Three squares theorem

Anton Danilkin, Loïc Chevalier

March 17, 2025

Abstract

We formalize the Legendre's three squares theorem and its consequences, in particular the following results:

1. A natural number can be represented as the sum of three squares of natural numbers if and only if it is not of the form $4^a(8k+7)$, where a and k are natural numbers.
2. If n is a natural number such that $n \equiv 3 \pmod 8$, then n can be be represented as the sum of three squares of odd natural numbers.

Consequences include the following:

1. An integer n can be written as $n = x^2 + y^2 + z^2 + z$, where x, y, z are integers, if and only if $n \geq 0$.
2. The Legendre's four squares theorem: any natural number can be represented as the sum of four squares of natural numbers.

We follow the book of Melvyn B. Nathanson 'Additive Number Theory: The Classical Bases' [1].

We plan to make use of the first consequence mentioned above in an upcoming AFP entry on Diophantine equations. More concretely, we intend to formalize universal pairs over the integers which requires expressing a natural number as a polynomial in integers while only using few variables.

Contents

1 Properties of residues, congruences, quadratic residues and the Legendre symbol

theory *Residues-Properties*
imports *HOL−Number-Theory.Quadratic-Reciprocity*
begin

1.1 Properties of residues and congruences

lemma *mod-diff-eq-nat*:
fixes $a\ b\ m :: nat$
assumes $a \geq b$
shows $(a - b)\ mod\ m = (m + (a\ mod\ m) - (b\ mod\ m))\ mod\ m$
⟨*proof*⟩

lemma *prime-invertible-int*:
fixes $a\ p :: int$
assumes *prime p*
assumes $\neg\ p\ dvd\ a$
shows $\exists\, b.\ [a * b = 1]\ (mod\ p)$
⟨*proof*⟩

lemma *power-cong*:
fixes $x\ y\ a\ m :: nat$
assumes *coprime a m*
assumes $[x = y]\ (mod\ totient\ m)$
shows $[a\ \hat{}\ x = a\ \hat{}\ y]\ (mod\ m)$
⟨*proof*⟩

lemma *power-cong-alt*:
fixes $x\ a\ m :: nat$
assumes *coprime a m*
shows $a\ \hat{}\ x\ mod\ m = a\ \hat{}\ (x\ mod\ totient\ m)\ mod\ m$
⟨*proof*⟩

1.2 Properties of quadratic residues

lemma *QuadRes-cong*:
fixes $a\ b\ p :: int$
assumes $[a = b]\ (mod\ p)$
assumes *QuadRes p a*
shows *QuadRes p b*
⟨*proof*⟩

lemma *QuadRes-mult*:
fixes *a b p* :: *int*
assumes *QuadRes p a*
assumes *QuadRes p b*
shows *QuadRes p* (*a* ∗ *b*)
⟨*proof*⟩

lemma *QuadRes-inv*:
fixes *a b p* :: *int*
assumes *prime p*
assumes [*a* ∗ *b* = *1*] (*mod p*)
assumes *QuadRes p a*
shows *QuadRes p b*
⟨*proof*⟩

1.3 Properties of the Legendre symbol

lemma *Legendre-cong*:
fixes *a b p* :: *int*
assumes [*a* = *b*] (*mod p*)
shows *Legendre a p* = *Legendre b p*
⟨*proof*⟩

lemma *Legendre-one*:
fixes *p* :: *int*
assumes *p* > *2*
shows *Legendre 1 p* = *1*
⟨*proof*⟩

lemma *Legendre-minus-one*:
fixes *p* :: *int*
assumes *prime p*
assumes *p* > *2*
shows *Legendre* (− *1*) *p* = *1* ⟷ [*p* = *1*] (*mod 4*)
⟨*proof*⟩

lemma *Legendre-minus-one-alt*:
fixes *p* :: *int*
assumes *prime p*
assumes *p* > *2*
shows *Legendre* (− *1*) *p* = (*if* [*p* = *1*] (*mod 4*) *then 1 else* − *1*)
⟨*proof*⟩

lemma *Legendre-two*:
fixes *p* :: *int*
assumes *prime p*
assumes *p* > *2*
shows *Legendre 2 p* = *1* ⟷ [*p* = *1*] (*mod 8*) ∨ [*p* = *7*] (*mod 8*)
⟨*proof*⟩

lemma *Legendre-two-alt*:
fixes $p :: int$
assumes *prime p*
assumes $p > 2$
shows *Legendre 2 p* = (*if* [$p = 1$] (*mod 8*) ∨ [$p = 7$] (*mod 8*) *then 1 else* − *1*)
⟨*proof*⟩

lemma *Legendre-mult*:
fixes *a b p* :: *int*
assumes *prime p*
shows *Legendre* (*a* ∗ *b*) *p* = *Legendre a p* ∗ *Legendre b p*
⟨*proof*⟩

lemma *Legendre-power*:
fixes *a* :: *int*
fixes *n* :: *nat*
fixes *p* :: *int*
assumes *prime p*
assumes $p > 2$
shows *Legendre* (*a* ^ *n*) *p* = (*Legendre a p*) ^ *n*
⟨*proof*⟩

lemma *Legendre-prod*:
fixes *A* :: *'a set*
fixes *f* :: *'a* ⇒ *int*
fixes *p* :: *int*
assumes *prime p*
assumes $p > 2$
shows *Legendre* (*prod f A*) *p* = (∏ *x*∈*A*. *Legendre* (*f x*) *p*)
⟨*proof*⟩

lemma *Legendre-equal*:
fixes *p q* :: *int*
assumes *prime p prime q*
assumes $p > 2$ $q > 2$
assumes $p \neq q$
assumes [$p = 1$] (*mod 4*) ∨ [$q = 1$] (*mod 4*)
shows *Legendre p q* = *Legendre q p*
⟨*proof*⟩

lemma *Legendre-opposite*:
fixes *p q* :: *int*
assumes *prime p prime q*
assumes $p > 2$ $q > 2$
assumes $p \neq q$
assumes [$p = 3$] (*mod 4*) ∧ [$q = 3$] (*mod 4*)
shows *Legendre p q* = − *Legendre q p*
⟨*proof*⟩

end

2 Vectors and matrices, determinants and their properties in dimensions 2 and 3

theory *Low-Dimensional-Linear-Algebra*
imports *Main*
begin

datatype *vec2* =
vec2
($vec2_1$: *int*)
($vec2_2$: *int*)

datatype *vec3* =
vec3
($vec3_1$: *int*)
($vec3_2$: *int*)
($vec3_3$: *int*)

datatype *mat2* =
mat2
($mat2_{11}$: *int*) ($mat2_{12}$: *int*)
($mat2_{21}$: *int*) ($mat2_{22}$: *int*)

datatype *mat3* =
mat3
($mat3_{11}$: *int*) ($mat3_{12}$: *int*) ($mat3_{13}$: *int*)
($mat3_{21}$: *int*) ($mat3_{22}$: *int*) ($mat3_{23}$: *int*)
($mat3_{31}$: *int*) ($mat3_{32}$: *int*) ($mat3_{33}$: *int*)

instantiation *vec2* :: *ab-group-add*
begin

definition *zero-vec2* **where**
zero-vec2 =
vec2
0
0

definition *uminus-vec2* **where**
uminus-vec2 v =
vec2
(− $vec2_1$ *v*)
(− $vec2_2$ *v*)

definition *plus-vec2* **where**

plus-vec2 v1 v2 =
  vec2
  (*vec2*$_1$ *v1* + *vec2*$_1$ *v2*)
  (*vec2*$_2$ *v1* + *vec2*$_2$ *v2*)

definition *minus-vec2* **where**
minus-vec2 v1 v2 =
  vec2
  (*vec2*$_1$ *v1* − *vec2*$_1$ *v2*)
  (*vec2*$_2$ *v1* − *vec2*$_2$ *v2*)

instance
  ⟨*proof*⟩

end

instantiation *vec3* :: *ab-group-add*
begin

definition *zero-vec3* **where**
zero-vec3 =
  vec3
  0
  0
  0

definition *uminus-vec3* **where**
uminus-vec3 v =
  vec3
  (− *vec3*$_1$ *v*)
  (− *vec3*$_2$ *v*)
  (− *vec3*$_3$ *v*)

definition *plus-vec3* **where**
plus-vec3 v1 v2 =
  vec3
  (*vec3*$_1$ *v1* + *vec3*$_1$ *v2*)
  (*vec3*$_2$ *v1* + *vec3*$_2$ *v2*)
  (*vec3*$_3$ *v1* + *vec3*$_3$ *v2*)

definition *minus-vec3* **where**
minus-vec3 v1 v2 =
  vec3
  (*vec3*$_1$ *v1* − *vec3*$_1$ *v2*)
  (*vec3*$_2$ *v1* − *vec3*$_2$ *v2*)
  (*vec3*$_3$ *v1* − *vec3*$_3$ *v2*)

instance
  ⟨*proof*⟩

end

instantiation *mat2* :: *ring-1*
begin

definition *zero-mat2* **where**
zero-mat2 =
mat2
0 0
0 0

definition *one-mat2* **where**
one-mat2 =
mat2
1 0
0 1

definition *uminus-mat2* **where**
uminus-mat2 m =
mat2
(− $mat2_{11}$ *m*) (− $mat2_{12}$ *m*)
(− $mat2_{21}$ *m*) (− $mat2_{22}$ *m*)

definition *plus-mat2* **where**
plus-mat2 m1 m2 =
mat2
($mat2_{11}$ *m1* + $mat2_{11}$ *m2*) ($mat2_{12}$ *m1* + $mat2_{12}$ *m2*)
($mat2_{21}$ *m1* + $mat2_{21}$ *m2*) ($mat2_{22}$ *m1* + $mat2_{22}$ *m2*)

definition *minus-mat2* **where**
minus-mat2 m1 m2 =
mat2
($mat2_{11}$ *m1* − $mat2_{11}$ *m2*) ($mat2_{12}$ *m1* − $mat2_{12}$ *m2*)
($mat2_{21}$ *m1* − $mat2_{21}$ *m2*) ($mat2_{22}$ *m1* − $mat2_{22}$ *m2*)

definition *times-mat2* **where**
times-mat2 m1 m2 =
mat2
($mat2_{11}$ *m1* ∗ $mat2_{11}$ *m2* + $mat2_{12}$ *m1* ∗ $mat2_{21}$ *m2*) ($mat2_{11}$ *m1* ∗ $mat2_{12}$ *m2* + $mat2_{12}$ *m1* ∗ $mat2_{22}$ *m2*)
($mat2_{21}$ *m1* ∗ $mat2_{11}$ *m2* + $mat2_{22}$ *m1* ∗ $mat2_{21}$ *m2*) ($mat2_{21}$ *m1* ∗ $mat2_{12}$ *m2* + $mat2_{22}$ *m1* ∗ $mat2_{22}$ *m2*)

instance
⟨*proof*⟩

end

instantiation *mat3* :: *ring-1*
begin

definition *zero-mat3* **where**
zero-mat3 =
mat3
0 0 0
0 0 0
0 0 0

definition *one-mat3* **where**
one-mat3 =
mat3
1 0 0
0 1 0
0 0 1

definition *uminus-mat3* **where**
uminus-mat3 m =
mat3
(− $mat3_{11}$ *m*) (− $mat3_{12}$ *m*) (− $mat3_{13}$ *m*)
(− $mat3_{21}$ *m*) (− $mat3_{22}$ *m*) (− $mat3_{23}$ *m*)
(− $mat3_{31}$ *m*) (− $mat3_{32}$ *m*) (− $mat3_{33}$ *m*)

definition *plus-mat3* **where**
plus-mat3 m1 m2 =
mat3
($mat3_{11}$ *m1* + $mat3_{11}$ *m2*) ($mat3_{12}$ *m1* + $mat3_{12}$ *m2*) ($mat3_{13}$ *m1* + $mat3_{13}$ *m2*)
($mat3_{21}$ *m1* + $mat3_{21}$ *m2*) ($mat3_{22}$ *m1* + $mat3_{22}$ *m2*) ($mat3_{23}$ *m1* + $mat3_{23}$ *m2*)
($mat3_{31}$ *m1* + $mat3_{31}$ *m2*) ($mat3_{32}$ *m1* + $mat3_{32}$ *m2*) ($mat3_{33}$ *m1* + $mat3_{33}$ *m2*)

definition *minus-mat3* **where**
minus-mat3 m1 m2 =
mat3
($mat3_{11}$ *m1* − $mat3_{11}$ *m2*) ($mat3_{12}$ *m1* − $mat3_{12}$ *m2*) ($mat3_{13}$ *m1* − $mat3_{13}$ *m2*)
($mat3_{21}$ *m1* − $mat3_{21}$ *m2*) ($mat3_{22}$ *m1* − $mat3_{22}$ *m2*) ($mat3_{23}$ *m1* − $mat3_{23}$ *m2*)
($mat3_{31}$ *m1* − $mat3_{31}$ *m2*) ($mat3_{32}$ *m1* − $mat3_{32}$ *m2*) ($mat3_{33}$ *m1* − $mat3_{33}$ *m2*)

definition *times-mat3* **where**
times-mat3 m1 m2 =
mat3
($mat3_{11}$ *m1* ∗ $mat3_{11}$ *m2* + $mat3_{12}$ *m1* ∗ $mat3_{21}$ *m2* + $mat3_{13}$ *m1* ∗ $mat3_{31}$ *m2*) ($mat3_{11}$ *m1* ∗ $mat3_{12}$ *m2* + $mat3_{12}$ *m1* ∗ $mat3_{22}$ *m2* + $mat3_{13}$ *m1* ∗

$mat3_{32}$ $m2$) ($mat3_{11}$ $m1$ * $mat3_{13}$ $m2$ + $mat3_{12}$ $m1$ * $mat3_{23}$ $m2$ + $mat3_{13}$ $m1$ * $mat3_{33}$ $m2$)
($mat3_{21}$ $m1$ * $mat3_{11}$ $m2$ + $mat3_{22}$ $m1$ * $mat3_{21}$ $m2$ + $mat3_{23}$ $m1$ * $mat3_{31}$ $m2$) ($mat3_{21}$ $m1$ * $mat3_{12}$ $m2$ + $mat3_{22}$ $m1$ * $mat3_{22}$ $m2$ + $mat3_{23}$ $m1$ * $mat3_{32}$ $m2$) ($mat3_{21}$ $m1$ * $mat3_{13}$ $m2$ + $mat3_{22}$ $m1$ * $mat3_{23}$ $m2$ + $mat3_{23}$ $m1$ * $mat3_{33}$ $m2$)
($mat3_{31}$ $m1$ * $mat3_{11}$ $m2$ + $mat3_{32}$ $m1$ * $mat3_{21}$ $m2$ + $mat3_{33}$ $m1$ * $mat3_{31}$ $m2$) ($mat3_{31}$ $m1$ * $mat3_{12}$ $m2$ + $mat3_{32}$ $m1$ * $mat3_{22}$ $m2$ + $mat3_{33}$ $m1$ * $mat3_{32}$ $m2$) ($mat3_{31}$ $m1$ * $mat3_{13}$ $m2$ + $mat3_{32}$ $m1$ * $mat3_{23}$ $m2$ + $mat3_{33}$ $m1$ * $mat3_{33}$ $m2$)

instance
⟨*proof*⟩

end

consts *vec-dot* :: *'a* ⇒ *'a* ⇒ *int* (‹<- | ->› *65*)

definition *vec2-dot* :: *vec2* ⇒ *vec2* ⇒ *int* **where**
vec2-dot v1 v2 = $vec2_1$ *v1* * $vec2_1$ *v2* + $vec2_2$ *v1* * $vec2_2$ *v2*

adhoc-overloading *vec-dot* ⇌ *vec2-dot*

definition *vec3-dot* :: *vec3* ⇒ *vec3* ⇒ *int* **where**
vec3-dot v1 v2 = $vec3_1$ *v1* * $vec3_1$ *v2* + $vec3_2$ *v1* * $vec3_2$ *v2* + $vec3_3$ *v1* * $vec3_3$ *v2*

adhoc-overloading *vec-dot* ⇌ *vec3-dot*

lemma *vec2-dot-zero-left* [*simp*]:
fixes *v* :: *vec2*
shows <*0* | *v*> = *0*
⟨*proof*⟩

lemma *vec2-dot-zero-right* [*simp*]:
fixes *v* :: *vec2*
shows <*v* | *0*> = *0*
⟨*proof*⟩

lemma *vec3-dot-zero-left* [*simp*]:
fixes *v* :: *vec3*
shows <*0* | *v*> = *0*
⟨*proof*⟩

lemma *vec3-dot-zero-right* [*simp*]:
fixes *v* :: *vec3*
shows <*v* | *0*> = *0*
⟨*proof*⟩

consts *mat-app* :: $'a \Rightarrow 'b \Rightarrow 'b$ (**infixr** ‹$› *65*)

definition *mat2-app* :: *mat2* ⇒ *vec2* ⇒ *vec2* **where**
mat2-app m v =
vec2
(*mat2*$_{11}$ *m* ∗ *vec2*$_1$ *v* + *mat2*$_{12}$ *m* ∗ *vec2*$_2$ *v*)
(*mat2*$_{21}$ *m* ∗ *vec2*$_1$ *v* + *mat2*$_{22}$ *m* ∗ *vec2*$_2$ *v*)

adhoc-overloading *mat-app* ⇌ *mat2-app*

definition *mat3-app* :: *mat3* ⇒ *vec3* ⇒ *vec3* **where**
mat3-app m v =
vec3
(*mat3*$_{11}$ *m* ∗ *vec3*$_1$ *v* + *mat3*$_{12}$ *m* ∗ *vec3*$_2$ *v* + *mat3*$_{13}$ *m* ∗ *vec3*$_3$ *v*)
(*mat3*$_{21}$ *m* ∗ *vec3*$_1$ *v* + *mat3*$_{22}$ *m* ∗ *vec3*$_2$ *v* + *mat3*$_{23}$ *m* ∗ *vec3*$_3$ *v*)
(*mat3*$_{31}$ *m* ∗ *vec3*$_1$ *v* + *mat3*$_{32}$ *m* ∗ *vec3*$_2$ *v* + *mat3*$_{33}$ *m* ∗ *vec3*$_3$ *v*)

adhoc-overloading *mat-app* ⇌ *mat3-app*

lemma *mat2-app-zero* [*simp*]:
fixes *m* :: *mat2*
shows *m* $ *0* = *0*
⟨*proof*⟩

lemma *mat3-app-zero* [*simp*]:
fixes *m* :: *mat3*
shows *m* $ *0* = *0*
⟨*proof*⟩

lemma *mat2-app-one* [*simp*]:
fixes *v* :: *vec2*
shows *1* $ *v* = *v*
⟨*proof*⟩

lemma *mat3-app-one* [*simp*]:
fixes *v* :: *vec3*
shows *1* $ *v* = *v*
⟨*proof*⟩

lemma *mat2-app-mul* [*simp*]:
fixes *m1 m2* :: *mat2*
fixes *v* :: *vec2*
shows *m1* ∗ *m2* $ *v* = *m1* $ *m2* $ *v*
⟨*proof*⟩

lemma *mat3-app-mul* [*simp*]:
fixes *m1 m2* :: *mat3*
fixes *v* :: *vec3*
shows *m1* ∗ *m2* $ *v* = *m1* $ *m2* $ *v*

⟨*proof*⟩

consts *mat-det* :: *'a* ⇒ *int*

definition *mat2-det* **where**
mat2-det m = *mat2*$_{11}$ *m* ∗ *mat2*$_{22}$ *m* − *mat2*$_{12}$ *m* ∗ *mat2*$_{21}$ *m*

adhoc-overloading *mat-det* ⇌ *mat2-det*

definition *mat3-det* **where**
mat3-det m =
  mat3$_{11}$ *m* ∗ *mat3*$_{22}$ *m* ∗ *mat3*$_{33}$ *m*
+ *mat3*$_{12}$ *m* ∗ *mat3*$_{23}$ *m* ∗ *mat3*$_{31}$ *m*
+ *mat3*$_{13}$ *m* ∗ *mat3*$_{21}$ *m* ∗ *mat3*$_{32}$ *m*
− *mat3*$_{11}$ *m* ∗ *mat3*$_{23}$ *m* ∗ *mat3*$_{32}$ *m*
− *mat3*$_{12}$ *m* ∗ *mat3*$_{21}$ *m* ∗ *mat3*$_{33}$ *m*
− *mat3*$_{13}$ *m* ∗ *mat3*$_{22}$ *m* ∗ *mat3*$_{31}$ *m*

adhoc-overloading *mat-det* ⇌ *mat3-det*

lemma *mat2-mul-det* [*simp*]:
fixes *m1 m2* :: *mat2*
shows *mat-det* (*m1* ∗ *m2*) = *mat-det m1* ∗ *mat-det m2*
⟨*proof*⟩

lemma *mat3-mul-det* [*simp*]:
fixes *m1 m2* :: *mat3*
shows *mat-det* (*m1* ∗ *m2*) = *mat-det m1* ∗ *mat-det m2*
⟨*proof*⟩

consts *mat-sym* :: *'a* ⇒ *bool*

definition *mat2-sym* :: *mat2* ⇒ *bool* **where**
mat2-sym m = (*mat2*$_{12}$ *m* = *mat2*$_{21}$ *m*)

adhoc-overloading *mat-sym* ⇌ *mat2-sym*

definition *mat3-sym* :: *mat3* ⇒ *bool* **where**
mat3-sym m = (*mat3*$_{12}$ *m* = *mat3*$_{21}$ *m* ∧ *mat3*$_{13}$ *m* = *mat3*$_{31}$ *m* ∧ *mat3*$_{23}$ *m* = *mat3*$_{32}$ *m*)

adhoc-overloading *mat-sym* ⇌ *mat3-sym*

consts *mat-transpose* :: *'a* ⇒ *'a* (‹-^{T}› [*91*] *90*)

definition *mat2-transpose* :: *mat2* ⇒ *mat2* **where**
mat2-transpose m =
  mat2
  (*mat2*$_{11}$ *m*) (*mat2*$_{21}$ *m*)

$(mat2_{12}\ m)\ (mat2_{22}\ m)$

adhoc-overloading *mat-transpose* $\rightleftharpoons$ *mat2-transpose*

definition *mat3-transpose* :: *mat3* $\Rightarrow$ *mat3* **where**
mat3-transpose m =
mat3
$(mat3_{11}\ m)\ (mat3_{21}\ m)\ (mat3_{31}\ m)$
$(mat3_{12}\ m)\ (mat3_{22}\ m)\ (mat3_{32}\ m)$
$(mat3_{13}\ m)\ (mat3_{23}\ m)\ (mat3_{33}\ m)$

adhoc-overloading *mat-transpose* $\rightleftharpoons$ *mat3-transpose*

lemma *mat2-transpose-involution* [*simp*]:
fixes *m* :: *mat2*
shows $(m^T)^T = m$
⟨*proof*⟩

lemma *mat3-transpose-involution* [*simp*]:
fixes *m* :: *mat3*
shows $(m^T)^T = m$
⟨*proof*⟩

lemma *mat2-sym-criterion*:
fixes *m* :: *mat2*
shows *mat-sym m* $\longleftrightarrow$ $m^T = m$
⟨*proof*⟩

lemma *mat3-sym-criterion*:
fixes *m* :: *mat3*
shows *mat-sym m* $\longleftrightarrow$ $m^T = m$
⟨*proof*⟩

lemma *mat2-transpose-one* [*simp*]: $(1 :: mat2)^T = 1$
⟨*proof*⟩

lemma *mat3-transpose-one* [*simp*]: $(1 :: mat3)^T = 1$
⟨*proof*⟩

lemma *mat2-transpose-mul* [*simp*]:
fixes *a b* :: *mat2*
shows $(a * b)^T = b^T * a^T$
⟨*proof*⟩

lemma *mat3-transpose-mul* [*simp*]:
fixes *a b* :: *mat3*
shows $(a * b)^T = b^T * a^T$
⟨*proof*⟩

lemma *vec2-dot-transpose-left*:
fixes m :: *mat2*
fixes u v :: *vec2*
shows $<m^T$ \$ $u \mid v> = <u \mid m$ \$ $v>$
⟨*proof*⟩

lemma *vec2-dot-transpose-right*:
fixes m :: *mat2*
fixes u v :: *vec2*
shows $<u \mid m^T$ \$ $v> = <m$ \$ $u \mid v>$
⟨*proof*⟩

lemma *vec3-dot-transpose-left*:
fixes m :: *mat3*
fixes u v :: *vec3*
shows $<m^T$ \$ $u \mid v> = <u \mid m$ \$ $v>$
⟨*proof*⟩

lemma *vec3-dot-transpose-right*:
fixes m :: *mat3*
fixes u v :: *vec3*
shows $<u \mid m^T$ \$ $v> = <m$ \$ $u \mid v>$
⟨*proof*⟩

lemma *mat2-det-tranpose* [*simp*]:
fixes m :: *mat2*
shows *mat-det* (m^T) = *mat-det* m
⟨*proof*⟩

lemma *mat3-det-tranpose* [*simp*]:
fixes m :: *mat3*
shows *mat-det* (m^T) = *mat-det* m
⟨*proof*⟩

consts *mat-inverse* :: $'a \Rightarrow 'a$ ($\langle$-$^{-1}\rangle$ [*91*] *90*)

definition *mat2-inverse* :: *mat2* ⇒ *mat2* **where**
mat2-inverse m =
mat2
($mat2_{22}$ m) (− $mat2_{12}$ m)
(− $mat2_{21}$ m) ($mat2_{11}$ m)

adhoc-overloading *mat-inverse* ⇌ *mat2-inverse*

definition *mat3-inverse* :: *mat3* ⇒ *mat3* **where**
mat3-inverse m =
mat3
($mat3_{22}$ m * $mat3_{33}$ m − $mat3_{23}$ m * $mat3_{32}$ m) ($mat3_{13}$ m * $mat3_{32}$ m −

$mat3_{12}\ m * mat3_{33}\ m)\ (mat3_{12}\ m * mat3_{23}\ m - mat3_{13}\ m * mat3_{22}\ m)$
$(mat3_{23}\ m * mat3_{31}\ m - mat3_{21}\ m * mat3_{33}\ m)\ (mat3_{11}\ m * mat3_{33}\ m -$
$mat3_{13}\ m * mat3_{31}\ m)\ (mat3_{13}\ m * mat3_{21}\ m - mat3_{11}\ m * mat3_{23}\ m)$
$(mat3_{21}\ m * mat3_{32}\ m - mat3_{22}\ m * mat3_{31}\ m)\ (mat3_{12}\ m * mat3_{31}\ m -$
$mat3_{11}\ m * mat3_{32}\ m)\ (mat3_{11}\ m * mat3_{22}\ m - mat3_{12}\ m * mat3_{21}\ m)$

adhoc-overloading *mat-inverse* $\rightleftharpoons$ *mat3-inverse*

lemma *mat2-inverse-cancel*:
fixes m :: *mat2*
assumes *mat-det* $m = 1$
shows $m * m^{-1} = 1\ m^{-1} * m = 1$
⟨*proof*⟩

lemma *mat3-inverse-cancel*:
fixes m :: *mat3*
assumes *mat-det* $m = 1$
shows $m * m^{-1} = 1\ m^{-1} * m = 1$
⟨*proof*⟩

lemma *mat2-inverse-cancel-left*:
fixes $m\ a$ :: *mat2*
assumes *mat-det* $m = 1$
shows $m * (m^{-1} * a) = a\ m^{-1} * (m * a) = a$
⟨*proof*⟩

lemma *mat3-inverse-cancel-left*:
fixes $m\ a$ :: *mat3*
assumes *mat-det* $m = 1$
shows $m * (m^{-1} * a) = a\ m^{-1} * (m * a) = a$
⟨*proof*⟩

lemma *mat2-inverse-cancel-right*:
fixes $m\ a$ :: *mat2*
assumes *mat-det* $m = 1$
shows $a * (m * m^{-1}) = a\ a * (m^{-1} * m) = a$
⟨*proof*⟩

lemma *mat3-inverse-cancel-right*:
fixes $m\ a$ :: *mat3*
assumes *mat-det* $m = 1$
shows $a * (m * m^{-1}) = a\ a * (m^{-1} * m) = a$
⟨*proof*⟩

lemma *mat2-inversable-cancel-left*:
fixes $m\ a1\ a2$ :: *mat2*
assumes *mat-det* $m = 1$
assumes $m * a1 = m * a2$

shows *a1 = a2*
⟨*proof*⟩

lemma *mat3-inversable-cancel-left*:
fixes *m a1 a2* :: *mat3*
assumes *mat-det m = 1*
assumes *m * a1 = m * a2*
shows *a1 = a2*
⟨*proof*⟩

lemma *mat2-inversable-cancel-right*:
fixes *m a1 a2* :: *mat2*
assumes *mat-det m = 1*
assumes *a1 * m = a2 * m*
shows *a1 = a2*
⟨*proof*⟩

lemma *mat3-inversable-cancel-right*:
fixes *m a1 a2* :: *mat3*
assumes *mat-det m = 1*
assumes *a1 * m = a2 * m*
shows *a1 = a2*
⟨*proof*⟩

lemma *mat2-inverse-det* [*simp*]:
fixes *m* :: *mat2*
shows *mat-det* (m^{-1}) = *mat-det m*
⟨*proof*⟩

lemma *mat3-inverse-det* [*simp*]:
fixes *m* :: *mat3*
shows *mat-det* (m^{-1}) = $(\textit{mat-det}\ m)^2$
⟨*proof*⟩

lemma *mat2-inverse-transpose*:
fixes *m* :: *mat2*
shows $(m^T)^{-1} = (m^{-1})^T$
⟨*proof*⟩

lemma *mat3-inverse-transpose*:
fixes *m* :: *mat3*
shows $(m^T)^{-1} = (m^{-1})^T$
⟨*proof*⟩

lemma *mat2-special-preserves-zero*:
fixes *u* :: *mat2*
fixes *v* :: *vec2*
assumes *mat-det u = 1*
shows *u $ v = 0* ⟷ *v = 0*

⟨*proof*⟩

lemma *mat3-special-preserves-zero*:
fixes *u* :: *mat3*
fixes *v* :: *vec3*
assumes *mat-det u* = *1*
shows *u* $ *v* = *0* ⟷ *v* = *0*
⟨*proof*⟩

end

3 Properties of quadratic forms and their equivalences

theory *Quadratic-Forms*
imports *Complex-Main Low-Dimensional-Linear-Algebra*
begin

consts *qf-app* :: *'a* ⇒ *'b* ⇒ *int* (**infixl** ‹$$› *65*)

definition *qf2-app* :: *mat2* ⇒ *vec2* ⇒ *int* **where**
qf2-app m v = *<v* | *m* $ *v>*

adhoc-overloading *qf-app* ⇌ *qf2-app*

definition *qf3-app* :: *mat3* ⇒ *vec3* ⇒ *int* **where**
qf3-app m v = *<v* | *m* $ *v>*

adhoc-overloading *qf-app* ⇌ *qf3-app*

lemma *qf2-app-zero* [*simp*]:
fixes *m* :: *mat2*
shows *m* $$ *0* = *0*
⟨*proof*⟩

lemma *qf3-app-zero* [*simp*]:
fixes *m* :: *mat3*
shows *m* $$ *0* = *0*
⟨*proof*⟩

consts *qf-positive-definite* :: *'a* ⇒ *bool*

definition *qf2-positive-definite* :: *mat2* ⇒ *bool* **where**
qf2-positive-definite m = (∀ *v*. *v* ≠ *0* ⟶ *m* $$ *v* > *0*)

adhoc-overloading *qf-positive-definite* ⇌ *qf2-positive-definite*

definition *qf3-positive-definite* :: *mat3* ⇒ *bool* **where**

qf3-positive-definite $m = (\forall\, v.\ v \neq 0 \longrightarrow m\ \$\$\ v > 0)$

adhoc-overloading *qf-positive-definite* $\rightleftharpoons$ *qf3-positive-definite*

lemma *qf2-positive-definite-positive*:
fixes m :: *mat2*
assumes *qf-positive-definite* m
shows $\forall\, v.\ m\ \$\$\ v \geq 0$
⟨*proof*⟩

lemma *qf3-positive-definite-positive*:
fixes m :: *mat3*
assumes *qf-positive-definite* m
shows $\forall\, v.\ m\ \$\$\ v \geq 0$
⟨*proof*⟩

consts *qf-action* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** ‹·› *55*)

definition *qf2-action* :: *mat2* ⇒ *mat2* ⇒ *mat2* **where**
qf2-action $a\ u = u^T * a * u$

adhoc-overloading *qf-action* $\rightleftharpoons$ *qf2-action*

definition *qf3-action* :: *mat3* ⇒ *mat3* ⇒ *mat3* **where**
qf3-action $a\ u = u^T * a * u$

adhoc-overloading *qf-action* $\rightleftharpoons$ *qf3-action*

lemma *qf2-action-id*:
fixes a :: *mat2*
shows $a \cdot 1 = a$
⟨*proof*⟩

lemma *qf3-action-id*:
fixes a :: *mat3*
shows $a \cdot 1 = a$
⟨*proof*⟩

lemma *qf2-action-mul* [*simp*]:
fixes $a\ u\ v$ :: *mat2*
shows $a \cdot (u * v) = (a \cdot u) \cdot v$
⟨*proof*⟩

lemma *qf3-action-mul* [*simp*]:
fixes $a\ u\ v$ :: *mat3*
shows $a \cdot (u * v) = (a \cdot u) \cdot v$
⟨*proof*⟩

consts *qf-equiv* :: $'a \Rightarrow 'a \Rightarrow$ *bool* (**infix** ‹~› *65*)

definition *qf2-equiv* :: *mat2* ⇒ *mat2* ⇒ *bool* **where**
qf2-equiv a b = (∃ *u*. *mat-det u* = *1* ∧ *a* · *u* = *b*)

adhoc-overloading *qf-equiv* ⇌ *qf2-equiv*

definition *qf3-equiv* :: *mat3* ⇒ *mat3* ⇒ *bool* **where**
qf3-equiv a b = (∃ *u*. *mat-det u* = *1* ∧ *a* · *u* = *b*)

adhoc-overloading *qf-equiv* ⇌ *qf3-equiv*

lemma *qf2-equiv-sym-impl*:
fixes *a b* :: *mat2*
shows *a* ~ *b* ⟹ *b* ~ *a*
⟨*proof*⟩

lemma *qf3-equiv-sym-impl*:
fixes *a b* :: *mat3*
shows *a* ~ *b* ⟹ *b* ~ *a*
⟨*proof*⟩

lemma *qf2-equiv-sym*:
fixes *a b* :: *mat2*
shows *a* ~ *b* ⟷ *b* ~ *a*
⟨*proof*⟩

lemma *qf3-equiv-sym*:
fixes *a b* :: *mat3*
shows *a* ~ *b* ⟷ *b* ~ *a*
⟨*proof*⟩

lemma *qf2-equiv-trans*:
fixes *a b c* :: *mat2*
assumes *a* ~ *b*
assumes *b* ~ *c*
shows *a* ~ *c*
⟨*proof*⟩

lemma *qf3-equiv-trans*:
fixes *a b c* :: *mat3*
assumes *a* ~ *b*
assumes *b* ~ *c*
shows *a* ~ *c*
⟨*proof*⟩

lemma *qf2-action-app* [*simp*]:
fixes *a u* :: *mat2*
fixes *v* :: *vec2*
shows (*a* · *u*) $$ *v* = *a* $$ (*u* $ *v*)

⟨*proof*⟩

lemma *qf3-action-app* [*simp*]:
fixes *a u* :: *mat3*
fixes *v* :: *vec3*
shows (*a* · *u*) $$ *v* = *a* $$ (*u* $ *v*)
⟨*proof*⟩

lemma *qf2-equiv-preserves-positive-definite*:
fixes *a b* :: *mat2*
assumes *a* ~ *b*
shows *qf-positive-definite a* ⟷ *qf-positive-definite b*
⟨*proof*⟩

lemma *qf3-equiv-preserves-positive-definite*:
fixes *a b* :: *mat3*
assumes *a* ~ *b*
shows *qf-positive-definite a* ⟷ *qf-positive-definite b*
⟨*proof*⟩

lemma *qf2-equiv-preserves-sym*:
fixes *a b* :: *mat2*
assumes *a* ~ *b*
shows *mat2-sym a* ⟷ *mat2-sym b*
⟨*proof*⟩

lemma *qf3-equiv-preserves-sym*:
fixes *a b* :: *mat3*
assumes *a* ~ *b*
shows *mat3-sym a* ⟷ *mat3-sym b*
⟨*proof*⟩

lemma *qf2-equiv-preserves-det*:
fixes *a b* :: *mat2*
assumes *a* ~ *b*
shows *mat-det a* = *mat-det b*
⟨*proof*⟩

lemma *qf3-equiv-preserves-det*:
fixes *a b* :: *mat3*
assumes *a* ~ *b*
shows *mat-det a* = *mat-det b*
⟨*proof*⟩

lemma *qf2-equiv-preserves-range-subset*:
fixes *a b* :: *mat2*
assumes *a* ~ *b*
shows *range* (($$) *b*) ⊆ *range* (($$) *a*)
⟨*proof*⟩

lemma *qf3-equiv-preserves-range-subset*:
fixes *a b* :: *mat3*
assumes *a* ~ *b*
shows *range* ((\$\$) *b*) ⊆ *range* ((\$\$) *a*)
⟨*proof*⟩

lemma *qf2-equiv-preserves-range*:
fixes *a b* :: *mat2*
assumes *a* ~ *b*
shows *range* ((\$\$) *a*) = *range* ((\$\$) *b*)
⟨*proof*⟩

lemma *qf3-equiv-preserves-range*:
fixes *a b* :: *mat3*
assumes *a* ~ *b*
shows *range* ((\$\$) *a*) = *range* ((\$\$) *b*)
⟨*proof*⟩

Lemma 1.1 from [1].

lemma *qf2-positive-definite-criterion*:
fixes *a*
assumes *mat-sym a*
shows *qf-positive-definite a* ⟷ *mat2*$_{11}$ *a* > *0* ∧ *mat-det a* > *0*
⟨*proof*⟩

lemma *congruence-class-close*:
fixes *k m* :: *int*
assumes *m* > *0*
shows ∃ *t*. *2* ∗ |*k* + *m* ∗ *t*| ≤ *m* (**is** ∃ *t*. *?P t*)
⟨*proof*⟩

Lemma 1.2 from [1].

lemma *lemma-1-2*:
fixes *b* :: *mat2*
assumes *mat-sym b*
assumes *qf-positive-definite b*
shows ∃ *a*. *a* ~ *b* ∧
2 ∗ |*mat2*$_{12}$ *a*| ≤ *mat2*$_{11}$ *a* ∧
mat2$_{11}$ *a* ≤ (*2* / *sqrt 3*) ∗ *sqrt* (*mat-det a*) (**is** ∃ *a*. *?P a*)
⟨*proof*⟩

Theorem 1.2 from [1].

theorem *qf2-det-one-equiv-canonical*:
fixes *f* :: *mat2*
assumes *mat-sym f*
assumes *qf-positive-definite f*
assumes *mat-det f* = *1*
shows *f* ~ *1*

⟨*proof*⟩

Lemma 1.3 from [1].

lemma *lemma-1-3*:
fixes a :: *mat3*
assumes *mat-sym a*
defines $a' \equiv$
mat2
$(mat3_{11}\ a * mat3_{22}\ a - (mat3_{12}\ a)^2)\ (mat3_{11}\ a * mat3_{23}\ a - mat3_{12}\ a * mat3_{13}\ a)$
$(mat3_{11}\ a * mat3_{23}\ a - mat3_{12}\ a * mat3_{13}\ a)\ (mat3_{11}\ a * mat3_{33}\ a - (mat3_{13}\ a)^2)$

defines $d' \equiv$
mat-det (
mat2
$(mat3_{11}\ a)\ (mat3_{12}\ a)$
$(mat3_{12}\ a)\ (mat3_{22}\ a)$
)

shows
mat-det $a' = mat3_{11}\ a *$ *mat-det a* (**is** *?P*)
$\bigwedge x.\ mat3_{11}\ a * (a\ \$\$\ x) =$
$(mat3_{11}\ a * vec3_1\ x + mat3_{12}\ a * vec3_2\ x + mat3_{13}\ a * vec3_3\ x)^2 +$
$(a'\ \$\$\ (vec2\ (vec3_2\ x)\ (vec3_3\ x)))$ (**is** $\bigwedge x.\ ?Q\ x$)
qf-positive-definite a $\Longrightarrow$ *qf-positive-definite* a'
qf-positive-definite a $\longleftrightarrow$ $mat3_{11}\ a > 0 \wedge d' > 0 \wedge$ *mat-det a* > 0
⟨*proof*⟩

Lemma 1.4 from [1].

lemma *lemma-1-4*:
fixes b :: *mat3*
fixes v' :: *mat2*
fixes $r\ s$ :: *int*
assumes *mat-sym b*
assumes *qf-positive-definite b*
assumes *mat-det* $v' = 1$
defines $b' \equiv$
mat2
$(mat3_{11}\ b * mat3_{22}\ b - (mat3_{12}\ b)^2)\ (mat3_{11}\ b * mat3_{23}\ b - mat3_{12}\ b * mat3_{13}\ b)$
$(mat3_{11}\ b * mat3_{23}\ b - mat3_{12}\ b * mat3_{13}\ b)\ (mat3_{11}\ b * mat3_{33}\ b - (mat3_{13}\ b)^2)$

defines $a' \equiv b' \cdot v'$
defines $v \equiv$
mat3
1 r s
0 $(mat2_{11}\ v')\ (mat2_{12}\ v')$

0 ($mat2_{21}$ *v'*) ($mat2_{22}$ *v'*)

defines $a \equiv b \cdot v$
shows
$\bigwedge y.\ mat3_{11}\ b * (b\ \$\$\ y) =$
$(mat3_{11}\ b * vec3_1\ y + mat3_{12}\ b * vec3_2\ y + mat3_{13}\ b * vec3_3\ y)^2 +$
$(b'\ \$\$\ (vec2\ (vec3_2\ y)\ (vec3_3\ y)))$ (**is** $\bigwedge y.\ ?P\ y$)
$mat3_{11}\ a = mat3_{11}\ b$
$\bigwedge x.\ mat3_{11}\ a * (a\ \$\$\ x) =$
$(mat3_{11}\ a * vec3_1\ x + mat3_{12}\ a * vec3_2\ x + mat3_{13}\ a * vec3_3\ x)^2 +$
$(a'\ \$\$\ (vec2\ (vec3_2\ x)\ (vec3_3\ x)))$ (**is** $\bigwedge x.\ ?Q\ x$)
⟨*proof*⟩

Lemma 1.5 from [1].

lemma *lemma-1-5*:
fixes $u_{11}\ u_{21}\ u_{31}$
assumes *Gcd* $\{u_{11}, u_{21}, u_{31}\} = 1$
shows $\exists u.\ mat3_{11}\ u = u_{11} \wedge mat3_{21}\ u = u_{21} \wedge mat3_{31}\ u = u_{31} \wedge$ *mat-det u* $= 1$
⟨*proof*⟩

Lemma 1.6 from [1].

lemma *lemma-1-6*:
fixes *c* :: *mat3*
assumes *mat-sym c*
assumes *qf-positive-definite c*
shows $\exists a.\ a \sim c \wedge$
$2 * (max\ |mat3_{12}\ a|\ |mat3_{13}\ a|) \leq mat3_{11}\ a \wedge$
$mat3_{11}\ a \leq (4\ /\ 3) * root\ 3\ (mat\text{-}det\ a)$
⟨*proof*⟩

Theorem 1.3 from [1].

theorem *qf3-det-one-equiv-canonical*:
fixes *f* :: *mat3*
assumes *mat-sym f*
assumes *qf-positive-definite f*
assumes *mat-det f* $= 1$
shows $f \sim 1$
⟨*proof*⟩

end

4 Legendre's three squares theorem and its consequences

theory *Three-Squares*
imports *Dirichlet-L.Dirichlet-Theorem Residues-Properties Quadratic-Forms*
begin

4.1 Legendre's three squares theorem

definition *quadratic-residue-alt* :: *int* ⇒ *int* ⇒ *bool* **where**
quadratic-residue-alt m a = (∃ *x y*. $x^2 - a = y * m$)

lemma *quadratic-residue-alt-equiv*: *quadratic-residue-alt* = *QuadRes*
⟨*proof*⟩

lemma *sq-nat-abs*: $(nat\ |v|)^2 = nat\ (v^2)$
⟨*proof*⟩

Lemma 1.7 from [1].

lemma *three-squares-using-quadratic-residue*:
fixes *n d′* :: *nat*
assumes *n* ≥ *2*
assumes *d′* > *0*
assumes *QuadRes* (*d′* ∗ *n* − *1*) (− *d′*)
shows ∃ $x_1\ x_2\ x_3$. $n = {x_1}^2 + {x_2}^2 + {x_3}^2$
⟨*proof*⟩

lemma *prime-linear-combination*:
fixes *a m* :: *nat*
assumes *m* > *1*
assumes *coprime a m*
obtains *j* :: *nat* **where** *prime* (*a* + *m* ∗ *j*) ∧ *j* ≠ *0*
⟨*proof*⟩

Lemma 1.8 from [1].

lemma *three-squares-using-mod-four*:
fixes *n* :: *nat*
assumes *n mod 4* = *2*
shows ∃ $x_1\ x_2\ x_3$. $n = {x_1}^2 + {x_2}^2 + {x_3}^2$
⟨*proof*⟩

lemma *three-mod-eight-power-iff*:
fixes *n* :: *nat*
shows (*3* :: *int*) ^ *n mod 8* = (*if even n then 1 else 3*)
⟨*proof*⟩

Lemma 1.9 from [1].

lemma *three-squares-using-mod-eight*:
fixes *n* :: *nat*
assumes *n mod 8* ∈ {*1*, *3*, *5*}
shows ∃ $x_1\ x_2\ x_3$. $n = {x_1}^2 + {x_2}^2 + {x_3}^2$
⟨*proof*⟩

lemma *power-two-mod-eight*:
fixes *n* :: *nat*
shows n^2 *mod 8* ∈ {*0*, *1*, *4*}

⟨*proof*⟩

lemma *power-two-mod-four*:
  fixes n :: *nat*
  shows n^2 *mod* 4 ∈ {0, 1}
  ⟨*proof*⟩

Theorem 1.4 from [1].

theorem *three-squares-iff*:
  fixes n :: *nat*
  shows $(\exists\, x_1\ x_2\ x_3.\ n = {x_1}^2 + {x_2}^2 + {x_3}^2) \longleftrightarrow (\nexists\, a\ k.\ n = 4\ \hat{}\ a * (8 * k + 7))$
⟨*proof*⟩

Theorem 1.5 from [1].

theorem *odd-three-squares-using-mod-eight*:
  fixes n :: *nat*
  assumes n *mod* 8 = 3
  shows $\exists\, x_1\ x_2\ x_3.$ *odd* $x_1 \wedge$ *odd* $x_2 \wedge$ *odd* $x_3 \wedge n = {x_1}^2 + {x_2}^2 + {x_3}^2$
⟨*proof*⟩

4.2 Consequences

lemma *four-decomposition*:
  fixes n :: *nat*
  shows $\exists\, x\ y\ z.\ n = x^2 + y^2 + z^2 + z$
⟨*proof*⟩

theorem *four-decomposition-int*:
  fixes n :: *int*
  shows $(\exists\, x\ y\ z.\ n = x^2 + y^2 + z^2 + z) \longleftrightarrow n \geq 0$
⟨*proof*⟩

theorem *four-squares*:
  fixes n :: *nat*
  shows $\exists\, x_1\ x_2\ x_3\ x_4.\ n = {x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2$
⟨*proof*⟩

end

References

[1] M. B. Nathanson. *Additive Number Theory: The Classical Bases*, volume 164 of *Graduate Texts in Mathematics*. Springer, New York, 1996.