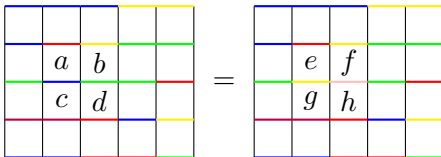


## Two-dimensional algebra



Alistair Savage  
University of Ottawa

Slides available online: [alistairsavage.ca/talks](http://alistairsavage.ca/talks)

# Outline

**Goal:** Introduce a two-dimensional approach to algebra by playing with blocks.

## Overview:

- 1 Traditional (one-dimensional) algebra
- 2 Two-dimensional algebra
- 3 Being precise: monoidal categories
- 4 Going a bit further

# Monoids

## Monoid

A **monoid** is a set  $M$  with a binary operation

$$M \times M \rightarrow M, \quad (a, b) \mapsto ab,$$

that

- is associative, so  $(ab)c = a(bc)$  for all  $a, b, c \in M$ ;
- has an identity element  $1$ , so  $1a = a = a1$  for all  $a \in M$ .

Associativity means we can omit parentheses. E.g. we can write:

$$ab^2ca1ba^3$$

Think of lining up boxes:

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $a$ | $b$ | $b$ | $c$ | $a$ | $1$ | $b$ | $a$ | $a$ | $a$ |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|

# Games with blocks

We have **relations**; some sequences of boxes are equal.

Some relations are forced by the monoid axioms:

$$\boxed{1} \boxed{a} = \boxed{a} = \boxed{a} \boxed{1}$$

This relation is **local**; we can use it inside larger sequences of blocks:

$$\boxed{a} \boxed{b} \boxed{b} \boxed{c} \boxed{a} \boxed{1} \boxed{b} \boxed{a} \boxed{a} \boxed{a} = \boxed{a} \boxed{b} \boxed{b} \boxed{c} \boxed{a} \boxed{b} \boxed{a} \boxed{a} \boxed{a}$$

**Particular** monoids may have other relations. E.g. if a monoid is commutative, we have the local relation:

$$\boxed{a} \boxed{b} = \boxed{b} \boxed{a} \quad \text{for all } a, b.$$

## Note

The axiom of **associativity** is built into our framework—we just omit parentheses.

# More games with blocks

## Presentations

We can define a monoid by giving a **presentation**:

- a set  $S$  of **generators** (the blocks we can use),
- a set  $R$  of **(local) relations**.

## Example

Consider the monoid with

- one generator  $a$ ,
- one relation  $a^2 = 1$ .

With blocks, we have  $\boxed{a} \boxed{a} = \boxed{1}$ .

This monoid is the **cyclic group with two elements**.

**Recall:** A group is a monoid where every element  $a$  has an inverse  $a^{-1}$ , with  $aa^{-1} = 1 = a^{-1}a$ . This is just a type of relation.

# More games with blocks

## Example

Consider the monoid with

- one generator  $a$ ,
- one relation  $a^n = 1$ .

With blocks, we have:

$$\underbrace{\boxed{a} \boxed{a} \cdots \boxed{a}}_{n \text{ blocks}} = \boxed{1}$$

This monoid is the **cyclic group with  $n$  elements**.

# More games with blocks

## Example

Fix  $n \geq 3$  and consider the monoid with two generators  $r, s$  and relations:

$$\underbrace{\boxed{r} \boxed{r} \cdots \boxed{r}}_{n \text{ blocks}} = \boxed{1}, \quad \boxed{s} \boxed{s} = \boxed{1}, \quad \boxed{r} \boxed{s} \boxed{r} \boxed{s} = \boxed{1}$$

This is the **dihedral group** (group of symmetries of a regular  $n$ -gon).

# More games with blocks

## Example

Fix  $n \geq 1$  and consider the monoid with generators

$$s_1, \dots, s_{n-1},$$

and relations

$$\boxed{s_i} \boxed{s_i} = \boxed{1}, \quad 1 \leq i \leq n-1,$$

$$\boxed{s_i} \boxed{s_{i+1}} \boxed{s_i} = \boxed{s_{i+1}} \boxed{s_i} \boxed{s_{i+1}}, \quad 1 \leq i \leq n-2,$$

$$\boxed{s_i} \boxed{s_j} = \boxed{s_j} \boxed{s_i}, \quad 1 \leq i, j \leq n-1, |i-j| > 1.$$

This is the **symmetric group**  $S_n$ .

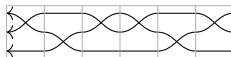


# From blocks to strings

In the symmetric group  $S_3$ , let's replace our blocks by **string diagrams**:

$$\boxed{s_1} = \begin{array}{|c|} \hline \begin{array}{c} \leftarrow \quad \leftarrow \\ \diagup \quad \diagdown \\ \leftarrow \quad \leftarrow \end{array} \\ \hline \end{array}, \quad \boxed{s_2} = \begin{array}{|c|} \hline \begin{array}{c} \leftarrow \quad \leftarrow \\ \diagdown \quad \diagup \\ \leftarrow \quad \leftarrow \end{array} \\ \hline \end{array}, \quad \boxed{1} = \begin{array}{|c|} \hline \leftarrow \\ \leftarrow \\ \leftarrow \\ \hline \end{array}$$

Lining up blocks becomes concatenation of string diagrams:



As for relations:

$$\boxed{s_1} \boxed{s_1} = \boxed{1} \quad \text{becomes} \quad \begin{array}{|c|} \hline \begin{array}{c} \leftarrow \quad \leftarrow \\ \diagup \quad \diagdown \\ \leftarrow \quad \leftarrow \end{array} \\ \hline \end{array} = \begin{array}{|c|} \hline \leftarrow \\ \leftarrow \\ \leftarrow \\ \hline \end{array}$$

$$\boxed{s_2} \boxed{s_2} = \boxed{1} \quad \text{becomes} \quad \begin{array}{|c|} \hline \begin{array}{c} \leftarrow \quad \leftarrow \\ \diagdown \quad \diagup \\ \leftarrow \quad \leftarrow \end{array} \\ \hline \end{array} = \begin{array}{|c|} \hline \leftarrow \\ \leftarrow \\ \leftarrow \\ \hline \end{array}$$

$$\boxed{s_1} \boxed{s_2} \boxed{s_1} = \boxed{s_2} \boxed{s_1} \boxed{s_2} \quad \text{becomes} \quad \begin{array}{|c|} \hline \begin{array}{c} \leftarrow \quad \leftarrow \\ \diagup \quad \diagdown \\ \leftarrow \quad \leftarrow \\ \diagdown \quad \diagup \\ \leftarrow \quad \leftarrow \end{array} \\ \hline \end{array} = \begin{array}{|c|} \hline \begin{array}{c} \leftarrow \quad \leftarrow \\ \diagdown \quad \diagup \\ \leftarrow \quad \leftarrow \\ \diagup \quad \diagdown \\ \leftarrow \quad \leftarrow \end{array} \\ \hline \end{array}$$

Exercise:

$$\begin{array}{|c|} \hline \begin{array}{c} \leftarrow \quad \leftarrow \\ \diagup \quad \diagdown \\ \leftarrow \quad \leftarrow \\ \diagdown \quad \diagup \\ \leftarrow \quad \leftarrow \end{array} \\ \hline \end{array} = \begin{array}{|c|} \hline \leftarrow \\ \leftarrow \\ \leftarrow \\ \hline \end{array}$$

# Set maps

By definition, the symmetric group  $S_n$  consists of **bijections**

$$\{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}.$$

The monoid operation is composition. Every bijection is invertible, so  $S_n$  is a group.

We could also consider the monoid of **all set maps**

$$\{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}.$$

This is no longer a group, but it is still a monoid. We can still use blocks:

$$\boxed{f \mid g} = \boxed{f \circ g}$$

**Exercise:** Come up with a natural way of drawing string diagrams for this monoid.

## Composition problems

What if we want to consider maps between **different** sets?

For  $n \geq 0$ , let

$$X_n = \{1, 2, \dots, n\}.$$

By convention,  $X_0 = \emptyset$ .

Let

$$\mathcal{C}(m, n) = \{f: X_m \rightarrow X_n\} \quad \text{and} \quad \mathcal{C} = \bigcup_{m, n=0}^{\infty} \mathcal{C}(m, n).$$

**Question:** Does the operation of composition make this a monoid?

**NO!** Composition is not always defined. E.g. if

$$f: X_2 \rightarrow X_3 \quad \text{and} \quad g: X_5 \rightarrow X_4$$

then  $f \circ g$  is not defined.

## Colourful boxes

So  $\mathcal{C}$  is **not** a monoid. So our boxes description breaks down.

How can we fix it? Colour the sides of boxes!

$$\boxed{a} \quad \boxed{b} \quad \boxed{c} \quad \boxed{d} \quad \boxed{e} \quad \boxed{f}$$

Now we can only compose boxes where the sides match:

$$\boxed{a \mid d \mid b \mid b \mid c \mid a \mid d \mid b \mid e \mid f}$$

The identity element 1 in a monoid is replaced by an identity for each colour:

$$\boxed{1} \quad \boxed{1} \quad \boxed{1} \quad \boxed{1}$$

These satisfy

$$\boxed{1 \mid a} = \boxed{a} = \boxed{a \mid 1}, \quad \boxed{1 \mid d} = \boxed{d} = \boxed{d \mid 1}, \quad \text{etc.}$$

# Back to sets

Recall

$$X_n = \{1, 2, \dots, n\},$$

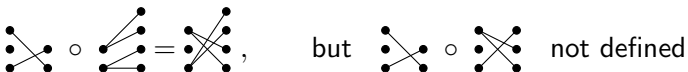
$$\mathcal{C}(m, n) = \{f: X_m \rightarrow X_n\} \quad \text{and} \quad \mathcal{C} = \bigcup_{m, n=0}^{\infty} \mathcal{C}(m, n).$$

Now we depict  $f: X_m \rightarrow X_n$  as a box with labelled/coloured sides

$$n \boxed{f} m$$

We can only compose boxes (i.e. maps) when the labels match.

If we use string diagrams for set maps ( $\circ$  = composition):



$$\begin{array}{c} \bullet \\ \bullet \end{array} \begin{array}{c} \diagup \\ \diagdown \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \circ \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \diagup \\ \diagdown \\ \diagup \\ \diagdown \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \end{array} = \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \diagup \\ \diagdown \\ \diagup \\ \diagdown \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \end{array}, \quad \text{but} \quad \begin{array}{c} \bullet \\ \bullet \end{array} \begin{array}{c} \diagup \\ \diagdown \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \circ \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \diagup \\ \diagdown \\ \diagup \\ \diagdown \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \end{array} \text{ not defined}$$

# Making things precise

## Category

A (small) **category**  $\mathcal{C}$  consists of:

- a set  $\text{Ob } \mathcal{C}$  of **objects** (“colours”),
- for all  $X, Y \in \text{Ob } \mathcal{C}$ , a set of **morphisms**  $\mathcal{C}(X, Y)$  (“boxes with edges coloured  $X$  and  $Y$ ”),

together with **composition maps**

$$\circ: \mathcal{C}(Y, Z) \times \mathcal{C}(X, Y) \rightarrow \mathcal{C}(X, Z)$$

such that

- composition is **associative** (don't worry about parentheses),
- each  $X \in \text{Ob } \mathcal{C}$  has a identity  $1_X$  such that

$$1_Y \circ f = f = f \circ 1_X \quad \text{for all } f \in \mathcal{C}(X, Y).$$

# Sets

Recall our example

$$\mathcal{C}(m, n) = \{f: X_m \rightarrow X_n\} \quad \text{and} \quad \mathcal{C} = \bigcup_{m, n=0}^{\infty} \mathcal{C}(m, n).$$

This is a category with

- **Objects:**  $X_n, n \geq 0$
- For  $X_n, X_m \in \text{Ob } \mathcal{C}$ , the set of **morphisms** from  $X_m$  to  $X_n$  is  $\mathcal{C}(m, n)$ .

Composition is usual composition of set maps.

- Composition is associative.
- The identity  $1_{X_n}: X_n \rightarrow X_n$  is the identity map.

# Categories

## Example (Sets)

- **Objects:** sets
- **Morphisms:** set maps

## Example (Vector spaces)

- **Objects:** real vector spaces
- **Morphisms:** linear maps

## Example (Groups)

- **Objects:** groups
- **Morphisms:** group homomorphisms



# Categories

## Example (Rings)

- **Objects:** rings
- **Morphisms:** ring homomorphisms

## Example (Topological spaces)

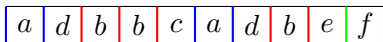
- **Objects:** topological spaces
- **Morphisms:** continuous maps

## Other examples

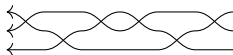
- modules over a fixed ring
- smooth manifolds
- algebraic varieties
- ...

# Stuck in one dimension

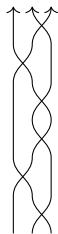
So far, everything is **one-dimensional**. We compose blocks in a line:







Or string diagrams:



From now on, we'll rotate these pictures clockwise 90°:



Relations:  =  ,  =  , etc.

But we're still stuck in one dimension—only have vertical composition ○.

## Moving to two dimensions

Let's introduce a new horizontal associative composition  $\otimes$ !

$$\begin{array}{|c|} \hline e \\ \hline b \\ \hline d \\ \hline a \\ \hline \end{array} \otimes \begin{array}{|c|} \hline f \\ \hline e \\ \hline d \\ \hline c \\ \hline \end{array} = \begin{array}{|c|c|} \hline e & f \\ \hline b & e \\ \hline d & d \\ \hline a & c \\ \hline \end{array} \quad (1)$$

How should we interpret the right-hand side? We have objects

blue, red, green, yellow, etc.

However, (1) seems to be a morphism

blue-red  $\rightarrow$  green-yellow.

So we need a way of “combining colours”.

Precisely, we want an associative map

$$\otimes: \text{Ob } \mathcal{C} \times \text{Ob } \mathcal{C} \rightarrow \text{Ob } \mathcal{C}.$$

## Moving to two dimensions

We want the associative map

$$\otimes: \text{Ob } \mathcal{C} \times \text{Ob } \mathcal{C} \rightarrow \text{Ob } \mathcal{C}$$

to have an identity. Precisely, we want an **identity object**  $\mathbb{1}$  such that

$$\mathbb{1} \otimes X = X = X \otimes \mathbb{1} \quad \text{for all } X \in \text{Ob } \mathcal{C}.$$

Let's denote this identity object by a dashed edge, e.g.

$$\boxed{a}: \mathbb{1} \rightarrow \text{red}$$

The identity object also has an identity morphism  $1_{\mathbb{1}}: \mathbb{1} \rightarrow \mathbb{1}$ :

$$\begin{array}{c} \boxed{a} \\ \hline \boxed{1_{\mathbb{1}}} \end{array} = \boxed{a}, \quad \begin{array}{c} \boxed{1_{\mathbb{1}}} \\ \hline \boxed{b} \end{array} = \boxed{b}$$

## Moving to two dimensions

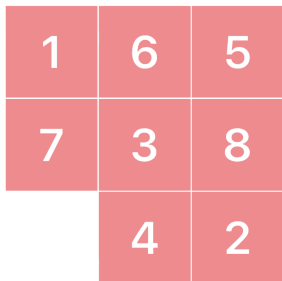
We want  $1_{\perp}$  to also be an identity for horizontal composition:

$$\boxed{1_{\perp}} \boxed{a} = \boxed{a} = \boxed{a} \boxed{1_{\perp}}$$

For technical reasons, we also want the **interchange relation**:

$$\begin{array}{|c|c|} \hline a & 1 \\ \hline 1 & b \\ \hline \end{array} = \begin{array}{|c|c|} \hline a & b \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & b \\ \hline a & 1 \\ \hline \end{array}$$

You can think of  $1_{\perp}$  a bit like an empty slot in a slide puzzle:



## 2D local relations

We can now impose **two-dimensional relations**:

$$\begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array} = \begin{array}{|c|c|} \hline e & f \\ \hline g & h \\ \hline \end{array}, \quad \begin{array}{|c|c|} \hline a & \bar{b} \\ \hline c & d \\ \hline \end{array} = \boxed{1}, \quad \text{etc.}$$

We can use these relations anywhere in a wall of blocks:

$$\begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & a & b & & \\ \hline & c & d & & \\ \hline & & & & \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & e & f & & \\ \hline & g & h & & \\ \hline & & & & \\ \hline \end{array}$$

# Making things precise

## Strict monoidal category

A **strict monoidal category** is a category  $\mathcal{C}$  equipped with

- a bifunctor (the **tensor product**)  $\otimes: \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ , and
- a **unit object**  $\mathbb{1}$ ,

such that, for all objects  $A, B, C$ ,

- $(A \otimes B) \otimes C = A \otimes (B \otimes C)$  for all objects  $A, B, C$ ,
- $\mathbb{1} \otimes A = A = A \otimes \mathbb{1}$  for all objects  $A$ ,

and, for all morphisms  $a, b, c$ ,

- $(a \otimes b) \otimes c = a \otimes (b \otimes c)$ ,
- $1_{\mathbb{1}} \otimes a = a = 1_{\mathbb{1}} \otimes a$ .

○: **vertical composition**

⊗: **horizontal composition**

# String diagrams

We will denote a morphism  $f: A \rightarrow B$  by:



The **identity map**  $1_A: A \rightarrow A$  is a string with no label:

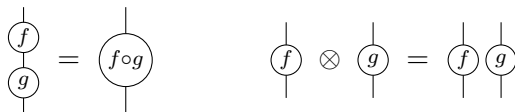


We sometimes omit the object labels when they are clear or unimportant.

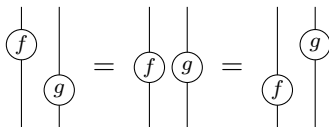


# String diagrams

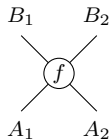
Composition is **vertical stacking** and tensor product is **horizontal juxtaposition**:



The **interchange law** then becomes:



A morphism  $f: A_1 \otimes A_2 \rightarrow B_1 \otimes B_2$  can be depicted:



## Presentations of strict monoidal categories

One can give **presentations** of strict monoidal categories, just as for monoids, groups, etc.

**Objects:** If the objects are generated by some collection  $A_i, i \in I$ , then we have all possible tensor products of these objects:

$$1, \quad A_i, \quad A_i \otimes A_j \otimes A_k \otimes A_\ell, \quad \text{etc.}$$

**Morphisms:** If the morphisms are generated by some collection  $f_j, j \in J$ , then we have all possible compositions and tensor products of these morphisms (whenever these make sense):

$$1_{A_i}, \quad f_j \otimes (f_i f_k) \otimes (f_\ell), \quad \text{etc.}$$

We then often impose some **relations** on these morphism spaces.

**String diagrams:** We can build complex diagrams out of our simple generating diagrams.

# Recall the “one-dimensional” symmetric group $S_n$

We needed  $n - 1$  generators  $s_i = \uparrow \cdots \uparrow \begin{array}{c} \nearrow \searrow \\ \searrow \nearrow \\ i+1 \quad i \end{array} \uparrow \cdots \uparrow$ ,

the  $n - 1$  quadratic relations

$$\uparrow \cdots \uparrow \begin{array}{c} \nearrow \searrow \\ \searrow \nearrow \\ i+1 \quad i \end{array} \uparrow \cdots \uparrow = \uparrow \cdots \uparrow,$$

the  $n - 2$  braid relations

$$\uparrow \cdots \uparrow \begin{array}{c} \nearrow \searrow \\ \searrow \nearrow \\ i+1 \quad i \end{array} \uparrow \cdots \uparrow = \uparrow \cdots \uparrow \begin{array}{c} \searrow \nearrow \\ \nearrow \searrow \\ i+1 \quad i \end{array} \uparrow \cdots \uparrow,$$

and the (order  $n^2$ ) distant braid relations

$$\begin{array}{c} \nearrow \searrow \\ \searrow \nearrow \\ \uparrow \quad \uparrow \end{array} \uparrow \cdots \uparrow \begin{array}{c} \searrow \nearrow \\ \nearrow \searrow \\ \uparrow \quad \uparrow \end{array} = \begin{array}{c} \searrow \nearrow \\ \nearrow \searrow \\ \uparrow \quad \uparrow \end{array} \uparrow \cdots \uparrow \begin{array}{c} \nearrow \searrow \\ \searrow \nearrow \\ \uparrow \quad \uparrow \end{array}.$$

# Recall the “one-dimensional” symmetric group $S_n$

Too many generators!!

Many of the relations are just “shifts” of each other, e.g.

$$\begin{array}{c} \uparrow \dots \uparrow \begin{array}{c} \nearrow \\ \searrow \\ \nwarrow \\ \nearrow \end{array} \uparrow \dots \uparrow = \uparrow \dots \uparrow \\ i+1 \quad i \end{array}$$

If we can compose horizontally, we just need **one** generator



and **one** relation

$$\begin{array}{c} \nearrow \\ \searrow \\ \nwarrow \\ \nearrow \end{array} = \begin{array}{c} \uparrow \\ \uparrow \end{array},$$

and similarly for the other relations.

## Monoidally generated symmetric groups

Define a strict monoidal category  $\mathcal{S}$  with one generating object  $X$  and denote

$$1_X = \uparrow$$

We have one generating morphism

$$\begin{array}{c} \nearrow \\ \times \\ \searrow \end{array} : X \otimes X \rightarrow X \otimes X.$$

We impose the relations:

$$\begin{array}{c} \nearrow \\ \times \\ \searrow \end{array} \begin{array}{c} \nearrow \\ \times \\ \searrow \end{array} = \begin{array}{c} \uparrow \\ \uparrow \end{array}, \quad \begin{array}{c} \nearrow \\ \times \\ \searrow \end{array} \begin{array}{c} \nearrow \\ \times \\ \searrow \end{array} = \begin{array}{c} \nearrow \\ \times \\ \searrow \end{array} \begin{array}{c} \nearrow \\ \times \\ \searrow \end{array}.$$

Then

$$\mathcal{S}(X^{\otimes n}, X^{\otimes n}) = S_n$$

is the **symmetric group** on  $n$  letters.

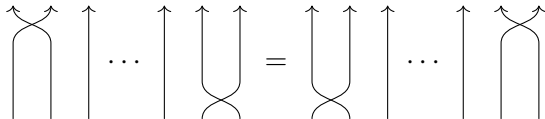
# Monoidally generated symmetric groups

This monoidal presentation of  $S_n$  is very efficient! We only needed

- one generating morphism, and
- two relations,

to get **all** the symmetric groups.

Note that the **distant braid relation** follows for **free** from the interchange law:



# Monoidal categories in the wild

## Example (Sets I)

- **Objects:** finite sets
- **Morphisms:** set maps
- **Vertical composition:** composition of set maps
- **Horizontal composition:** cartesian product
- **Unit object  $\mathbb{1}$ :** one element set  $\{\star\}$

## Example (Sets II)

- **Objects:** finite sets
- **Morphisms:** set maps
- **Vertical composition:** composition of set maps
- **Horizontal composition:** disjoint union
- **Unit object  $\mathbb{1}$ :** empty set  $\emptyset$

# Monoidal categories in the wild

## Example (Vector spaces)

- **Objects:**  $\mathbb{R}$ -vector spaces
- **Morphisms:** linear transformations
- **Vertical composition:** composition of linear transformations
- **Horizontal composition:** tensor product  $\otimes$
- **Unit object  $\mathbb{1}$ :**  $\mathbb{R}$

## Example (Abelian groups)

- **Objects:** abelian groups
- **Morphisms:** group homomorphisms
- **Vertical composition:** composition of group homomorphisms
- **Horizontal composition:** tensor product  $\otimes_{\mathbb{Z}}$
- **Unit object  $\mathbb{1}$ :**  $\mathbb{Z}$



## Example (Rings)

- **Objects:** rings
- **Morphisms:** ring homomorphisms
- **Vertical composition:** composition of ring homomorphisms
- **Horizontal composition:** tensor product  $\otimes_{\mathbb{Z}}$
- **Unit object  $\mathbb{1}$ :**  $\mathbb{Z}$

## Technicality

The above examples are not really **strict** monoidal categories. E.g. for sets  $X, Y, Z$ , we don't have  $(X \times Y) \times Z = X \times (Y \times Z)$ .

However, we have a **natural isomorphism**

$$(X \times Y) \times Z \cong X \times (Y \times Z).$$

So the above are **strict monoidal categories**. Every monoidal category is **equivalent** to a strict one.

# More colours please!

Recall our horizontal composition

$$\begin{array}{|c|} \hline e \\ \hline b \\ \hline d \\ \hline a \\ \hline \end{array} \otimes \begin{array}{|c|} \hline f \\ \hline e \\ \hline d \\ \hline c \\ \hline \end{array} = \begin{array}{|c|c|} \hline e & f \\ \hline b & e \\ \hline d & d \\ \hline a & c \\ \hline \end{array}$$

Why are only the horizontal edges coloured?

Indeed, we can colour the vertical edges as well:

$$\begin{array}{|c|} \hline a \\ \hline \end{array}$$

Now we can only compose horizontally when colours match (just like for vertical composition)!

## 2-categories

### 2-category

- objects (vertical edges of boxes),
- 1-morphisms (horizontal edges of boxes),
- 2-morphisms (boxes).

### Example

#### Bimodules over rings

- **Objects:** rings
- **1-Morphisms:** bimodules
- **2-morphisms:** bimodule homomorphisms.

A monoidal category is a 2-category with one object (one vertical edge colour):

- 1-morphisms of 2-category  $\rightsquigarrow$  objects of monoidal category
- 2-morphisms of 2-category  $\rightsquigarrow$  morphisms of monoidal category