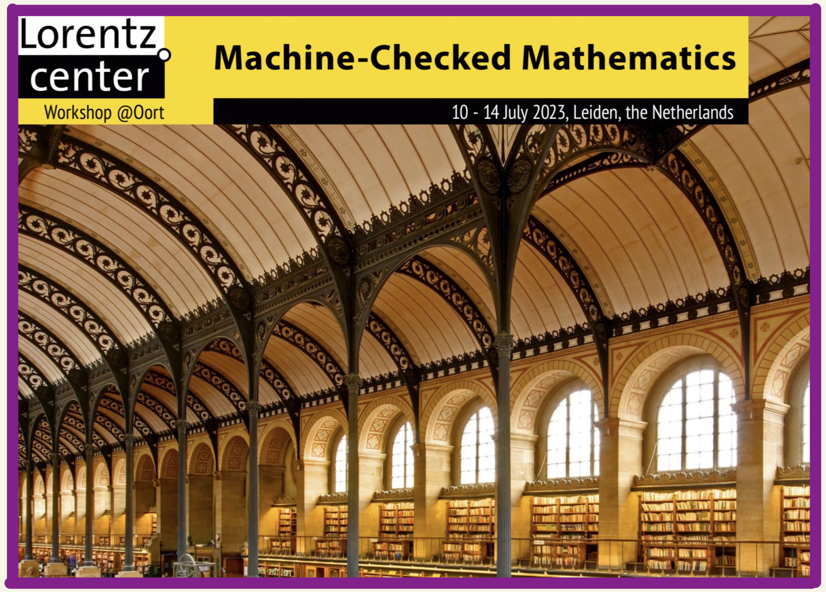
Formally real fields



Riccardo Brasca :: [Mahoor Alavioun, Maryam Emamjomeh Zadeh, Ignasi Sánchez Rodríguez, Florent Schaffhauser]

10-14 July 2023

Definition

A field F is formally real if -1 is not a sum of squares in F.

$$\neg \left(\exists n : \mathbb{N},\ \exists x_1, \ldots, x_n : F,\ -1 = x_1^2 + \cdots + x_n^2 \right)$$

Examples

$F = \mathbb{R}$, $\overline{\mathbb{Q}} \cap \mathbb{R}$, $\mathbb{Q}(\sqrt{2})$, $\mathbb{R}(t)$, real Puiseux series, ...

Theorem (Artin- Schreier theory)

If F is formally real, then there exists a linear ordering on F.

Definition

A ring R is formally real if:

$$\forall n : \mathbb{N}, \ \forall x_1, \dots, x_n : R,$$

$$x_1^2 + \dots + x_n^2 = 0 \to \forall i, \ x_i = 0.$$

Example

1. A linearly ordered ring is formally real.
2. If the ring R is a field, the condition of being formally real is equivalent to the previous one.
3. A formally real (non- trivial) ring has characteristic zero.

Theorem (Artin-Schreier theory)

If F is formally real, then there exists a linear ordering on F.

Proof

sub-semiring $P \subset F$ such that
$\forall x \in F,\ x^2 \in P$ and $-1 \notin P$

Consider the set of positive cones on F. It is inductively ordered by inclusion.

Since F is formally real, that set is non-empty (sums of squares form a positive cone). By Zorn, it admits a maximal element.

A maximal positive cone induces a linear ordering.

$a \leq_P b$ if $b - a \in P$

$\square$

Formal definitions

1. Being formally real

```
def sum_of_squares {R : Type _} [Semiring R] : List R → R
  | [] => 0
  | (a :: L) => (a ^ 2) + (sum_of_squares L)

example : sum_of_squares [1, -2, 3] = 14 := rfl
```

```
@[mk_iff]
class IsFormallyReal (R : Type _) [Semiring R] : Prop where
  is_formally_real : ∀ L : List R, sum_of_squares L = 0 → (∀ x ∈ L, x = 0)
```

```
theorem formally_real_semifield_equiv {F : Type _} [Semifield F] :
  (IsFormallyReal F) ↔ ¬ (∃ L : List F, 1 + sum_of_squares L = 0) := by
classical
constructor
· exact one_add_sum_of_squares_neq_zero
· exact sum_of_sq_eq_zero_iff_all_zero
done
```

non-trivial semiring ok

2. The set of positive cones

```
def squares (A : Type _) [Semiring A] : Set A := {a | ∃ (b : A), a = b ^ 2}

def cone_of_squares (A : Type _) [Semiring A] := AddSubmonoid.closure (squares A)
```

```
def PositiveCones (A : Type _) [Ring A] :=
  { P : Subsemiring A | squares A ⊆ P ∧ -1 ∉ P }
```

Implementation of other approaches

```
lemma IsFormallyReal_iff_Fin (R : Type _) [Semiring R] : IsFormallyReal R ↔
    ∀ (n : ℕ), ∀ (f : Fin n → R), (∑ i, (f i) ^ 2 = 0) → (∀ i, f i = 0) := by
  refine' ⟨fun h n f hf i => _, fun h => ⟨fun L => List.ofFnRec (fun n f H a ha => _) L⟩⟩
  · refine' h.is_formally_real (List.ofFn f) _ (f i) (by simp [List.mem_ofFn])
    simp [sum_of_squares, sum_of_squares_of_list, List.sum_ofFn, hf]
  · rw [sum_of_squares_of_list, List.map_ofFn, List.sum_ofFn] at H
    obtain ⟨j, rfl⟩ := (List.mem_ofFn _ _).1 ha
    exact h n f H j

lemma IsFormallyReal_iff_Multiset (R : Type _) [Semiring R] : IsFormallyReal R ↔
    ∀ (M : Multiset R), (M.map (.^2)).sum = 0 → (∀ x ∈ M, x = 0) := by
  refine' ⟨fun h M hM x hx => _, fun h => ⟨fun L hL x hx => _⟩⟩
  · refine' h.is_formally_real M.toList _ x (Multiset.mem_toList.2 hx)
    convert hM
    rw [sum_of_squares_of_list]
    conv_rhs => rw [← Multiset.coe_toList M]
    rw [Multiset.coe_map, Multiset.coe_sum]
  · refine' h L _ _ (by simp [hx])
    convert hL
    simp [sum_of_squares_of_list]
```

In Mathlib

1. The notion of positive cone is formalized differently

```
structure PositiveCone (α : Type _) [AddCommGroup α] where
  nonneg : α → Prop
  pos : α → Prop := fun a => nonneg a ∧ ¬nonneg (-a)
  pos_iff : ∀ a, pos a ↔ nonneg a ∧ ¬nonneg (-a) := by intros; rfl
  zero_nonneg : nonneg 0
  add_nonneg : ∀ {a b}, nonneg a → nonneg b → nonneg (a + b)
  nonneg_antisymm : ∀ {a}, nonneg a → nonneg (-a) → a = 0
```

```
structure PositiveCone (α : Type _) [Ring α] extends AddCommGroup.PositiveCone α where
  /-- In a positive cone, `1` is `nonneg` -/
  one_nonneg : nonneg 1
  /-- In a positive cone, if `a` and `b` are `pos` then so is `a * b` -/
  mul_pos : ∀ a b, pos a → pos b → pos (a * b)
```

```
structure TotalPositiveCone (α : Type _) [AddCommGroup α] extends PositiveCone α where
  /-- For any `a` the proposition `nonneg a` is decidable -/
  nonnegDecidable : DecidablePred nonneg
  /-- Either `a` or `-a` is `nonneg` -/
  nonneg_total : ∀ a : α, nonneg a ∨ nonneg (-a)
```

In Mathlib

2. Positivity of squares in linearly ordered semirings?

We prove that linearly ordered rings are formally real.

In ℕ too, the squares are positive.

```
@[simp]
lemma sum_sq_nonneg {A : Type _} [LinearOrderedRing A] (L : List A) : 0  ≤ sum_of_squares L := by
  induction' L with head tail ih
  . rfl
  . apply add_nonneg
    . exact sq_nonneg head
    . exact ih
```

```
theorem sq_nonneg                                                        source
        {R : Type u_1} [inst : LinearOrderedRing R] (a : R) :
    0 ≤ a ^ 2
```

Mathlib

```
instance {A : Type _} [LinearOrderedRing A] : IsFormallyReal A where
  is_formally_real := fun (L : List A) (sum_sq_zero: sum_of_squares L = 0) ↦ by
    intro a a_in_L
    by_contra c
    have a_sq_pos : 0 < a ^ 2 := by exact Iff.mpr (sq_pos_iff a) c
    have h : a ^ 2 + sum_of_squares (L.erase a) = sum_of_squares L := by
      exact Eq.symm (sum_of_squares_erase L a a_in_L)
    rw [sum_sq_zero] at h
    have sum_sq_nonneg : 0 ≤ sum_of_squares (L.erase a) := by simp
    have sum_sq_pos: 0 < a ^ 2 + sum_of_squares (L.erase a) := by
      exact add_pos_of_pos_of_nonneg a_sq_pos sum_sq_nonneg
    have : a ^ 2 + sum_of_squares (L.erase a) ≠ 0 := by exact ne_of_gt sum_sq_pos
    contradiction
```

The final definition

FormallyReal.lean > ...

```lean

instance LinearOrderedRing.isFormallyReal (A : Type _) [LinearOrderedRing A] :
    IsFormallyReal A where
  is_formally_real := fun (L : List A) (sum_sq_zero: sum_of_squares L = 0) ↦ by
    intro a a_in_L
    by_contra c
    have a_sq_pos : 0 < a ^ 2 := by exact Iff.mpr (sq_pos_iff a) c
    have h : a ^ 2 + sum_of_squares (L.erase a) = sum_of_squares L := by
      exact Eq.symm (sum_of_squares_erase L a a_in_L)
    rw [sum_sq_zero] at h
    have sum_sq_nonneg : 0 ≤ sum_of_squares (L.erase a) := by simp
    have sum_sq_pos: 0 < a ^ 2 + sum_of_squares (L.erase a) := by
      exact add_pos_of_pos_of_nonneg a_sq_pos sum_sq_nonneg
    have : a ^ 2 + sum_of_squares (L.erase a) ≠ 0 := by exact ne_of_gt sum_sq_pos
    contradiction

noncomputable
def IsFormallyReal.toLinearOrderedRing {F : Type _} [Field F] [IsFormallyReal F] :
    LinearOrderedRing F :=
  LinearOrderedRing.mkOfPositiveCone (IsFormallyReal.toTotalPositiveCone F)

#print axioms LinearOrderedRing.isFormallyReal    'LinearOrderedRing.isFormallyReal' depends on axioms: [C
#print axioms IsFormallyReal.toLinearOrderedRing    'IsFormallyReal.toLinearOrderedRing' depends on axioms

```

▼ FormallyReal.lean:599:0

▼ Messages (1)

▼ FormallyReal.lean:599:0

```
'IsFormallyReal.toLinearOrderedRing'
depends on axioms: [propext,
Classical.choice, Quot.sound]
```

▶ All Messages (11)