Groups and Rings

Carter Aitken

2025-05-05

Abstract

We're studying abstract algebra, specifically groups and rings.

Contents

Chapter 1

Operations

- $\mathbb{N} \quad +, \cdot$
- $\mathbb{Z} \quad +, \cdot, -$
- $\mathbb{Q} \quad +, \cdot, -$
- $\mathbb{R} \quad +, \cdot, -$
- $\mathbb{C} \quad +, \cdot, -, x \mapsto \overline{x}, x \mapsto \sqrt{x}$
- $\text{(Vectors)} \quad +, \text{(scalarmul)}$
- $\text{(Matricies)} \quad +, \text{(scalarmul)}, \text{(matrixmul)}$
- $\text{(polynomials)} \quad +, \cdot$

In abstract algebra, we're iinterested in what notions of "numbers" exists.

The different "types" of numbers really are distinguished by the operations on them. In this class we'll stick with operating on sets.

Definition 1.0.1: Binary Operations. *A* ***binary operation*** *on a set X is a function $b : X \times X \to X$.*

Note: we often write binary operators inline (like in Haskell).

We could use $+, \cdot, \times, \div, \otimes, \boxtimes, \oplus, \boxplus, \diamond$

Definition 1.0.2. *a* ***k-ary operator*** *on X is a func $f : \underbrace{X \times \cdots \times X}_{k} \to X$.*

$x \mapsto \frac{1}{x}$ on $\mathbb{Q}$ isn't a unary operation b/c $\frac{1}{0}$ isn't defined.
$\mathbb{Q}^\times = \{x \in \mathbb{Q} : x \neq 0\}$ does have the reciprocal as a binary operator, but not minus.

Definition 1.0.3. *a binary operator $\boxtimes$ on X is **associative** if*

$$x \boxtimes (y \boxtimes z) = (x \boxtimes y) \boxtimes z, \quad \forall x, y, z \in X$$

$+, \cdot$ on $\mathbb{N}, \mathbb{Z}$ are associative. $- : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ isn't associative. Neither is $\div : \mathbb{Q}^\times \times \mathbb{Q}^\times \to \mathbb{Q}^\times$. Function composition is associative.

Definition 1.0.4: (Informal) Bracketing. *Let $\boxtimes$ be a bin operator on a set X. A **bracketing** of a seq $a_1, \ldots, a_n \in X$ is a way of inserting brackets into*

$$a_1 \boxtimes \cdots \boxtimes a_n \text{ s/t the expression can be evaluated}$$

Definition 1.0.5: Bracketing. *A **bracket** of $a_1, \ldots, a_n$ is*

$$\begin{aligned} &n = 1 : (\text{word})\ a_1 \\ &n > 1 : (w_1 \boxtimes w_2)\ \textit{where} \\ &\qquad w_1 \leftarrow (\text{bracket})\ \textit{of}\ a_1, \ldots a_k \\ &\qquad w_2 \leftarrow (\text{bracket})\ \textit{of}\ a_{k+1}, \ldots, a_n \end{aligned}$$

```
data Bracket t = Number t | Branch (Bracket t) (Bracket t)
evalBracket :: (t -> t -> t) -> Bracket t -> t
evalBracket fn aseq =
  case aseq of
    Number x            -> x
    Branch left' right' -> fn (evalBracket fn left')
                              (evalBracket fn right')
```

Proposition 1.0.1. *a binary operation $\boxtimes$ on X is associative **iff** for every seq $a_1, \ldots, a_n$, $n \geq 1$, every bracketing of $a_1, \ldots a_n$ evaluates to the same elem of X.*

Proof. ($\Longleftarrow$) Take $n = 3$. Then

$$(a \boxtimes b) \boxtimes c = a \boxtimes (b \boxtimes c),\ \forall a, b, c \in X$$

($\implies$) Proof by induction.

Base Case: $n = 1$. Every bracketing of a word evaluates to that same word.

Assume proposition is true for $n < k$, where $k > 1$. Let $a_1, \cdots, a_k \in X$. If w is a bracketing of $a_1, \ldots, a_k$ then $w = (w_1 \boxtimes w_2)$, where w_1 is a bracketing of $a_1, \ldots, a_l$ and w_2 is a bracketing of $a_{l+1}, \ldots, a_k$.

$$w_1 = (\cdots(a_1 \boxtimes a_2) \boxtimes \cdots) \boxtimes a_l$$

$$\begin{aligned}
w_2 &= (a_{l+1} \boxtimes (\cdots(a_{k-1} \boxtimes a_k)\cdots)) \\
w &\overset{\text{in } X}{=} w_1 \boxtimes w_2 \\
&= (A \boxtimes a_l) \boxtimes w_2 \\
&= A \boxtimes (a_l \boxtimes w_2) \text{ by assoc.} \\
\cdots &= a_1 \boxtimes (\cdots(a_{k-1} \boxtimes a_k)\cdots)
\end{aligned}$$

Hence any 2 bracketings of $a_1, \ldots, a_k$ evaluate to $a_1 \boxtimes (\cdots(a_{k-1} \boxtimes x_k)\cdots)$. By induction, the prop holds. $\square$