

Three squares theorem

Anton Danilkin, Loïc Chevalier

September 13, 2023

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 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	2
1.1	Properties of residues and congruences	2
1.2	Properties of quadratic residues	2
1.3	Properties of the Legendre symbol	3
2	Vectors and matrices, determinants and their properties in dimensions 2 and 3	5
3	Properties of quadratic forms and their equivalences	16

4	Legendre’s three squares theorem and its consequences	22
4.1	Legendre’s three squares theorem	23
4.2	Consequences	24

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 ≥ 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 ¬ p dvd a
  shows ∃ 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 ^ x = a ^ y] (mod m)
⟨proof⟩
```

```
lemma power-cong-alt:
  fixes x a m :: nat
  assumes coprime a m
  shows a ^ x mod m = a ^ (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* $p\ (a * b)$
 $\langle proof \rangle$

lemma *QuadRes-inv*:
fixes $a\ b\ p :: int$
assumes *prime* p
assumes $[a * b = 1] (mod\ p)$
assumes *QuadRes* $p\ a$
shows *QuadRes* $p\ b$
 $\langle proof \rangle$

1.3 Properties of the Legendre symbol

lemma *Legendre-cong*:
fixes $a\ b\ p :: int$
assumes $[a = b] (mod\ p)$
shows *Legendre* $a\ p = Legendre\ b\ p$
 $\langle proof \rangle$

lemma *Legendre-one*:
fixes $p :: int$
assumes $p > 2$
shows *Legendre* $1\ p = 1$
 $\langle proof \rangle$

lemma *Legendre-minus-one*:
fixes $p :: int$
assumes *prime* p
assumes $p > 2$
shows *Legendre* $(- 1)\ p = 1 \iff [p = 1] (mod\ 4)$
 $\langle proof \rangle$

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

lemma *Legendre-two*:
fixes $p :: int$
assumes *prime* p
assumes $p > 2$
shows *Legendre* $2\ p = 1 \iff [p = 1] (mod\ 8) \vee [p = 7] (mod\ 8)$
 $\langle proof \rangle$

lemma *Legendre-two-alt:*

fixes $p :: int$

assumes $prime\ p$

assumes $p > 2$

shows $Legendre\ 2\ p = (if\ [p = 1]\ (mod\ 8) \vee [p = 7]\ (mod\ 8)\ then\ 1\ else\ -\ 1)$

$\langle proof \rangle$

lemma *Legendre-mult:*

fixes $a\ b\ p :: int$

assumes $prime\ p$

shows $Legendre\ (a * b)\ p = Legendre\ a\ p * Legendre\ b\ p$

$\langle proof \rangle$

lemma *Legendre-power:*

fixes $a :: int$

fixes $n :: nat$

fixes $p :: int$

assumes $prime\ p$

assumes $p > 2$

shows $Legendre\ (a \wedge^n)\ p = (Legendre\ a\ p) \wedge^n$

$\langle proof \rangle$

lemma *Legendre-prod:*

fixes $A :: 'a\ set$

fixes $f :: 'a \Rightarrow int$

fixes $p :: int$

assumes $prime\ p$

assumes $p > 2$

shows $Legendre\ (prod\ f\ A)\ p = (\prod_{x \in A}. Legendre\ (f\ x)\ p)$

$\langle proof \rangle$

lemma *Legendre-equal:*

fixes $p\ q :: int$

assumes $prime\ p\ prime\ q$

assumes $p > 2\ q > 2$

assumes $p \neq q$

assumes $[p = 1]\ (mod\ 4) \vee [q = 1]\ (mod\ 4)$

shows $Legendre\ p\ q = Legendre\ q\ p$

$\langle proof \rangle$

lemma *Legendre-opposite:*

fixes $p\ q :: int$

assumes $prime\ p\ prime\ q$

assumes $p > 2\ q > 2$

assumes $p \neq q$

assumes $[p = 3]\ (mod\ 4) \wedge [q = 3]\ (mod\ 4)$

shows $Legendre\ p\ q = - Legendre\ q\ p$

$\langle proof \rangle$

end

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

```
theory Low-Dimensional-Linear-Algebra
  imports HOL-Library.Adhoc-Overloading
begin
```

```
datatype vec2 =
  vec2
  (vec21 : int)
  (vec22 : int)
```

```
datatype vec3 =
  vec3
  (vec31 : int)
  (vec32 : int)
  (vec33 : int)
```

```
datatype mat2 =
  mat2
  (mat211 : int) (mat212 : int)
  (mat221 : int) (mat222 : int)
```

```
datatype mat3 =
  mat3
  (mat311 : int) (mat312 : int) (mat313 : int)
  (mat321 : int) (mat322 : int) (mat323 : int)
  (mat331 : int) (mat332 : int) (mat333 : int)
```

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

```
definition zero-vec2 where
  zero-vec2 =
    vec2
    0
    0
```

```
definition uminus-vec2 where
  uminus-vec2 v =
    vec2
    (− vec21 v)
    (− vec22 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

$\langle proof \rangle$

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

$\langle proof \rangle$

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*)


```

mat332 m2) (mat311 m1 * mat313 m2 + mat312 m1 * mat323 m2 + mat313 m1
* mat333 m2)
  (mat321 m1 * mat311 m2 + mat322 m1 * mat321 m2 + mat323 m1 * mat331
m2) (mat321 m1 * mat312 m2 + mat322 m1 * mat322 m2 + mat323 m1 *
mat332 m2) (mat321 m1 * mat313 m2 + mat322 m1 * mat323 m2 + mat323 m1
* mat333 m2)
  (mat331 m1 * mat311 m2 + mat332 m1 * mat321 m2 + mat333 m1 * mat331
m2) (mat331 m1 * mat312 m2 + mat332 m1 * mat322 m2 + mat333 m1 *
mat332 m2) (mat331 m1 * mat313 m2 + mat332 m1 * mat323 m2 + mat333 m1
* mat333 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 ⇒ 'b ⇒ 'b (**infix** \$ 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 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 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 $mat\text{-}sym\ m \longleftrightarrow m^T = m$
<proof>

lemma *mat3-sym-criterion*:

fixes *m* :: *mat3*
shows $mat\text{-}sym\ m \longleftrightarrow m^T = m$
<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 $\langle m^T \$ u \mid v \rangle = \langle u \mid m \$ v \rangle$
 $\langle proof \rangle$

lemma *vec2-dot-transpose-right*:
fixes $m :: mat2$
fixes $u v :: vec2$
shows $\langle u \mid m^T \$ v \rangle = \langle m \$ u \mid v \rangle$
 $\langle proof \rangle$

lemma *vec3-dot-transpose-left*:
fixes $m :: mat3$
fixes $u v :: vec3$
shows $\langle m^T \$ u \mid v \rangle = \langle u \mid m \$ v \rangle$
 $\langle proof \rangle$

lemma *vec3-dot-transpose-right*:
fixes $m :: mat3$
fixes $u v :: vec3$
shows $\langle u \mid m^T \$ v \rangle = \langle m \$ u \mid v \rangle$
 $\langle proof \rangle$

lemma *mat2-det-tranpose* [*simp*]:
fixes $m :: mat2$
shows $mat-det (m^T) = mat-det m$
 $\langle proof \rangle$

lemma *mat3-det-tranpose* [*simp*]:
fixes $m :: mat3$
shows $mat-det (m^T) = mat-det m$
 $\langle proof \rangle$

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

definition *mat2-inverse* :: $mat2 \Rightarrow mat2$ **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* :: $mat3 \Rightarrow mat3$ **where**
mat3-inverse $m =$
 $mat3$
 $(mat3_{22} m * mat3_{33} m - mat3_{23} m * mat3_{32} m) (mat3_{13} m * mat3_{32} m -$

$$\begin{aligned}
& \text{mat3}_{12} m * \text{mat3}_{33} m) (\text{mat3}_{12} m * \text{mat3}_{23} m - \text{mat3}_{13} m * \text{mat3}_{22} m) \\
& (\text{mat3}_{23} m * \text{mat3}_{31} m - \text{mat3}_{21} m * \text{mat3}_{33} m) (\text{mat3}_{11} m * \text{mat3}_{33} m - \\
& \text{mat3}_{13} m * \text{mat3}_{31} m) (\text{mat3}_{13} m * \text{mat3}_{21} m - \text{mat3}_{11} m * \text{mat3}_{23} m) \\
& (\text{mat3}_{21} m * \text{mat3}_{32} m - \text{mat3}_{22} m * \text{mat3}_{31} m) (\text{mat3}_{12} m * \text{mat3}_{31} m - \\
& \text{mat3}_{11} m * \text{mat3}_{32} m) (\text{mat3}_{11} m * \text{mat3}_{22} m - \text{mat3}_{12} m * \text{mat3}_{21} m)
\end{aligned}$$

adhoc-overloading *mat-inverse mat3-inverse*

lemma *mat2-inverse-cancel:*

fixes $m :: \text{mat2}$
assumes $\text{mat-det } m = 1$
shows $m * m^{-1} = 1 \ m^{-1} * m = 1$
<proof>

lemma *mat3-inverse-cancel:*

fixes $m :: \text{mat3}$
assumes $\text{mat-det } m = 1$
shows $m * m^{-1} = 1 \ m^{-1} * m = 1$
<proof>

lemma *mat2-inverse-cancel-left:*

fixes $m a :: \text{mat2}$
assumes $\text{mat-det } m = 1$
shows $m * (m^{-1} * a) = a \ m^{-1} * (m * a) = a$
<proof>

lemma *mat3-inverse-cancel-left:*

fixes $m a :: \text{mat3}$
assumes $\text{mat-det } m = 1$
shows $m * (m^{-1} * a) = a \ m^{-1} * (m * a) = a$
<proof>

lemma *mat2-inverse-cancel-right:*

fixes $m a :: \text{mat2}$
assumes $\text{mat-det } m = 1$
shows $a * (m * m^{-1}) = a \ a * (m^{-1} * m) = a$
<proof>

lemma *mat3-inverse-cancel-right:*

fixes $m a :: \text{mat3}$
assumes $\text{mat-det } m = 1$
shows $a * (m * m^{-1}) = a \ a * (m^{-1} * m) = a$
<proof>

lemma *mat2-inversable-cancel-left:*

fixes $m a1 a2 :: \text{mat2}$
assumes $\text{mat-det } m = 1$
assumes $m * a1 = m * a2$

shows $a1 = a2$
 $\langle proof \rangle$

lemma *mat3-inversible-cancel-left*:
fixes $m a1 a2 :: mat3$
assumes $mat-det\ m = 1$
assumes $m * a1 = m * a2$
shows $a1 = a2$
 $\langle proof \rangle$

lemma *mat2-inversible-cancel-right*:
fixes $m a1 a2 :: mat2$
assumes $mat-det\ m = 1$
assumes $a1 * m = a2 * m$
shows $a1 = a2$
 $\langle proof \rangle$

lemma *mat3-inversible-cancel-right*:
fixes $m a1 a2 :: mat3$
assumes $mat-det\ m = 1$
assumes $a1 * m = a2 * m$
shows $a1 = a2$
 $\langle proof \rangle$

lemma *mat2-inverse-det [simp]*:
fixes $m :: mat2$
shows $mat-det\ (m^{-1}) = mat-det\ m$
 $\langle proof \rangle$

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

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

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

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

<proof>

lemma *mat3-special-preserves-zero*:

fixes $u :: \text{mat3}$

fixes $v :: \text{vec3}$

assumes $\text{mat-det } u = 1$

shows $u \$ v = 0 \iff v = 0$

<proof>

end

3 Properties of quadratic forms and their equivalences

theory *Quadratic-Forms*

imports *Complex-Main Low-Dimensional-Linear-Algebra*

begin

consts $qf\text{-app} :: 'a \Rightarrow 'b \Rightarrow \text{int}$ (**infixl** \$\$ 65)

definition $qf2\text{-app} :: \text{mat2} \Rightarrow \text{vec2} \Rightarrow \text{int}$ **where**

$qf2\text{-app } m v = \langle v \mid m \$ v \rangle$

adhoc-overloading $qf\text{-app } qf2\text{-app}$

definition $qf3\text{-app} :: \text{mat3} \Rightarrow \text{vec3} \Rightarrow \text{int}$ **where**

$qf3\text{-app } m v = \langle v \mid m \$ v \rangle$

adhoc-overloading $qf\text{-app } qf3\text{-app}$

lemma $qf2\text{-app-zero}$ [*simp*]:

fixes $m :: \text{mat2}$

shows $m \$ 0 = 0$

<proof>

lemma $qf3\text{-app-zero}$ [*simp*]:

fixes $m :: \text{mat3}$

shows $m \$ 0 = 0$

<proof>

consts $qf\text{-positive-definite} :: 'a \Rightarrow \text{bool}$

definition $qf2\text{-positive-definite} :: \text{mat2} \Rightarrow \text{bool}$ **where**

$qf2\text{-positive-definite } m = (\forall v. v \neq 0 \longrightarrow m \$ v > 0)$

adhoc-overloading $qf\text{-positive-definite } qf2\text{-positive-definite}$

definition $qf3\text{-positive-definite} :: \text{mat3} \Rightarrow \text{bool}$ **where**

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

adhoc-overloading *qf-positive-definite* *qf3-positive-definite*

lemma *qf2-positive-definite-positive*:

fixes $m :: \text{mat2}$

assumes *qf-positive-definite* m

shows $\forall v. m \ \$\$ v \geq 0$

<proof>

lemma *qf3-positive-definite-positive*:

fixes $m :: \text{mat3}$

assumes *qf-positive-definite* m

shows $\forall v. m \ \$\$ v \geq 0$

<proof>

consts *qf-action* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** \cdot 55)

definition *qf2-action* :: $\text{mat2} \Rightarrow \text{mat2} \Rightarrow \text{mat2}$ **where**

qf2-action $a \ u = u^T * a * u$

adhoc-overloading *qf-action* *qf2-action*

definition *qf3-action* :: $\text{mat3} \Rightarrow \text{mat3} \Rightarrow \text{mat3}$ **where**

qf3-action $a \ u = u^T * a * u$

adhoc-overloading *qf-action* *qf3-action*

lemma *qf2-action-id*:

fixes $a :: \text{mat2}$

shows $a \cdot 1 = a$

<proof>

lemma *qf3-action-id*:

fixes $a :: \text{mat3}$

shows $a \cdot 1 = a$

<proof>

lemma *qf2-action-mul* [*simp*]:

fixes $a \ u \ v :: \text{mat2}$

shows $a \cdot (u * v) = (a \cdot u) \cdot v$

<proof>

lemma *qf3-action-mul* [*simp*]:

fixes $a \ u \ v :: \text{mat3}$

shows $a \cdot (u * v) = (a \cdot u) \cdot v$

<proof>

consts *qf-equiv* :: $'a \Rightarrow 'a \Rightarrow \text{bool}$ (**infix** \sim 65)

definition *gf2-equiv* :: *mat2* \Rightarrow *mat2* \Rightarrow *bool* **where**
gf2-equiv *a b* = ($\exists u. \text{mat-det } u = 1 \wedge a \cdot u = b$)

adhoc-overloading *gf-equiv gf2-equiv*

definition *gf3-equiv* :: *mat3* \Rightarrow *mat3* \Rightarrow *bool* **where**
gf3-equiv *a b* = ($\exists u. \text{mat-det } u = 1 \wedge a \cdot u = b$)

adhoc-overloading *gf-equiv gf3-equiv*

lemma *gf2-equiv-sym-impl*:

fixes *a b* :: *mat2*

shows $a \sim b \implies b \sim a$

<proof>

lemma *gf3-equiv-sym-impl*:

fixes *a b* :: *mat3*

shows $a \sim b \implies b \sim a$

<proof>

lemma *gf2-equiv-sym*:

fixes *a b* :: *mat2*

shows $a \sim b \longleftrightarrow b \sim a$

<proof>

lemma *gf3-equiv-sym*:

fixes *a b* :: *mat3*

shows $a \sim b \longleftrightarrow b \sim a$

<proof>

lemma *gf2-equiv-trans*:

fixes *a b c* :: *mat2*

assumes $a \sim b$

assumes $b \sim c$

shows $a \sim c$

<proof>

lemma *gf3-equiv-trans*:

fixes *a b c* :: *mat3*

assumes $a \sim b$

assumes $b \sim c$

shows $a \sim c$

<proof>

lemma *gf2-action-app [simp]*:

fixes *a u* :: *mat2*

fixes *v* :: *vec2*

shows $(a \cdot u) \ \$\$ v = a \ \$\$ (u \$ v)$

<proof>

lemma *gf3-action-app [simp]*:
 fixes $a\ u :: \text{mat3}$
 fixes $v :: \text{vec3}$
 shows $(a \cdot u)\ \$\$ v = a\ \$\$ (u\ \$ v)$
<proof>

lemma *gf2-equiv-preserves-positive-definite*:
 fixes $a\ b :: \text{mat2}$
 assumes $a \sim b$
 shows $\text{gf-positive-definite } a \longleftrightarrow \text{gf-positive-definite } b$
<proof>

lemma *gf3-equiv-preserves-positive-definite*:
 fixes $a\ b :: \text{mat3}$
 assumes $a \sim b$
 shows $\text{gf-positive-definite } a \longleftrightarrow \text{gf-positive-definite } b$
<proof>

lemma *gf2-equiv-preserves-sym*:
 fixes $a\ b :: \text{mat2}$
 assumes $a \sim b$
 shows $\text{mat2-sym } a \longleftrightarrow \text{mat2-sym } b$
<proof>

lemma *gf3-equiv-preserves-sym*:
 fixes $a\ b :: \text{mat3}$
 assumes $a \sim b$
 shows $\text{mat3-sym } a \longleftrightarrow \text{mat3-sym } b$
<proof>

lemma *gf2-equiv-preserves-det*:
 fixes $a\ b :: \text{mat2}$
 assumes $a \sim b$
 shows $\text{mat-det } a = \text{mat-det } b$
<proof>

lemma *gf3-equiv-preserves-det*:
 fixes $a\ b :: \text{mat3}$
 assumes $a \sim b$
 shows $\text{mat-det } a = \text{mat-det } b$
<proof>

lemma *gf2-equiv-preserves-range-subset*:
 fixes $a\ b :: \text{mat2}$
 assumes $a \sim b$
 shows $\text{range } ((\$\$) b) \subseteq \text{range } ((\$\$) a)$
<proof>

lemma *qf3-equiv-preserves-range-subset*:
fixes $a\ b :: \text{mat3}$
assumes $a \sim b$
shows $\text{range } ((\$\$) b) \subseteq \text{range } ((\$\$) a)$
 $\langle \text{proof} \rangle$

lemma *qf2-equiv-preserves-range*:
fixes $a\ b :: \text{mat2}$
assumes $a \sim b$
shows $\text{range } ((\$\$) a) = \text{range } ((\$\$) b)$
 $\langle \text{proof} \rangle$

lemma *qf3-equiv-preserves-range*:
fixes $a\ b :: \text{mat3}$
assumes $a \sim b$
shows $\text{range } ((\$\$) a) = \text{range } ((\$\$) b)$
 $\langle \text{proof} \rangle$

Lemma 1.1 from [1].

lemma *qf2-positive-definite-criterion*:
fixes a
assumes $\text{mat-sym } a$
shows $\text{qf-positive-definite } a \longleftrightarrow \text{mat2}_{11} a > 0 \wedge \text{mat-det } a > 0$
 $\langle \text{proof} \rangle$

lemma *congruence-class-close*:
fixes $k\ m :: \text{int}$
assumes $m > 0$
shows $\exists t. 2 * |k + m * t| \leq m \ (\text{is } \exists t. ?P t)$
 $\langle \text{proof} \rangle$

Lemma 1.2 from [1].

lemma *lemma-1-2*:
fixes $b :: \text{mat2}$
assumes $\text{mat-sym } b$
assumes $\text{qf-positive-definite } b$
shows $\exists a. a \sim b \wedge$
 $2 * |\text{mat2}_{12} a| \leq \text{mat2}_{11} a \wedge$
 $\text{mat2}_{11} a \leq (2 / \text{sqrt } 3) * \text{sqrt } (\text{mat-det } a) \ (\text{is } \exists a. ?P a)$
 $\langle \text{proof} \rangle$

Theorem 1.2 from [1].

theorem *qf2-det-one-equiv-canonical*:
fixes $f :: \text{mat2}$
assumes $\text{mat-sym } f$
assumes $\text{qf-positive-definite } f$
assumes $\text{mat-det } f = 1$
shows $f \sim 1$

<proof>

Lemma 1.3 from [1].

lemma *lemma-1-3*:

fixes $a :: \text{mat3}$

assumes *mat-sym a*

defines $a' \equiv$

$$\begin{aligned} & \text{mat2} \\ & (\text{mat3}_{11} a * \text{mat3}_{22} a - (\text{mat3}_{12} a)^2) (\text{mat3}_{11} a * \text{mat3}_{23} a - \text{mat3}_{12} a * \\ \text{mat3}_{13} a) \\ & (\text{mat3}_{11} a * \text{mat3}_{23} a - \text{mat3}_{12} a * \text{mat3}_{13} a) (\text{mat3}_{11} a * \text{mat3}_{33} a - \\ (\text{mat3}_{13} a)^2) \end{aligned}$$

defines $d' \equiv$

$$\begin{aligned} & \text{mat-det} (\\ & \text{mat2} \\ & (\text{mat3}_{11} a) (\text{mat3}_{12} a) \\ & (\text{mat3}_{12} a) (\text{mat3}_{22} a) \\ &) \end{aligned}$$

shows

$$\text{mat-det } a' = \text{mat3}_{11} a * \text{mat-det } a \text{ (is } ?P)$$

$$\bigwedge x. \text{mat3}_{11} a * (a \ \$\$ x) =$$

$$(\text{mat3}_{11} a * \text{vec3}_1 x + \text{mat3}_{12} a * \text{vec3}_2 x + \text{mat3}_{13} a * \text{vec3}_3 x)^2 +$$

$$(a' \ \$\$ (\text{vec2} (\text{vec3}_2 x) (\text{vec3}_3 x))) \text{ (is } \bigwedge x. ?Q x)$$

$$\text{qf-positive-definite } a \implies \text{qf-positive-definite } a'$$

$$\text{qf-positive-definite } a \iff \text{mat3}_{11} a > 0 \wedge d' > 0 \wedge \text{mat-det } a > 0$$

<proof>

Lemma 1.4 from [1].

lemma *lemma-1-4*:

fixes $b :: \text{mat3}$

fixes $v' :: \text{mat2}$

fixes $r s :: \text{int}$

assumes *mat-sym b*

assumes *qf-positive-definite b*

assumes *mat-det v' = 1*

defines $b' \equiv$

$$\begin{aligned} & \text{mat2} \\ & (\text{mat3}_{11} b * \text{mat3}_{22} b - (\text{mat3}_{12} b)^2) (\text{mat3}_{11} b * \text{mat3}_{23} b - \text{mat3}_{12} b * \\ \text{mat3}_{13} b) \\ & (\text{mat3}_{11} b * \text{mat3}_{23} b - \text{mat3}_{12} b * \text{mat3}_{13} b) (\text{mat3}_{11} b * \text{mat3}_{33} b - \\ (\text{mat3}_{13} b)^2) \end{aligned}$$

defines $a' \equiv b' \cdot v'$

defines $v \equiv$

$$\begin{aligned} & \text{mat3} \\ & 1 \ r \ s \\ & 0 \ (\text{mat2}_{11} v') \ (\text{mat2}_{12} v') \end{aligned}$$

$$0 \text{ (mat2}_{21} \ v') \text{ (mat2}_{22} \ v')$$

defines $a \equiv b \cdot v$

shows

$$\begin{aligned} \bigwedge y. \text{mat3}_{11} \ b * (b \ \$\$ \ y) = \\ (\text{mat3}_{11} \ b * \text{vec3}_1 \ y + \text{mat3}_{12} \ b * \text{vec3}_2 \ y + \text{mat3}_{13} \ b * \text{vec3}_3 \ y)^2 + \\ (b' \ \$\$ (\text{vec2} (\text{vec3}_2 \ y) (\text{vec3}_3 \ y))) \text{ (is } \bigwedge y. \ ?P \ y) \\ \text{mat3}_{11} \ a = \text{mat3}_{11} \ b \\ \bigwedge x. \text{mat3}_{11} \ a * (a \ \$\$ \ x) = \\ (\text{mat3}_{11} \ a * \text{vec3}_1 \ x + \text{mat3}_{12} \ a * \text{vec3}_2 \ x + \text{mat3}_{13} \ a * \text{vec3}_3 \ x)^2 + \\ (a' \ \$\$ (\text{vec2} (\text{vec3}_2 \ x) (\text{vec3}_3 \ x))) \text{ (is } \bigwedge x. \ ?Q \ x) \end{aligned}$$

$\langle \text{proof} \rangle$

Lemma 1.5 from [1].

lemma *lemma-1-5*:

fixes $u_{11} \ u_{21} \ u_{31}$

assumes $\text{Gcd} \{u_{11}, u_{21}, u_{31}\} = 1$

shows $\exists u. \text{mat3}_{11} \ u = u_{11} \wedge \text{mat3}_{21} \ u = u_{21} \wedge \text{mat3}_{31} \ u = u_{31} \wedge \text{mat-det} \ u = 1$

$\langle \text{proof} \rangle$

Lemma 1.6 from [1].

lemma *lemma-1-6*:

fixes $c :: \text{mat3}$

assumes *mat-sym* c

assumes *qf-positive-definite* c

shows $\exists a. a \sim c \wedge$

$$\begin{aligned} 2 * (\max |\text{mat3}_{12} \ a| \ |\text{mat3}_{13} \ a|) \leq \text{mat3}_{11} \ a \wedge \\ \text{mat3}_{11} \ a \leq (4 / 3) * \text{root } 3 \ (\text{mat-det} \ a) \end{aligned}$$

$\langle \text{proof} \rangle$

Theorem 1.3 from [1].

theorem *qf3-det-one-equiv-canonical*:

fixes $f :: \text{mat3}$

assumes *mat-sym* f

assumes *qf-positive-definite* f

assumes *mat-det* $f = 1$

shows $f \sim 1$

$\langle \text{proof} \rangle$

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 $n \geq 2$
 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* ($a + m * j$) ∧ $j \neq 0$
<proof>

Lemma 1.8 from [1].

lemma *three-squares-using-mod-four*:
 fixes *n* :: *nat*
 assumes $n \bmod 4 = 2$
 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 $(3 :: \text{int}) \wedge n \bmod 8 = (\text{if even } n \text{ then } 1 \text{ else } 3)$
<proof>

Lemma 1.9 from [1].

lemma *three-squares-using-mod-eight*:
 fixes *n* :: *nat*
 assumes $n \bmod 8 \in \{1, 3, 5\}$
 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 $n^2 \bmod 8 \in \{0, 1, 4\}$

<proof>

lemma *power-two-mod-four*:

fixes $n :: nat$

shows $n^2 \bmod 4 \in \{0, 1\}$

<proof>

Theorem 1.4 from [1].

theorem *three-squares-iff*:

fixes $n :: nat$

shows $(\exists x_1 x_2 x_3. n = x_1^2 + x_2^2 + x_3^2) \longleftrightarrow (\nexists a k. n = 4^a * (8 * k + 7))$

<proof>

Theorem 1.5 from [1].

theorem *odd-three-squares-using-mod-eight*:

fixes $n :: nat$

assumes $n \bmod 8 = 3$

shows $\exists x_1 x_2 x_3. \text{odd } x_1 \wedge \text{odd } x_2 \wedge \text{odd } x_3 \wedge n = x_1^2 + x_2^2 + x_3^2$

<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.