Higher Algebraic Structures

Groups, Rings, Fields, and Vector Spaces

Kleis Language Project

1 Higher Algebraic Structures: Groups, Rings, Fields, and Vector Spaces

This section extends the algebraic hierarchy beyond monoids. The presentation follows the classical structural method of Bourbaki: each structure is defined by a carrier set together with operations and equational axioms.

1.1 Groups

Definition 1 (Group)**.** *A* group *is a triple* $(G, \cdot, e)$ *where:*

1. $(G, \cdot, e)$ *is a monoid;*
2. *every element* $x \in G$ *admits a (two-sided) inverse, i.e., there exists* $x^{-1} \in G$ *such that*

$$x^{-1} \cdot x = e \qquad \text{and} \qquad x \cdot x^{-1} = e.$$

Example 1. *The integers* $(\mathbb{Z}, +, 0)$ *form a group, where* $x^{-1} = -x$.

Definition 2 (Abelian group)**.** *A group is* abelian *if its multiplication is commutative:*

$$x \cdot y = y \cdot x \qquad \text{for all } x, y \in G.$$

Example 2. $(\mathbb{Z}, +)$ *is abelian;* $(S_n, \circ)$*, the symmetric group, is not.*

1.2 Rings

Definition 3 (Ring)**.** *A* ring *is a triple* $(R, +, \times)$ *such that:*

1. $(R, +)$ *is an abelian group, with identity* 0 *and inverse* $-x$*;*
2. $(R, \times)$ *is a monoid, with identity* 1*;*
3. *multiplication distributes over addition:*

$$x \times (y + z) = (x \times y) + (x \times z), \qquad (x + y) \times z = (x \times z) + (y \times z).$$

Remark 1. *A ring need not be commutative under multiplication; commutative rings form a distinguished subclass.*

Example 3. *The integers* $(\mathbb{Z}, +, \times)$ *form a commutative ring. The* $n \times n$ *matrices with real entries form a (noncommutative) ring under matrix addition and multiplication.*

1.3 Fields

Definition 4 (Field)**.** *A* field *is a commutative ring* $(F, +, \times)$ *in which every nonzero element admits a multiplicative inverse. Thus:*

$$x \neq 0 \implies \exists\, x^{-1} \in F \text{ such that } x \times x^{-1} = 1.$$

Example 4. $\mathbb{Q}$, $\mathbb{R}$, *and* $\mathbb{C}$ *are fields. The integers are not a field, since only* ± 1 *are invertible.*

Remark 2. *Fields form the algebraic basis of linear algebra and support division and scalar multiplication in vector spaces.*

1.4 Vector Spaces

Definition 5 (Vector space)**.** *Let* F *be a field. A* vector space *over* F *is a pair* $(V, +)$ *together with a scalar-multiplication operation*

$$\cdot : F \times V \to V$$

such that:

1. $(V, +)$ *is an abelian group with identity* 0_v*;*
2. *scalar multiplication satisfies:*

$$\begin{aligned} a \cdot (v + w) &= (a \cdot v) + (a \cdot w), \\ (a + b) \cdot v &= (a \cdot v) + (b \cdot v), \\ (ab) \cdot v &= a \cdot (b \cdot v), \\ 1 \cdot v &= v, \end{aligned}$$

for all $a, b \in F$ *and* $v, w \in V$*.*

Example 5. $\mathbb{R}^n$ *is a vector space over the field* $\mathbb{R}$*. Matrices of size* $m \times n$ *form a vector space over* $\mathbb{R}$ *under entrywise addition and scalar multiplication.*

1.5 Structural Relationships

The hierarchy of algebraic structures can be summarized as:

$$\text{Group} \supset \text{Monoid} \supset \text{Semigroup} \supset \text{Magma},$$

and

$$\text{Field} \supset \text{Commutative Ring} \supset \text{Ring}.$$

Vector spaces are defined over fields:

$$\text{Vector Space}(V) \text{ is defined over a Field } F.$$

In categorical terms:

- a monoid is a category with one object;
- a group is a groupoid with one object;
- a ring is a rig with additive inverses;
- a field is a commutative ring in which all nonzero arrows are invertible.

1.6 Interpretation in Kleis

The Kleis algebraic hierarchy mirrors these classical structures. In Kleis, each structure is specified by its operations and axioms:

```
structure Group(G) extends Monoid(G) {
  operation inv : G → G
  axiom left_inverse:
    (x : G). inv(x) • x = e
}

structure Ring(R) {
  structure additive : AbelianGroup(R)
  structure multiplicative : Monoid(R)
  axiom distributivity:
    (x y z : R). x × (y + z) = (x × y) + (x × z)
}

structure Field(F) extends Ring(F) {
  operation inverse : F → F
  axiom multiplicative_inverse:
    (x : F) where x  zero.
      inverse(x) × x = one
}

structure VectorSpace(V) over Field(F) {
  operation (+) : V × V → V
  operation (·) : F × V → V
}
```