# Euclidean Domains

Juilan Kornweibel, Moritz Hlavatý

Computer-assisted Mathematics

23.07.2024

# Table of Contents

# PID - Definitions

## 1.1 Definition: ring

A **ring** is a tuple $(R,\ 0,\ 1,\ +,\ \cdot)$ of a set $R$ with elements $0, 1 \in R$ and two binary operators $+, \ \cdot : R \times R \to R$, for which the following holds true:

- $(R,\ 0,\ +)$ is an abelian group
- $\cdot$ is associative, meaning $\forall r, s, t \in R: \ (r \cdot s) \cdot t = r \cdot (s \cdot t)$
- 1 is neutral element for $\cdot$, meaning $\forall r \in R: \ 1 \cdot r = r \cdot 1 = r$
- The distributive properties hold true for $+$ and $\cdot$: $\forall v, s, t \in R : (r + s) \cdot t = r \cdot t + s \cdot t$ and $r \cdot (s + t) = r \cdot s + r \cdot t$

A ring is called commutative ring $\Leftrightarrow \forall r, s \in R : r \cdot s = s \cdot r$

### 1.2 Definition: integral domain

An **integral domain** is a nonzero ring R, for which the following property holds true: $\forall r, s \in R \backslash \{0\} : r \cdot s \neq 0$

## 1.3 Definition: ideal

An **left ideal** is a subset $I \subset R$ with $R$ being a Ring, such that:

- $0 \in I$
- $\forall r, s \in I : r + s \in I$
- $\forall r \in R,\ \forall s \in I : r \cdot s \in I$

An **right ideal** is a subset $I \subset R$, for which all properties above are true, with the last one being modified to:

- $\forall r \in R,\ \forall s \in I : s \cdot r \in I$

If both the orignial and the modified porperty hold true for an ideal $I$ it is called a two-sided ideal.
If $R$ is a commutative Ring we just call $I$ an ideal because $r \cdot s = s \cdot r,\ \forall r, s \in R$

### 1.4 Definition: principle ideal

A **principal ideal** is an Ideal I over a commutative Ring R, such that:
$\exists a \in I : I = (a) := R \cdot a := \{r \cdot a | \ r \in R\}$

```
ideal_principal : ∀ (I : Ideal R), ∃ (x : R), Ideal.span {x} = I
```

## 1.5 Definition: principle ideal domain

A **principal ideal domain** (PID) is a Ring $R$ with the following properties:

- $R$ is a integral domain
- every ideal $I$ of $R$ is a principal ideal

```
structure IsPID (R : Type) [CommRing R] : Prop where
  isDomain : IsDomain R
  ideal_principal : ∀ (I : Ideal R), ∃ (x : R), Ideal.span {x} = I
```

# PID - Theroem

**1.6 Theorem: fields are pid's**

Let $K$ be a Field. It suffices to show that $I = (0)$ and $J = (1)$ are the only ideals of $K$ and therefor every ideal is a principal ideal.
Let $I = \{0\} \Rightarrow 0 \in I$
And therefor $I$ is a principal ideal.
Let $a \in K \Rightarrow \exists a^{-1}$ with $a^{-1} \cdot a = 1$
$\Rightarrow \forall a \in K : a \in (1) = J$
And therefor (0) and (1) are the only Ideals of any Field K and both are principal ideals.

```
-- Fields are PID's
lemma isPID_of_field (k : Type) [Field k] : IsPID k where
  isDomain := inferInstance
  ideal_principal := by
    intro I
    by_cases h : I = 0
    -- Case 1: I = 0
    · subst h
      use 0
      simp
    -- Case 2: I ≠ 0
    · simp at h   --i dont think this does much...
      have h2 : ∃ x ∈ I, x ≠ 0 := by
        --since k is a field and I is a non zero ideal, it must contain a non zero element
        -- exact?
        exact Submodule.exists_mem_ne_zero_of_ne_bot h

      -- Let x be a nonzero element of I
      obtain ⟨x, hx, hnezero⟩ := h2
      use x
      apply Ideal.ext
      intro y
```

```
constructor
· intro
  have hxu: IsUnit x := by {
    rw[isUnit_iff_ne_zero]
    exact hnezero
  }
  have h2 : I = ⊤ := by exact Ideal.eq_top_of_isUnit_mem I hx hxu
  rw[h2]
  exact trivial

· intro
  have hxu: IsUnit x := by {
    rw[isUnit_iff_ne_zero]
    exact hnezero
  }
  rw[← Ideal.span_singleton_eq_top] at hxu
  rw[hxu]
  exact trivial
```

## 2.1 Definition: euclidean function

A **euclidean function** is a function $\beta : R\backslash\{0\} \to \mathbb{N}_0$ with the following porperty: $\forall x, y \in R$ with $y \neq 0$ $\exists q, r \in R$ such that $x = q \cdot y + r$ and $(r = 0 \vee \beta(r) < \beta(y))$

```
-- Euclidean Function
structure EuclideanFunction ( R : Type) [CommRing R] where
  /-- Height function. -/
  height : R → WithBot ℕ
  zero_of_bot (x : R) : height x = ⊥ → x = 0
  /-- Division by zero -/
  division (a b : R) (hb : b ≠ 0) : ∃ q r, a = b * q + r ∧ (r = 0 ∨ height r < height b)
```

## 2.2 Definition: euclidean domain

A **euclidean domain** is an integral domain with a euclidean function.

```
-- Euclidean domain
structure IsEuclideanDomain (R : Type) [CommRing R] : Prop where
  isDomain : IsDomain R
  exists_euclideanFunction : Nonempty (EuclideanFunction R)
```

```
def euclideanOfField (k : Type) [Field k] : EuclideanFunction k where
  height _ := 42
  zero_of_bot x h := by simp_all;/- absurd h; decide-/
  division a b hb := by
    use a / b
    use 0
    /- found by `simp?` -/
    simp only [add_zero, lt_self_iff_false, or_false, and_true]
    field_simp
```

```
-- Fields are euclidiean domains
theorem isEuclidean_of_field (k : Type) [Field k] : IsEuclideanDomain k where
  isDomain := inferInstance
  exists_euclideanFunction := ⟨euclideanOfField k⟩
```

```
def Int.euclidean : EuclideanFunction ℤ where
  height := λ n => n.natAbs
  zero_of_bot := by
    intro a
    simp
  division a b hb := by
    let q := a / b
    let r := a % b

    -- Proof that a = b * q + r
    have h1 : a = b * q + r := by{
      nth_rewrite 1 [← Int.emod_add_ediv a b, add_comm]
      rfl
    }
    --proof 0 ≤ r
    have h2 : 0 ≤ r := by{
      --exact?
      exact emod_nonneg a hb
    }
        --proof r = |r|
    have h3 :   r = |r| := by{
      rw [← abs_eq_self] at h2
      symm
      exact h2}
```

```
--proof |r| < |b|
have h4 :  natAbs r < natAbs b := by{
  zify
  rw[← h3]
  exact emod_lt a hb
}
use q, r
constructor
· apply h1
· right
  simp
  exact h4
```

```
a b : ℤ
hb : b ≠ 0
q : ℤ := a / b
r : ℤ := a % b
h1 : a = b * q + r
h2 : 0 ≤ r
h3 : r = |r|
h4 : r.natAbs < b.natAbs
⊢ ∃ q r, a = b * q + r ∧ (r
= 0 ∨ (fun n => ↑n.natAbs)
r < (fun n => ↑n.natAbs) b)
```

```
theorem Int.isEuclidean : IsEuclideanDomain ℤ where
  isDomain := inferInstance
  exists_euclideanFunction := ⟨Int.euclidean⟩
```

# Bibliography


- Böckle G., (summer Semester 2024) Lecture Notes Linear Algebra 2
- Proof Wiki, accessed: 22th July 2024, https://proofwiki.org/wiki/Main_Page