule zdvd-mult-cancel)
 with h have h dvd $2 \vee h \text{ dvd } r$ by (simp only: zprime-zdvd-zmult-general)
 moreover have $\neg h \text{ dvd } 2$
 proof (rule ccontr, simp)
 assume h dvd 2
 with h have h=2 by (auto simp add: zdvd-not-zless zprime-def)
 with hg have $2*3 \text{ dvd } ?g$ by auto
 hence 2 dvd ?g by (rule dvd-mult-left)
 with gOdd show False by simp
 qed
 ultimately have hr: h dvd r by simp
 then obtain t where r = $h*t$ by (auto simp add: dvd-def)
 hence t: $r^2 = h*(h*t^2)$ by (auto simp add: power2-eq-square)

```

with rgh have h*s = h*(3*h*t^2) + q^2 by simp
hence q^2 = h*(s - 3*h*t^2) by (simp add: zdiff-zmult-distrib2)
hence h dvd q^2 by simp
with h have h dvd q by (auto dest: zprime-zdvd-power)
with hr have h dvd zgcd q r by (simp add: zgcd-greatest-iff)
with h qr-relprime show False by (unfold zprime-def, auto)
qed
with k r have 3 = zgcd (2*(3*r)) ((3*r)^2 + 3*q^2) by auto
also have ... = zgcd (3*(2*r)) (3*(3*r^2 + q^2))
  by (simp add: power-mult-distrib)
also have ... = 3 * zgcd (2*r) (3*r^2 + q^2)
  by (simp only: zgcd-zmult-distrib2)
finally have zgcd (2*r) (3*r^2 + q^2) = 1 by auto
moreover have zgcd (3*3) (3*r^2 + q^2) = 1
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix h::int assume h dvd 3*3 and h: zprime h and hrq: h dvd 3*r^2 + q^2
  hence h dvd 3 ∨ h dvd 3 by (simp only: zprime-zdvd-zmult-general)
  hence h3: h dvd 3 by simp
  have h ≤ 3
  proof (rule ccontr)
    assume ¬ h ≤ 3 hence h > 3 by simp
    with h3 show False by (auto simp add: zdvd-not-zless)
  qed
  with h have h = 2 ∨ h = 3 by (unfold zprime-def, auto)
  with h h3 have h = 3 ∨ (2::int) dvd 3 by auto
  hence h=3 by arith
  with hrq obtain s where 3*r^2+q^2 = 3*s by (auto simp add: dvd-def)
  hence q^2 = 3*(s - r^2) by auto
  hence 3 dvd q^2 and zprime 3 by (auto simp only: dvd-triv-left zprime-3)
  hence 3 dvd q by (rule-tac p=3 in zprime-zdvd-power)
  with p3 have 3 dvd p+q ∧ 3 dvd p-q by simp
  with vw have 3 dvd zgcd v w by (simp add: zgcd-greatest-iff)
  with vwx show False by auto
qed
ultimately have zgcd ((3*3)*(2*r)) (3*r^2 + q^2) = 1
  by (simp only: zgcd-zmult-cancel)
thus ?thesis by (auto simp add: mult-ac add-ac)
qed
moreover have rqx: (18*r)*(q^2 + 3*r^2) = x^3
proof -
  from vwx vwpq have x^3 = 2*p*(p^2 + 3*q^2) by auto
  also with r have ... = 2*(3*r)*(9*r^2 + 3*q^2)
    by (auto simp add: power2-eq-square)
  finally show ?thesis by auto
qed
moreover have triv3: 3 = nat 3 ∧ nat 3 > 1 ∧ 3 ∈ zOdd
  by (unfold zOdd-def, auto)
ultimately have ∃ c. 18*r = c^3
  by (simp only: int-relprime-odd-power-divisors)
then obtain c1 where c1: c1^3 = 3*(6*r) by auto
hence 3 dvd c1^3 and zprime 3 by (auto simp only: dvd-triv-left zprime-3)
hence 3 dvd c1 by (rule-tac p=3 in zprime-zdvd-power)

```

with $c1$ **obtain** c **where** $c: 3*c^3 = 2*r$
by (*auto simp add: power-mult-distrib dvd-def*)
from rqx **factors-relprime** **have** $zgcd (q^2 + 3*r^2) (18*r) = 1$
and $(q^2 + 3*r^2)*(18*r) = x^3$ **by** (*auto simp add: zgcd-commute mult-ac*)
with $triv3$ **have** $\exists d. q^2 + 3*r^2 = d^3$
by (*simp only: int-relprime-odd-power-divisors*)
then obtain d **where** $d: q^2 + 3*r^2 = d^3$ **by** *auto*
have $d \in zOdd$
proof (*rule ccontr*)
assume $d \notin zOdd$ **hence** $d \in zEven$ **by** (*rule not-odd-impl-even*)
hence $d^3 \in zEven$ **by** (*simp only: power-preserves-even*)
hence $2 \text{ dvd } d^3$ **by** (*simp add: zEven-def dvd-def*)
moreover **have** $2 \text{ dvd } 2*(9*r)$ **by** (*rule dvd-triv-left*)
ultimately **have** $2 \text{ dvd } zgcd (2*(9*r)) (d^3)$ **by** (*simp add: zgcd-greatest-iff*)
with d **factors-relprime** **show** *False* **by** *auto*
qed
with d **qr-relprime** **have** $zgcd q r = 1 \wedge q^2 + 3*r^2 = d^3 \wedge d \in zOdd$
by *simp*
hence *is-cube-form* $q r$ **by** (*rule qf3-cube-impl-cube-form*)
then obtain $a b$ **where** $q = a^3 - 9*a*b^2 \wedge r = 3*a^2*b - 3*b^3$
by (*unfold is-cube-form-def, auto*)
hence $ab: q = a*(a+3*b)*(a-3*b) \wedge r = b*(a+b)*(a-b)*3$
by (*simp add: nat-number ring-simps*)
with c **have** $abc: (2*b)*(a+b)*(a-b) = c^3$ **by** *auto*
have $ab\text{-relprime}: zgcd a b = 1$
proof (*simp only: zgcd1-iff-no-common-primedivisor, clarify*)
fix z **assume** $z: zprime z$ **and** $za: z \text{ dvd } a$ **and** $zb: z \text{ dvd } b$
with ab **have** $z \text{ dvd } q \wedge z \text{ dvd } r$ **by** *simp*
with z **qr-relprime** **show** *False*
by (*auto simp add: zgcd1-iff-no-common-primedivisor*)
qed
have $ab1: zgcd (2*b) (a+b) = 1$
proof (*simp only: zgcd1-iff-no-common-primedivisor, clarify*)
fix z **assume** $z: zprime z$ **and** $z \text{ dvd } 2*b$ **and** $zab: z \text{ dvd } a+b$
hence $z \text{ dvd } 2 \vee z \text{ dvd } b$ **by** (*simp add: zprime-zdvd-zmult*)
moreover
{ **assume** $z2: z \text{ dvd } 2$
hence $z \leq 2$ **by** (*simp only: zdvd-imp-le*)
with z **have** $z = 2$ **by** (*unfold zprime-def, auto*)
with zab **have** $ab2: 2 \text{ dvd } a+b$ **by** *simp*
moreover **have** $2 \text{ dvd } 2*b$ **by** (*rule dvd-triv-left*)
ultimately **have** $2 \text{ dvd } a+b+2*b$ **by** (*rule dvd-add*)
hence $2 \text{ dvd } a+3*b$ **by** *arith*
hence $2 \text{ dvd } (a+3*b)*((a-3*b)*a)$ **by** (*rule dvd-mult2*)
with ab **have** $qEven: 2 \text{ dvd } q$ **by** (*simp only: mult-ac*)
from $ab2$ **have** $2 \text{ dvd } (a+b)*((a-b)*3*b)$ **by** (*rule dvd-mult2*)
with ab **have** $2 \text{ dvd } r$ **by** (*simp only: mult-ac*)
with $qEven$ **have** $2 \text{ dvd } zgcd q r$ **by** (*simp add: zgcd-greatest-iff*)
with $qr\text{-relprime}$ **have** *False* **by** *auto* }
moreover
{ **assume** $zb: z \text{ dvd } b$
with zab **have** $z \text{ dvd } a+b-b$ **by** (*simp only: dvd-diff*)

```

    hence z dvd a by simp
    with zb ab-relprime z have False
      by (auto simp add: zgcd1-iff-no-common-primedivisor) }
  ultimately show False by auto
qed
have ab2: zgcd (a+b) (a-b) = 1
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix z assume z: zprime z and zab1: z dvd a+b and zab2: z dvd a-b
  hence z dvd (a+b)-(a-b) by (simp only: dvd-diff)
  hence z dvd 2*b by simp
  with zab1 z ab1 show False
    by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed
have zgcd (a-b) (2*b) = 1
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix z assume z: zprime z and z2b: z dvd 2*b and zab: z dvd a-b
  hence z dvd a-b+2*b by (simp only: dvd-add)
  moreover have a-b+2*b = a+b by simp
  ultimately have z dvd a+b by simp
  with z2b z ab1 show False
    by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed
with abc ab1 ab2 triv3 have  $\exists k l m. 2*b = k^3 \wedge a+b = l^3 \wedge a-b = m^3$ 
  by (simp only: int-triple-relprime-odd-power-divisors)
then obtain  $\alpha 1 \beta \gamma$  where  $a1: 2*b = \gamma^3 \wedge a-b = \alpha 1^3 \wedge a+b = \beta^3$ 
  by auto
then obtain  $\alpha$  where  $\alpha = -\alpha 1$  by auto
— show this is a (smaller) solution
with triv3 a1 have  $a2: \alpha^3 = b-a$  by (auto simp only: neg-odd-power)
with a1 have  $\alpha^3 + \beta^3 = \gamma^3$  by auto
moreover have  $\alpha*\beta*\gamma \neq 0$ 
proof (rule ccontr, safe)
  assume  $\alpha * \beta * \gamma = 0$ 
  with a1 a2 ab have  $r=0$  by (auto simp add: power-0-left)
  with r vwpq vwx show False by auto
qed
moreover have  $\gamma \in zEven$ 
proof —
  have  $2*b \in zEven$  by (simp add: zEven-def)
  with a1 have  $\gamma^3 \in zEven$  by simp
  thus ?thesis by (simp only: power-preserves-even)
qed
moreover have  $zgcd \alpha \beta = 1$ 
proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
  fix z assume z: zprime z and za: z dvd  $\alpha$  and zb: z dvd  $\beta$ 
  hence z dvd  $\alpha * \alpha^2 \wedge z dvd \beta * \beta^2$  by simp
  hence z dvd  $\alpha^3 \wedge z dvd \beta^3$  by (auto simp only: power-Suc)
  with a1 a2 have z dvd  $b-a \wedge z dvd a+b$  by auto
  hence z dvd  $-(b-a) \wedge z dvd a+b$  by (auto simp only: dvd-minus-iff)
  with ab2 z show False
    by (auto simp add: zgcd1-iff-no-common-primedivisor)
qed

```

```

moreover have  $\text{nat}|\gamma^3| < \text{nat}|x^3|$ 
proof -
  let  $?A = p^2 + 3*q^2$ 
  from  $vwx\ vwpq$  have  $x^3 = 2*p*?A$  by auto
  also with  $r$  have  $\dots = 6*r*?A$  by auto
  also with  $ab$  have  $\dots = 2*b*(9*(a+b)*(a-b)*?A)$  by auto
  also with  $a1$  have  $\dots = \gamma^3 * (9*(a+b)*(a-b)*?A)$  by auto
  finally have  $\text{eq}: |x^3| = |\gamma^3| * |9*(a+b)*(a-b)*?A|$ 
    by (auto simp add: abs-mult)
  with  $\langle 0 < \text{nat}|x^3| \rangle$  have  $|9*(a+b)*(a-b)*?A| > 0$  by auto
  hence  $|(a+b)*(a-b)*?A| \geq 1$  by arith
  hence  $|9*(a+b)*(a-b)*?A| > 1$  by arith
  moreover have  $|\gamma^3| > 0$ 
  proof -
    from  $\text{eq}$  have  $|\gamma^3| = 0 \implies |x^3|=0$  by auto
    with  $\langle 0 < \text{nat}|x^3| \rangle$  show  $?thesis$  by auto
  qed
  ultimately have  $|\gamma^3| * 1 < |\gamma^3| * |9*(a+b)*(a-b)*?A|$ 
    by (rule zmult-zless-mono2)
  with  $\text{eq}$  have  $|x^3| > |\gamma^3|$  by auto
  thus  $?thesis$  by arith
qed
  ultimately have  $?thesis$  by auto }
  ultimately show  $?thesis$  by auto
qed
thus  $?case$  by auto
qed

```

The theorem. Puts equation in requested shape.

```

theorem fermat3:
  assumes  $\text{ass}: (x::\text{int})^3 + y^3 = z^3$ 
  shows  $x*y*z=0$ 
proof (rule ccontr)
  let  $?g = \text{zgcd } x\ y$ 
  let  $?c = z\ \text{div } ?g$ 
  assume  $xyz0: x*y*z \neq 0$ 
  — divide out the g.c.d.
  hence  $x \neq 0 \vee y \neq 0$  by simp
  then obtain  $a\ b$  where  $ab: x = ?g*a \wedge y = ?g*b \wedge \text{zgcd } a\ b=1$ 
    by (frule-tac a=x in make-zrelprime, auto)
  moreover have  $abc: ?c*?g = z \wedge a^3 + b^3 = ?c^3 \wedge a*b*?c \neq 0$ 
  proof -
    from  $xyz0$  have  $g0: ?g \neq 0$  by (simp add: zgcd-def gcd-zero)
    have  $zgab: z^3 = ?g^3 * (a^3 + b^3)$ 
    proof -
      from  $ab$  and  $\text{ass}$  have  $z^3 = (?g*a)^3 + (?g*b)^3$  by simp
      thus  $?thesis$  by (simp only: power-mult-distrib zadd-zmult-distrib2)
    qed
  have  $cgz: ?c * ?g = z$ 
  proof -
    from  $zgab$  have  $?g^3\ \text{dvd}\ z^3$  by simp
    hence  $?g\ \text{dvd}\ z$  by (simp only: zpower-zdvd-mono)

```

```

    thus ?thesis by (simp only: mult-ac zdvd-mult-div-cancel)
  qed
  moreover have  $a^3 + b^3 = ?c^3$ 
  proof -
    have  $?c^3 * ?g^3 = (a^3 + b^3) * ?g^3$ 
    proof -
      have  $?c^3 * ?g^3 = (?c * ?g)^3$  by (simp only: power-mult-distrib)
      also with cgz have  $\dots = z^3$  by simp
      also with zgab have  $\dots = ?g^3 * (a^3 + b^3)$  by simp
      finally show ?thesis by simp
    qed
    with g0 show ?thesis by auto
  qed
  moreover from ab and xyz0 and cgz have  $a * b * ?c \neq 0$  by auto
  ultimately show ?thesis by simp
qed
— make both sides even
have  $\exists u v w. u^3 + v^3 = w^3 \wedge u * v * w \neq 0 \wedge w \in zEven \wedge zgcd u v = 1$ 
proof -
  let ?Q  $u v w = u^3 + v^3 = w^3 \wedge u * v * w \neq 0 \wedge w \in zEven \wedge zgcd u v = 1$ 
  have  $a \in zEven \vee b \in zEven \vee ?c \in zEven$ 
  proof (rule ccontr)
    assume  $\neg(a \in zEven \vee b \in zEven \vee ?c \in zEven)$ 
    hence aodd:  $a \in zOdd$  and bodd:  $b \in zOdd$  and codd:  $?c \in zOdd$ 
    by (auto simp add: odd-iff-not-even)
    hence  $?c^3 - b^3 \in zEven$  by (simp only: power-preserves-odd odd-minus-odd)
    moreover from abc have  $?c^3 - b^3 = a^3$  by simp
    ultimately have  $a^3 \in zEven$  by auto
    hence  $a \in zEven$  by (simp only: power-preserves-even)
    with aodd show False by (simp add: odd-iff-not-even)
  qed
  moreover
  { assume  $a \in zEven$ 
    then obtain  $u v w$  where uvwabc:  $u = -b \wedge v = ?c \wedge w = a \wedge w \in zEven$ 
      by auto
    moreover with abc have  $u * v * w \neq 0$  by auto
    moreover have uvw:  $u^3 + v^3 = w^3$ 
    proof -
      from uvwabc have  $u^3 + v^3 = (-1 * b)^3 + ?c^3$  by simp
      also have  $\dots = (-1)^3 * b^3 + ?c^3$  by (simp only: power-mult-distrib)
      also have  $\dots = -(b^3) + ?c^3$  by (auto simp add: neg-one-odd-power)
      also with abc and uvwabc have  $\dots = w^3$  by auto
      finally show ?thesis by simp
    qed
  }
  moreover have  $zgcd u v = 1$ 
  proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
    fix  $h :: int$  assume hu:  $h \text{ dvd } u$  and hv:  $h \text{ dvd } v$  and h:  $zprime h$ 
    with uvwabc have  $h \text{ dvd } ?c * ?c^2$  by (simp only: dvd-mult2)
    with abc have  $h \text{ dvd } a^3 + b^3$  by (simp only: cube-square)
    moreover from hu uvwabc have  $h \text{ dvd } b * b^2$  by simp
    ultimately have  $h \text{ dvd } a^3 + b^3 - b^3$  by (simp only: cube-square dvd-diff)
    with h hu uvwabc have  $h \text{ dvd } a \wedge h \text{ dvd } b$  by (auto dest: zprime-zdvd-power)
  }

```

```

    with h ab show False by (auto simp add: zgcd1-iff-no-common-primedivisor)
  qed
  ultimately have ?Q u v w using ⟨a ∈ zEven⟩ by simp
  hence ?thesis by auto }
moreover
{ assume b ∈ zEven
  then obtain u v w where uvwabc: u = -a ∧ v = ?c ∧ w = b ∧ w ∈ zEven
    by auto
  moreover with abc have u*v*w≠0 by auto
  moreover have uvw: u^3+v^3=w^3
  proof -
    from uvwabc have u^3 + v^3 = (-1*a)^3 + ?c^3 by simp
    also have ... = (-1)^3*a^3 + ?c^3 by (simp only: power-mult-distrib)
    also have ... = - (a^3) + ?c^3 by (auto simp add: neg-one-odd-power)
    also with abc and uvwabc have ... = w^3 by auto
    finally show ?thesis by simp
  qed
  moreover have zgcd u v=1
  proof (simp only: zgcd1-iff-no-common-primedivisor, clarify)
    fix h::int assume hu: h dvd u and h dvd v and h: zprime h
    with uvwabc have h dvd ?c*?c^2 by (simp only: dvd-mult2)
    with abc have h dvd a^3+b^3 by (simp only: cube-square)
    moreover from hu uvwabc have h dvd a*a^2 by simp
    ultimately have h dvd a^3+b^3-a^3 by (simp only: cube-square dvd-diff)
    with h hu uvwabc have h dvd a ∧ h dvd b by (auto dest: zprime-zdvd-power)
    with h ab show False by (auto simp add: zgcd1-iff-no-common-primedivisor)
  qed
  ultimately have ?Q u v w using ⟨b ∈ zEven⟩ by simp
  hence ?thesis by auto }
moreover
{ assume ?c ∈ zEven
  then obtain u v w where uvwabc: u = a ∧ v = b ∧ w = ?c ∧ w ∈ zEven
    by auto
  with abc ab have ?thesis by auto }
ultimately show ?thesis by auto
qed
hence ∃ w. ∃ u v. u^3 + v^3 = w^3 ∧ u*v*w ≠ 0 ∧ w ∈ zEven ∧ zgcd u v=1
  by auto
— show contradiction using the earlier result
thus False by (auto simp only: no-rewritten-fermat3)
qed

```

corollary *fermat-mult3*:

assumes $xyz: (x::int)^n + y^n = z^n$ and $n: 3 \text{ dvd } n$
 shows $x*y*z=0$

proof —

from n obtain m where $n = m*3$ by (auto simp only: mult-ac dvd-def)
 with xyz have $(x^m)^3 + (y^m)^3 = (z^m)^3$ by (simp only: power-mult)
 hence $(x^m)*(y^m)*(z^m) = 0$ by (rule *fermat3*)
 thus ?thesis by auto

qed

end

References

- [DM05] David Delahaye and Micaela Mayero. Diophantus' 20th problem and fermat's last theorem for $n=4$: Formalization of fermat's proofs in the coq proof assistant. <http://hal.archives-ouvertes.fr/hal-00009425/en/>, 2005.
- [Edw77] Harold M. Edwards. *Fermat's Last Theorem. A Genetic Introduction to Algebraic Number Theory*. Springer Verlag, 1977.
- [Oos07] Roelof Oosterhuis. Mechanised theorem proving: Exponents 3 and 4 of Fermat's Last Theorem in Isabelle. Master's thesis, University of Groningen, 2007. <http://www.roelofoosterhuis.nl/MScthesis.pdf>.
- [Wie] Freek Wiedijk. Formalizing 100 theorems. <http://www.cs.ru.nl/~freek/100/>.