# GROUPS

ARTHUR RYMAN

Abstract. This article contains Z Notation definitions for groups and some related objects. It has been type checked with *f*uzz.

Contents



*Date*: August 29, 2022.

## 1. Introduction

Groups are ubiquitous throughout mathematics and physics. This article defines groups and their homomorphisms.

We build up the definition of a group in terms of some simpler, related algebraic objects, namely semigroups and monoids.

1.1. **Structures.** Semigroups, monoids, and groups are defined as sets of elements with an additional feature, namely a binary operation that has certain properties. In general, a set with some additional features is called a *mathematical structure*. In particular, semigroups, monoids, and groups are called *algebraic structures*.

In mathematical writing, authors do not normally distinguish between a set of elements and its associated mathematical structure since this rarely causes confusion. For example, one typically see statements such as: "Let $G$ be a group and let $g \in G$ be an element." However, a set of elements may be given many inequivalent structures. For example, addition and multiplication are distinct binary operations on the set of integers. In this case it is insufficient to specify only the set of elements.

In natural language, the practice of *metonymy*, which consists of referring to a thing, e.g. an algebraic structure, by naming a part of it, e.g. its set of elements, normally causes no confusion. However, an algebraic structure and its set of elements are distinct mathematical objects and so, in formal language, must be referred to using distinct names or expressions.

In order to distinguish between sets of elements and structures on them, we will adopt the practice of defining structures as being tuples consisting of the set of elements and one of more additional features.

## 2. Binary Operations

Let $\mathtt{t}$ be a set which we regard as a universe from which we draw elements. Let $elements \subseteq \mathtt{t}$ be a set of elements drawn from $\mathtt{t}$ let $op \in elements \times element \longrightarrow elements$ be an operation that combines pairs of elements to give another element. We call the pair $(elements, op)$ a *binary operation* on $elements$. Furthermore, say that it is a binary relation *in* the universe $\mathtt{t}$.

*BinaryOperation*[t]

$$elements : \mathbb{P}\,\mathtt{t}$$
$$op : \mathtt{t} \times \mathtt{t} \pfun \mathtt{t}$$

---

$$op \in elements \times elements \longrightarrow elements$$

- The set of elements is closed under the operation.

Let $binary\_operation[\mathtt{t}]$ denote the set of all binary operations in $\mathtt{t}$.

$$binary\_operation[\mathtt{t}] == \{\, BinaryOperation[\mathtt{t}] \bullet (elements, op) \,\}$$

**Example.** *Integer addition is a binary operation on $\mathbb{Z}$.*

$$(\mathbb{Z}, (\_ + \_)) \in binary\_operation[\mathbb{Z}]$$

**Example.** *Integer multiplication is a binary operation on $\mathbb{Z}$.*

$$(\mathbb{Z}, (\_ * \_)) \in binary\_operation[\mathbb{Z}]$$

Let the prefix generic notation $\mathrm{binop}\,\mathtt{t}$ denote the set of all binary operations in $\mathtt{t}$.

$$\mathrm{binop}\,\mathtt{t} == binary\_operation[\mathtt{t}]$$

2.1. **Notation.** The set of elements in a binary operation is normally denoted by variables such as $A$ or $B$. As a notational convention, we'll denote the corresponding structures by variables such as $\mathbf{A}$ or $\mathbf{B}$.

The result of applying a binary operation to a pair of elements $(x, y)$ is normally denoted by an expression formed using an infix operator symbol, e.g. $x + y$ or $x * y$.

Let $\mathtt{t}$ and $\mathtt{u}$ be sets, let $A \subseteq \mathtt{t}$ and $B \subseteq \mathtt{u}$ be subsets of elements, and let the infix expression $x * y$ denote binary operations on both $A$ and $B$. Here we follow the traditional practice of using the same symbol to denote different things when no confusion can occur. Although the symbols look the same, they are encoded distinctly at the source level, namely using the commands `\mulA` and `\mulB`. This practice makes the typeset expressions look as natural as possible while at the same time satisfying the type checker.

Let $BinaryOperation\_A$ denote the binary operation $\mathbf{A}$ where $A$ is the set of elements and $*$ (encoded as `\mulA`) is the operator.

*BinaryOperation_A*[t]

$$BinaryOperation[\mathtt{t}][A / elements, \_ * \_ / op]$$
$$\mathbf{A} : \mathrm{binop}\,\mathtt{t}$$

---

$$\mathbf{A} = (A, (\_ * \_))$$

Let $BinaryOperation\_B$ denote the binary operation $\mathbf{B}$ where $B$ is the set of elements and $*$ (encoded as `\mulB`) is the operator.

*BinaryOperation_B*[u]

$$BinaryOperation[\mathtt{u}][B / elements, \_ * \_ / op]$$
$$\mathbf{B} : \mathrm{binop}\,\mathtt{u}$$

---

$$\mathbf{B} = (B, (\_ * \_))$$

2.2. **Homomorphisms.** Let **A** and **B** be binary operations and let $f \in A \longrightarrow B$ be a function. Then $f$ is said to *preserve the operations* if it maps the product of elements of $A$ to the product of the mapped elements of $B$.

Let *MapPreservesOperation* denote this situation.

*MapPreservesOperation*[t, u]

$BinaryOperation\_A[\mathsf{t}]$
$BinaryOperation\_B[\mathsf{u}]$
$f : \mathsf{t} \pfun \mathsf{u}$

---

$f \in A \longrightarrow B$

$\forall\, x, y : A \bullet$
$\quad f(x * y) = (f\ x) * (f\ y)$

- $f$ is a total function from $A$ to $B$
- $f$ preserves the binary operations

TODO: express this examples in terms of the schema *MapPreservesOperation*.

**Example.** *Multiplication by a fixed integer $c$ maps $\mathbb{Z}$ to $\mathbb{Z}$ and preserves addition.*

$\forall\, c, x, y : \mathbb{Z} \bullet$
$\quad c * (x + y) = c * x + c * y$

**Example.** *Exponentiation by a fixed natural number $n$ maps $\mathbb{Z}$ to $\mathbb{Z}$ and preserves multiplication.*

$\forall\, n : \mathbb{N};\ x, y : \mathbb{Z} \bullet$
$\quad (x * y) ** n = x ** n * y ** n$

A map that preserves operations is said to be an *operation homomorphism*. Let $\alpha = (A, *)$ and $\beta = (B, *)$ be binary operations in $\mathsf{t}$ and $\mathsf{u}$. Let $hom\_op[\mathsf{t}, \mathsf{u}](\alpha, \beta)$ denote the set of all operation homomorphisms from $\alpha$ to $\beta$.

$hom\_op[\mathsf{t}, \mathsf{u}] ==$
$\quad (\lambda\, \alpha : \mathrm{binop}\, \mathsf{t};\ \beta : \mathrm{binop}\, \mathsf{u} \bullet$
$\quad\quad \{\, MapPreservesOperation[\mathsf{t}, \mathsf{u}] \mid$
$\quad\quad\quad \alpha = (A, (\_ * \_)) \land$
$\quad\quad\quad \beta = (B, (\_ * \_)) \bullet$
$\quad\quad\quad\quad f\, \})$

TODO: try to simplify the above by using a gendef paragraph.

**Remark.**

$hom\_op[\mathsf{T}, \mathsf{U}] \in \mathrm{binop}\, \mathsf{T} \times \mathrm{binop}\, \mathsf{U} \longrightarrow \mathbb{P}(\mathsf{T} \pfun \mathsf{U})$

Let the notation $\hom(\alpha, \beta)$, typeset using the command `\homBinOp`, denote the set of operation homomorphisms from $\alpha$ to $\beta$.

$\hom[\mathsf{t}, \mathsf{u}] == hom\_op[\mathsf{t}, \mathsf{u}]$

**Remark.** *The identity map preserves all operations.*

$$\forall A : \operatorname{binop} \mathsf{X} \bullet \\ \quad \operatorname{id} \mathsf{X} \in \hom(A, A)$$

**Remark.** *The composition of two operation homomorphisms is an operation homomorphism.*

$$\forall A : \operatorname{binop} \mathsf{X}; B : \operatorname{binop} \mathsf{Y}; C : \operatorname{binop} \mathsf{Z} \bullet \\ \quad \forall f : \hom(A, B); g : \hom(B, C) \bullet \\ \quad\quad g \circ f \in \hom(A, C)$$

## 3. Semigroups

3.1. *OperationIsAssociative*. A binary operation is said to be *associative* if the result of applying it to three elements is independent of the order in which it is applied pairwise.

Let *OperationIsAssociative* denote this situation.

*OperationIsAssociative*[t]

$$A : \operatorname{binop} \mathsf{t}$$

---

$$\forall x, y, z : \mathsf{t} \bullet \\ \quad A(A(x, y), z) = A(x, A(y, z))$$

A set of elements t with an associative binary operation is called a *semigroup*. Let *semigroup*[t] denote the set of all semigroups with elements t.

$$\mathit{semigroup}[\mathsf{t}] == \{ \mathit{OperationIsAssociative}[\mathsf{t}] \bullet A \}$$

Let the notation semigroup t, typeset using the prefix generic command `\semigroup`, denote the set of all semigroups on the set of elements t.

$$\operatorname{semigroup} \mathsf{t} == \mathit{semigroup}[\mathsf{t}]$$

**Remark.**

$$\operatorname{semigroup} \mathsf{T} \subseteq \operatorname{binop} \mathsf{T}$$

A *semigroup homomorphism* from $A$ to $B$ is a homomorphism of the underlying binary operation.

Let *hom_semigroup*$(A, B)$ denote the set of semigroup homomorphism from $A$ to $B$.

$$\mathit{hom\_semigroup}[\mathsf{T}, \mathsf{U}] == \\ \quad (\lambda A : \operatorname{semigroup} \mathsf{T}; B : \operatorname{semigroup} \mathsf{U} \bullet \hom(A, B))$$

STOPPED HERE.

Note that as a type, semigroups are a subset of binary operations. The operation homomorphisms of a semigroup are the same as the semigroup homomorphisms.

Is it clearer to introduce explicit additional structure? e.g. a semigroup is a pair (X, op) where X is a set of elements and op is a binary operation on X.

A monoid is a triple (X, op, e) where (X, op) is a semigroup and e is an identity element.

Try to define the category of semigroups. Do I need to duplicate the definitions to allow elements being subsets of the formal generic parameters?

If A is a semigroup and B is a binary operation and f is an operation homomorphism then the image of f is a semigroup.

Let $\hom_{\mathrm{sg}}(A, B)$ denote the set of all semigroup homomorphisms from $A$ to $B$.

$[\mathsf{t}, \mathsf{u}]$

$$\hom_{\mathrm{sg}} : \mathrm{semigroup}\,\mathsf{t} \times \mathrm{semigroup}\,\mathsf{u} \longrightarrow \mathbb{P}(\mathsf{t} \nrightarrow \mathsf{u})$$

---

$$\hom_{\mathrm{sg}} = (\lambda\, A : \mathrm{semigroup}\,\mathsf{t};\ B : \mathrm{semigroup}\,\mathsf{u} \bullet \hom(A, B))$$

**Remark.** *The identity mapping is a semigroup homomorphism.*

**Remark.** *The composition of two semigroup homomorphisms is another semigroup homomorphism.*

## 4. Monoids

4.1. *IdentityElement*. Let $\mathsf{t}$ be a set, let $A$ be a binary operation over $\mathsf{t}$, and let $e$ be an element of $\mathsf{t}$. The element $e$ is said to be an *identity element* of $A$ if left and right products with it leave all elements unchanged.

Let *IdentityElement* denote this situation.

*IdentityElement*$[\mathsf{t}]$

$$A : \mathrm{binop}\,\mathsf{t}$$
$$e : \mathsf{t}$$

---

$$\textbf{let}\ (\_ * \_) == A \bullet \forall\, x : \mathsf{t} \bullet e * x = x = x * e$$

4.2. *identity_element*. Let *identity_element* denote the relation that associates a binary operation one of its identity elements.

$[\mathsf{t}]$

$$\mathit{identity\_element} : \mathrm{binop}\,\mathsf{t} \leftrightarrow \mathsf{t}$$

---

$$\mathit{identity\_element} = \{\, \mathit{IdentityElement}[\mathsf{t}] \bullet A \mapsto e \,\}$$

**Remark.** *If a binary operation has an identity element then it is unique.*

*Proof.* Let $*$ be a binary operation. Suppose $e$ and $e'$ are identity elements.

$$\begin{aligned} & e \\ &= e * e' && [e' \text{ is an identity element}] \end{aligned}$$

$= e'$ [$e$ is an identity element]

□

**Remark.** *Since identity elements are unique if they exist, the relation from binary operations to identity elements is a partial function.*

$identity\_element \in \operatorname{binop} \mathsf{T} \pfun \mathsf{T}$

4.3. **Identity Element Symbols** 0 `\zeroG`**, and** 1 `\oneG`**.** Identity elements are typically denoted by the symbols 0 or 1.

4.4. monoid `\monoid`**.** Let `t` be a set of elements. A *monoid* over `t` is a semigroup over `t` that has an identity element.

Let $\operatorname{monoid} \mathsf{t}$ denote the set of all monoids over `t`.

$\operatorname{monoid} \mathsf{t} == \{ A : \operatorname{semigroup} \mathsf{t} \mid \exists e : \mathsf{t} \bullet IdentityElement[\mathsf{t}] \}$

4.5. *MapPreservesIdentity***.** Let $A$ and $B$ be monoids and let $f$ map the elements of $A$ to the elements of $B$. The map $f$ is said to *preserve the identity element* if it maps the identity element of $A$ to the identity element of $B$.

Let *MapPreservesIdentity* denote this situation.

*MapPreservesIdentity*[t, u]

$f : \mathsf{t} \longrightarrow \mathsf{u}$
$A : \operatorname{monoid} \mathsf{t}$
$B : \operatorname{monoid} \mathsf{u}$

---

**let** $e == identity\_element\ A$;
$e' == identity\_element\ B \bullet$
$f\ e = e'$

4.6. $\hom_{\mathrm{mon}}$ `\homMonoid`**.** A *monoid homomorphism* from $A$ to $B$ is a homomorphism $f$ of the underlying semigroups that preserves identity.

Let $\hom_{\mathrm{mon}}(A, B)$ denote the set of all monoid homomorphisms from $A$ to $B$.

[t, u]

$\hom_{\mathrm{mon}} : \operatorname{monoid} \mathsf{t} \times \operatorname{monoid} \mathsf{u} \longrightarrow \mathbb{P}(\mathsf{t} \longrightarrow \mathsf{u})$

---

$\hom_{\mathrm{mon}} =$
$(\lambda A : \operatorname{monoid} \mathsf{t}; B : \operatorname{monoid} \mathsf{u} \bullet$
$\{ f : \hom_{\mathrm{sg}}(A, B) \mid$
$MapPreservesIdentity[\mathsf{t}, \mathsf{u}] \})$

**Remark.** *The identity mapping is a monoid homomorphism.*

**Remark.** *The composition of two monoid homomorphisms is another monoid homomorphism.*

## 5. Groups

5.1. *InverseOperation* **and Postfix Operator symbol** $^{-1}$ `\invG`**.** Let $\mathsf{t}$ be a set of elements and let $A$ be a monoid on $\mathsf{t}$. A function $inv \in \mathsf{t} \longrightarrow \mathsf{t}$ is said to be an *inverse operation* if it maps each element to an element whose product with it is the identity element. Typically, the expression $x^{-1}$ is used to denote the inverse of $x$.

Let *InverseOperation* denote this situation.

*InverseOperation*[t]

$A : \text{monoid}\,\mathsf{t}$

$inv : \mathsf{t} \longrightarrow \mathsf{t}$

---

**let** $(\_ * \_) == A$;
$1 == \mathit{identity\_element}\ A$;
$(\_^{-1}) == inv \bullet$
$\forall x : \mathsf{t} \bullet$
$x * x^{-1} = 1 = x^{-1} * x$

5.2. *inverse_operation***.** Let *inverse_operation* denote the relation between monoids and their inverse operations.

[t]

$\mathit{inverse\_operation} : \text{monoid}\,\mathsf{t} \leftrightarrow \mathsf{t} \longrightarrow \mathsf{t}$

---

$\mathit{inverse\_operation} =$
$\{\ \mathit{InverseOperation}[\mathsf{t}] \bullet A \mapsto inv\ \}$

**Remark.** *If a monoid has an inverse operation then it is unique.*

*Proof.* Let $x$ be any element. Suppose $x^{-1}$ and $x^{\dagger}$ are inverses of $x$.

$$
\begin{aligned}
& x^{\dagger} \\
&= x^{\dagger} * 1 && [1 \text{ is an identity element}] \\
&= x^{\dagger} * (x * x^{-1}) && [x^{-1} \text{ is an inverse}] \\
&= (x^{\dagger} * x) * x^{-1} && [\text{associativity}] \\
&= 1 * x^{-1} && [x^{\dagger} \text{ is an inverse}] \\
&= x^{-1} && [1 \text{ is an identity element}]
\end{aligned}
$$

□

**Remark.** *Since if inverse operation exist they are unique, the relation between monoids and inverse operations is a partial function.*

$\mathit{inverse\_operation} \in \text{monoid}\,\mathsf{T} \pfun \mathsf{T} \longrightarrow \mathsf{T}$

5.3. group**.** A *group* is a monoid that has an inverse operation.

Let $\mathsf{t}$ be a set of elements. Let $\text{group}\,\mathsf{t}$ denote the set of all groups over $\mathsf{t}$.

$\text{group}\,\mathsf{t} == \{\ A : \text{monoid}\,\mathsf{t} \mid \exists\, inv : \mathsf{t} \longrightarrow \mathsf{t} \bullet \mathit{InverseOperation}[\mathsf{t}]\ \}$

5.4. *MapPreservesInverse*. Let $\mathsf{t}$ and $\mathsf{u}$ be sets of elements, let $A$ and $B$ be groups over $\mathsf{t}$ and $\mathsf{u}$, and let $f$ map $\mathsf{t}$ to $\mathsf{u}$. The map $f$ is said to *preserve the inverses* if it maps the inverses of elements of $A$ to the inverses of the corresponding elements of $B$.

Let *MapPreservesInverse* denote this situation.

*MapPreservesInverse*$[\mathsf{t}, \mathsf{u}]$

$f : \mathsf{t} \longrightarrow \mathsf{u}$
$A : \mathrm{group}\, \mathsf{t}$
$B : \mathrm{group}\, \mathsf{u}$

---

**let** $(\_^{-1}) == \mathit{inverse\_operation}\ A;$
$(\_^{\dagger}) == \mathit{inverse\_operation}\ B \bullet$
$\forall\, x : \mathsf{t} \bullet$
$f(x^{-1}) = (f\ x)^{\dagger}$

5.5. $\mathrm{hom}_{\mathrm{grp}}$ `\homGroup`. Let $A$ and $B$ be groups. A *group homomorphism* from $A$ to $B$ is a monoid homomorphism from $A$ to $B$ that preserves inverses.

Let $\mathrm{hom}_{\mathrm{grp}}(A, B)$ denote the set of all group homomorphisms from $A$ to $B$.

$[\mathsf{t}, \mathsf{u}]$

$\mathrm{hom}_{\mathrm{grp}} : \mathrm{group}\, \mathsf{t} \times \mathrm{group}\, \mathsf{u} \longrightarrow \mathbb{P}(\mathsf{t} \longrightarrow \mathsf{u})$

---

$\mathrm{hom}_{\mathrm{grp}} =$
$(\lambda\, A : \mathrm{group}\, \mathsf{t};\ B : \mathrm{group}\, \mathsf{u} \bullet$
$\{ f : \mathrm{hom}_{\mathrm{mon}}(A, B) \mid$
*MapPreservesInverse*$[\mathsf{t}, \mathsf{u}]\ \})$

**Remark.** *The identity mapping is a group homomorphism.*

**Remark.** *The composition of two group homomorphisms is another group homomorphism.*

5.6. *bij*. Let $\mathsf{t}$ be a set and let *bij*$[\mathsf{t}]$ denote the set of a bijections $\mathsf{t} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \mathsf{t}$ from $\mathsf{t}$ to itself.

$[\mathsf{t}]$

$\mathit{bij} : \mathbb{P}(\mathsf{t} \longrightarrow \mathsf{t})$

---

$\mathit{bij} = \mathsf{t} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \mathsf{t}$

**Remark.** *The composition of bijections is a bijection.*

$\forall\, f, g : \mathit{bij}[\mathsf{T}] \bullet$
$f \circ g \in \mathit{bij}[\mathsf{T}]$

**Remark.** *Composition is associative.*

$\forall\, f, g, h : \mathit{bij}[\mathsf{T}] \bullet$
$f \circ (g \circ h) = (f \circ g) \circ h$

**Remark.** *The identity function* $\mathrm{id}\,\mathsf{T}$ *acts as a left and right identity element under composition.*

$$\forall f : bij[\mathsf{T}] \bullet \\ \quad \mathrm{id}\,\mathsf{T} \circ f = f = f \circ \mathrm{id}\,\mathsf{T}$$

**Remark.** *The inverse* $f^\sim$ *of a bijection* $f$ *is its left and right inverse under composition.*

$$\forall f : bij[\mathsf{T}] \bullet \\ \quad f \circ f^\sim = \mathrm{id}\,\mathsf{T} = f^\sim \circ f$$

5.7. *Bij*. The preceding remarks show that set $bij[\mathsf{t}]$ under the operation of composition has the structure of a group. Let $Bij[\mathsf{t}]$ denote this group.

$[\mathsf{t}]$

$$Bij : bij[\mathsf{t}] \times bij[\mathsf{t}] \longrightarrow bij[\mathsf{t}]$$

---

$$Bij = (\lambda f, g : bij[\mathsf{t}] \bullet f \circ g)$$

**Example.** *Let* $\mathsf{T}$ *be any non-empty set. The composition operation* $Bij[\mathsf{T}]$ *is a group over the set of bijections* $bij[\mathsf{T}]$ *from* $\mathsf{T}$ *to* $\mathsf{T}$.

$$\mathsf{T} \neq \varnothing \Rightarrow \\ \quad Bij[\mathsf{T}] \in \mathrm{group}\, bij[\mathsf{T}]$$

## 6. Abelian Groups

6.1. **OperationIsCommutative.** Let $\mathsf{t}$ be a set of elements. A binary operation $A$ over $\mathsf{t}$ is said to be *commutative* when the product of two elements doesn't depend on their order.

Let *OperationIsCommutative* denote this situation.

*OperationIsCommutative*$[\mathsf{t}]$

$$A : \mathrm{binop}\,\mathsf{t}$$

---

$$\textbf{let}\ (\_ * \_) == A \bullet \\ \quad \forall x, y : \mathsf{t} \bullet \\ \quad\quad x * y = y * x$$

6.2. abgroup `\abgroup`. An *Abelian group* is a group in which the binary operation is commutative. Let $\mathsf{t}$ be a set of elements.

Let $\mathrm{abgroup}\,\mathsf{t}$ denote the set of all Abelian groups over $\mathsf{t}$.

$$\mathrm{abgroup}\,\mathsf{t} == \{\, A : \mathrm{group}\,\mathsf{t} \mid \textit{OperationIsCommutative}[\mathsf{t}] \,\}$$

6.3. + `\addG`, 0 `\zeroG`, **and** - `\negG`. Often in an Abelian group the binary operation is denoted as addition $x + y$, the identity element as a zero 0, and the inverse operation as negation - $x$.

**Example.** *Addition over the integers is an Abelian group.*

$$(\_ + \_) \in \mathrm{abgroup}\,\mathbb{Z}$$

*Email address*, Arthur Ryman: arthur.ryman@gmail.com