Coherence for tricategories via weak vertical composition

Eugenia Cheng

School of the Art Institute of Chicago

Plan

Aim: show that tricategories with just weak vertical composition are “weak enough”

Plan

Aim: show that tricategories with just weak vertical composition are "weak enough"

. . . and thus shed light on the source of weakness in higher categories.

Plan

Aim: show that tricategories with just weak vertical composition are "weak enough"

. . . and thus shed light on the source of weakness in higher categories.

1. Overview: degeneracy and braidings

Plan

Aim: show that tricategories with just weak vertical composition are "weak enough"

…and thus shed light on the source of weakness in higher categories.

1. Overview: degeneracy and braidings
2. Warm-up: strictification via cliques

Plan

Aim: show that tricategories with just weak vertical composition are "weak enough"

...and thus shed light on the source of weakness in higher categories.

1. Overview: degeneracy and braidings
2. Warm-up: strictification via cliques
3. Construction

Plan

Aim: show that tricategories with just weak vertical composition are "weak enough"

...and thus shed light on the source of weakness in higher categories.

1. Overview: degeneracy and braidings
2. Warm-up: strictification via cliques
3. Construction
4. Equivalence.

Test principle for weak n-categories

Doubly degenerate 3-categories should "be" braided monoidal categories.

Test principle for weak n-categories

Doubly degenerate 3-categories should "be" braided monoidal categories.

dd 3-category

$$\text{degenerate}\begin{cases}\text{0-cells}\\ \text{1-cells}\end{cases}$$

2-cells

3-cells

Test principle for weak *n*-categories

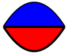
Doubly degenerate 3-categories should “be” braided monoidal categories.

	dd 3-category		**braided monoidal category**
degenerate	0-cells		
	1-cells		
	2-cells	$\longrightarrow$	objects
	3-cells	$\longrightarrow$	morphisms

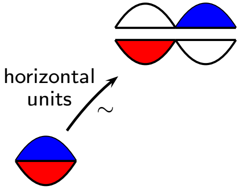
Test principle for weak *n*-categories

Doubly degenerate 3-categories should "be" braided monoidal categories.

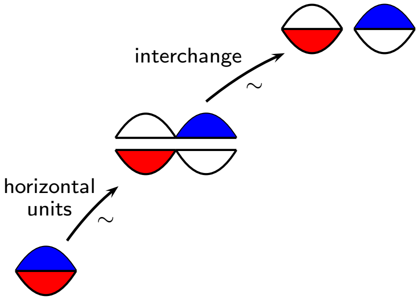
dd 3-category		**braided monoidal category**
degenerate: 0-cells, 1-cells		
2-cells	$\longrightarrow$	objects
3-cells	$\longrightarrow$	morphisms
vertical composition	$\longrightarrow$	$\otimes$
weak Eckmann–Hilton	$\longrightarrow$	braiding



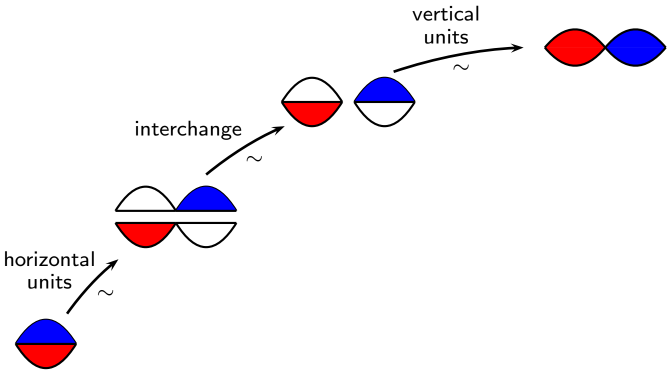
1. Overview: braiding from weak Eckmann–Hilton



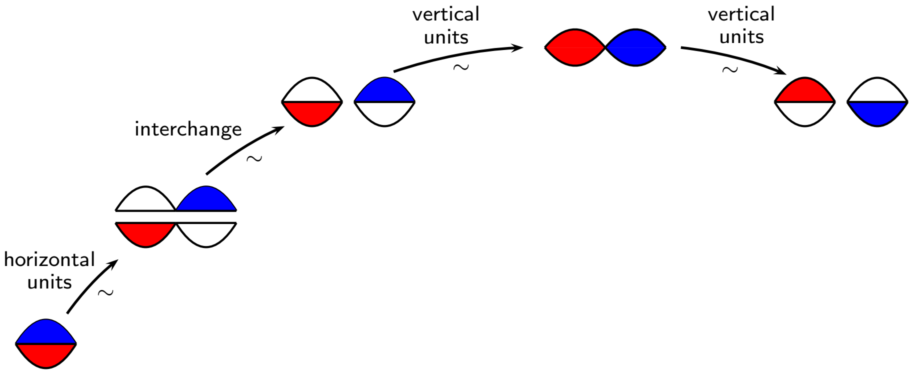
1. Overview: braiding from weak Eckmann–Hilton



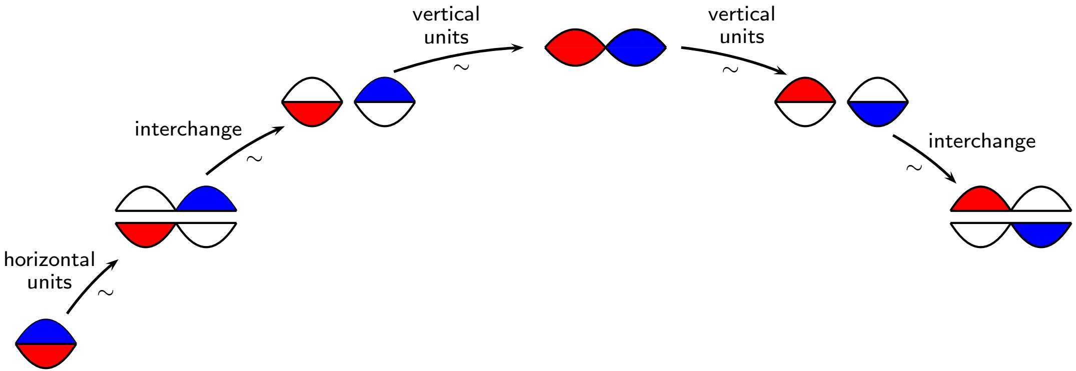
1. Overview: braiding from weak Eckmann–Hilton



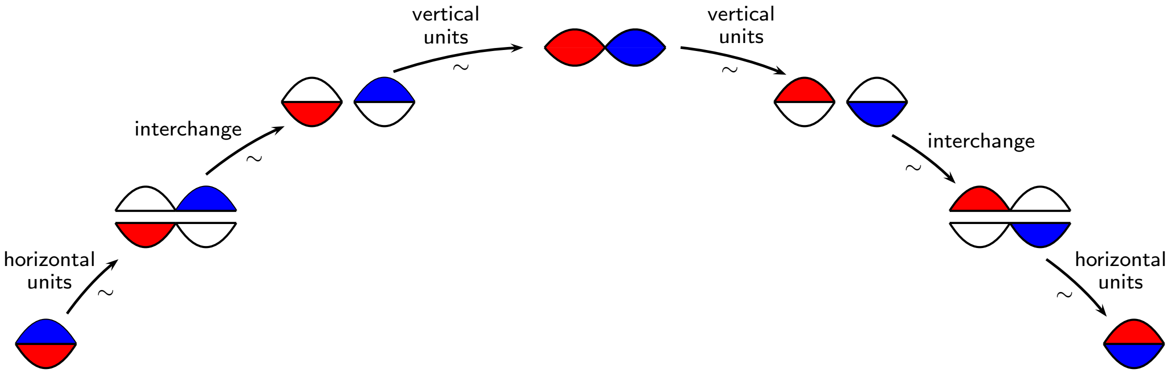
1. Overview: braiding from weak Eckmann–Hilton



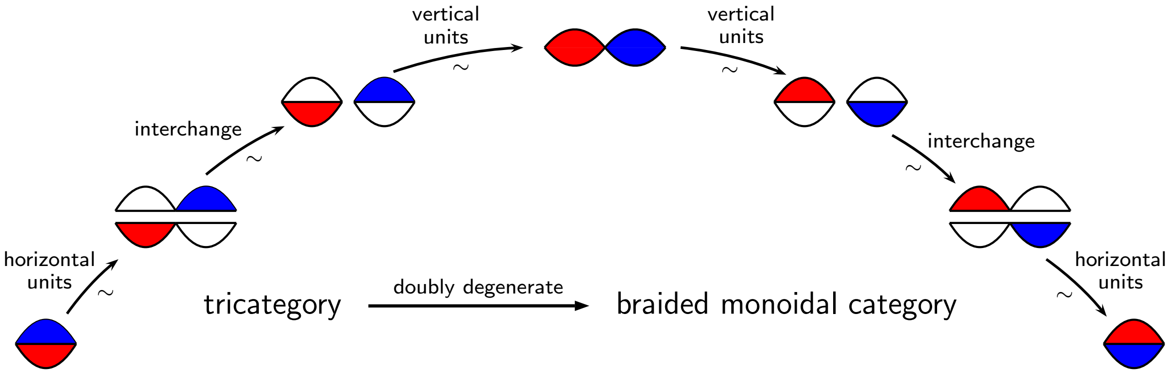
1. Overview: braiding from weak Eckmann–Hilton



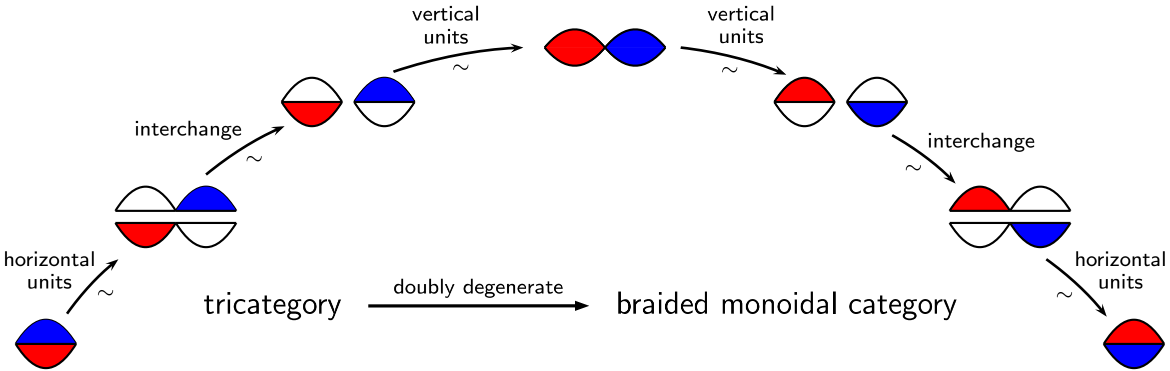
1. Overview: braiding from weak Eckmann–Hilton



1. Overview: braiding from weak Eckmann–Hilton



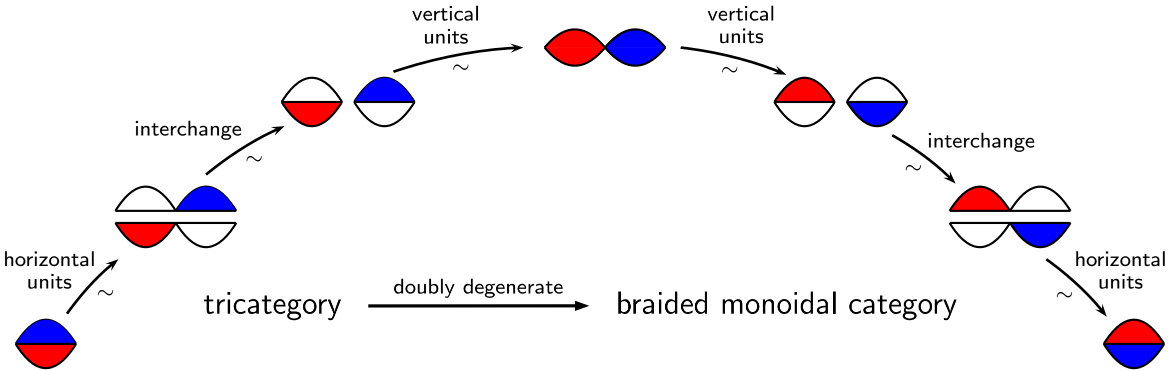
1. Overview: braiding from weak Eckmann–Hilton



tricategory $\xrightarrow{\text{doubly degenerate}}$ braided monoidal category

strict 3-category $\longrightarrow$ symmetric monoidal category

1. Overview: braiding from weak Eckmann–Hilton



tricategory $\xrightarrow{\text{doubly degenerate}}$ braided monoidal category

“semi-strict” $\dashrightarrow$ Gray area

strict 3-category $\longrightarrow$ symmetric monoidal category

1. Overview: flavours of semi-strictness

	vertical units	**horizontal units**	**interchange**
GPS	strict	strict	weak
JK	strict	weak	strict
C	weak	strict	strict

"Law of conservation of complicatedness"

1. Overview: flavours of semi-strictness

	vertical units	horizontal units	interchange	subtlety
GPS	strict	strict	weak	
JK	strict	weak	strict	not doubly degenerate
C	weak	strict	strict	need weak vertical associativity

"Law of conservation of complicatedness"

1. Overview: flavours of semi-strictness

	vertical units	**horizontal units**	**interchange**	**subtlety**
GPS	strict	strict	weak	
JK	strict	weak	strict	not doubly degenerate
C	weak	strict	strict	need weak vertical associativity

type of $\otimes$
or type of functors

"Law of conservation of complicatedness"

1. Overview: flavours of semi-strictness

	vertical units	horizontal units	interchange	subtlety
GPS	strict	strict	weak	
JK	strict	weak	strict	not doubly degenerate
C	weak	strict	strict	need weak vertical associativity

type of enrichment

type of $\otimes$
or type of functors

"Law of conservation of complicatedness"

1. Overview: flavours of semi-strictness

	vertical units	horizontal units	interchange	subtlety
GPS	strict	strict	weak	
JK	strict	weak	strict	not doubly degenerate
C	weak	strict	strict	need weak vertical associativity

what we enrich in

type of enrichment

type of $\otimes$
or type of functors

"Law of conservation of complicatedness"

1. Overview: flavours of semi-strictness

	vertical units	**horizontal units**	**interchange**	**subtlety**
GPS	strict	strict	weak	
JK	strict	weak	strict	not doubly degenerate
C	weak	strict	strict	need weak vertical associativity

what we enrich in

type of enrichment

type of $\otimes$
or type of functors

We enrich in $(\mathbf{Bicat}_s, \times)$:

- bicategories
- strict functors
- ordinary products
- strict enrichment

We write $\mathbf{Bicat}_s$-$\mathbf{Cat}$.

"Law of conservation of complicatedness"

$$\begin{tikzcd} \mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \arrow[r, bend left, "U"] & \mathrm{w}\mathbf{BrMonCat} \end{tikzcd}$$

$$\begin{tikzcd} \mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \arrow[r, bend left, "U"] & \mathrm{w}\mathbf{BrMonCat} \arrow[l, bend left, dashed, "\Sigma"] \end{tikzcd}$$

- eventually: biequivalence
- first: U essentially surjective on 0-cells

1. Overview: overall aim

$$\begin{tikzcd} \mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \arrow[rr, bend left, "U"] & & \mathrm{w}\mathbf{BrMonCat} \arrow[ll, bend left, dashed, "\Sigma"] \end{tikzcd}$$

- eventually: biequivalence
- first: U essentially surjective on 0-cells

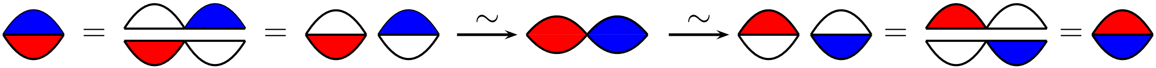
$$\begin{array}{ccc} A & & UA \\ \text{2-cells} & \longmapsto & \text{objects} \\ \text{3-cells} & \longmapsto & \text{morphisms} \\ \text{vertical } \circ & \longmapsto & \otimes \\ \text{(weak)} & & \text{(weak)} \end{array}$$

$$\begin{tikzcd} \mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \arrow[r, bend left, "U"] & \mathrm{w}\mathbf{BrMonCat} \arrow[l, bend left, dashed, "\Sigma"] \end{tikzcd}$$

- eventually: biequivalence
- first: U essentially surjective on 0-cells

$$\begin{array}{ccc} A & & UA \\ \text{2-cells} & \longmapsto & \text{objects} \\ \text{3-cells} & \longmapsto & \text{morphisms} \\ \text{vertical } \circ & \longmapsto & \otimes \\ \text{(weak)} & & \text{(weak)} \end{array}$$

Weak Eckmann-Hilton:



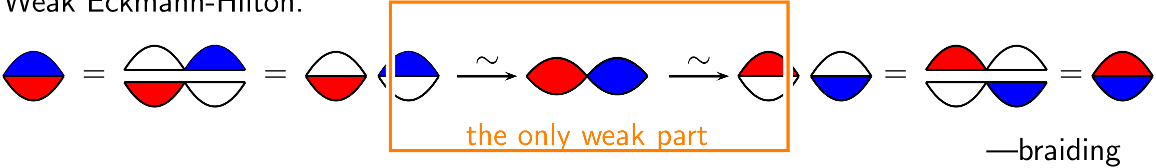
—braiding

$$\begin{tikzcd} \mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \arrow[r, bend left, "U"] & \mathrm{w}\mathbf{BrMonCat} \arrow[l, bend left, dashed, "\Sigma"] \end{tikzcd}$$

A		UA
2-cells	$\longmapsto$	objects
3-cells	$\longmapsto$	morphisms
vertical $\circ$ (weak)	$\longmapsto$	$\otimes$ (weak)

- eventually: biequivalence
- first: U essentially surjective on 0-cells

Weak Eckmann-Hilton:



1. Overview: essential surjectivity

We show $\mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

1. Overview: essential surjectivity

We show $\mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

- start with a braided monoidal category B (with strict $\otimes$ wlog),

1. Overview: essential surjectivity

We show $\mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

- start with a braided monoidal category B (with strict $\otimes$ wlog),
- construct a $\mathrm{dd}\mathbf{Bicat}_s$-category ΣB, and

We show $\mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

- start with a braided monoidal category B (with strict $\otimes$ wlog),
- construct a $\mathrm{dd}\mathbf{Bicat}_s$-category ΣB, and
- a braided monoidal equivalence $B \xrightarrow{\sim} \Sigma B$.

We show $\text{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

- start with a braided monoidal category B (with strict $\otimes$ wlog),
- construct a $\text{dd}\mathbf{Bicat}_s$-category ΣB, and
- a braided monoidal equivalence $B \xrightarrow{\sim} \Sigma B$.

How do we get horizontal *and* vertical composition from $\otimes$?

1. Overview: essential surjectivity

We show $\mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

- start with a braided monoidal category B (with strict $\otimes$ wlog),
- construct a $\mathrm{dd}\mathbf{Bicat}_s$-category ΣB, and
- a braided monoidal equivalence $B \xrightarrow{\sim} \Sigma B$.

$$\boxed{\text{How do we get horizontal } \textit{and} \text{ vertical composition from } \otimes\text{?}}$$

For tricategories: Put both as $\otimes$ and get interchange from the braiding γ

1. Overview: essential surjectivity

We show $\mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

- start with a braided monoidal category B (with strict $\otimes$ wlog),
- construct a $\mathrm{dd}\mathbf{Bicat}_s$-category ΣB, and
- a braided monoidal equivalence $B \xrightarrow{\sim} \Sigma B$.

> How do we get horizontal *and* vertical composition from $\otimes$?

For tricategories: Put both as $\otimes$ and get interchange from the braiding γ

Issues: 1. We can't have both compositions strict so they can't both be $\otimes$.

We show $\mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

- start with a braided monoidal category B (with strict $\otimes$ wlog),
- construct a $\mathrm{dd}\mathbf{Bicat}_s$-category ΣB, and
- a braided monoidal equivalence $B \xrightarrow{\sim} \Sigma B$.

How do we get horizontal *and* vertical composition from $\otimes$?

For tricategories: Put both as $\otimes$ and get interchange from the braiding γ

Issues:
1. We can't have both compositions strict so they can't both be $\otimes$.
2. We want interchange to be strict so it can't be γ.

We show $\mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \xrightarrow{U} w\mathbf{BrMonCat}$ is essentially surjective on objects:

- start with a braided monoidal category B (with strict $\otimes$ wlog),
- construct a $\mathrm{dd}\mathbf{Bicat}_s$-category ΣB, and
- a braided monoidal equivalence $B \xrightarrow{\sim} \Sigma B$.

How do we get horizontal *and* vertical composition from $\otimes$?

For tricategories: Put both as $\otimes$ and get interchange from the braiding γ

Issues:
1. We can't have both compositions strict so they can't both be $\otimes$.
2. We want interchange to be strict so it can't be γ.

Solution: Do "weakification" for the vertical direction.

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

2. Warm-up: strictification of weak monoidal categories

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

Follows from: $F1$ is monoidal equivalent to the discrete $(\mathbb{N}, +, 0)$.

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

Follows from: $F1$ is monoidal equivalent to the discrete $(\mathbb{N}, +, 0)$.

So
- $F1$ splits into connected components C_n "bracketed words of length n".

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

Follows from: $F1$ is monoidal equivalent to the discrete $(\mathbb{N}, +, 0)$.

So
- $F1$ splits into connected components C_n "bracketed words of length n".
- Each $C_n \simeq 1$ so all bracketings are uniquely isomorphic "all diagrams commute".

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

Follows from: $F1$ is monoidal equivalent to the discrete $(\mathbb{N}, +, 0)$.

So
- $F1$ splits into connected components C_n "bracketed words of length n".
- Each $C_n \simeq 1$ so all bracketings are uniquely isomorphic "all diagrams commute".

Given a monoidal category M we construct $\mathbf{st}\, M$ "strictification"

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

Follows from: $F1$ is monoidal equivalent to the discrete $(\mathbb{N}, +, 0)$.

So
- $F1$ splits into connected components C_n "bracketed words of length n".
- Each $C_n \simeq 1$ so all bracketings are uniquely isomorphic "all diagrams commute".

Given a monoidal category M we construct $\mathbf{st}\, M$ "strictification"

- objects: words in the objects of M (unbracketed)

2. Warm-up: strictification of weak monoidal categories

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

Follows from: $F1$ is monoidal equivalent to the discrete $(\mathbb{N}, +, 0)$.

So
- $F1$ splits into connected components C_n "bracketed words of length n".
- Each $C_n \simeq 1$ so all bracketings are uniquely isomorphic "all diagrams commute".

$$\boxed{\text{Given a monoidal category } M \text{ we construct } \mathbf{st}\, M \text{ ``strictification''}}$$

- objects: words in the objects of M (unbracketed)
- morphisms: evaluate words in M then take morphisms from M.

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

Follows from: $F1$ is monoidal equivalent to the discrete $(\mathbb{N}, +, 0)$.

So
- $F1$ splits into connected components C_n "bracketed words of length n".
- Each $C_n \simeq 1$ so all bracketings are uniquely isomorphic "all diagrams commute".

Given a monoidal category M we construct $\mathbf{st}\, M$ "strictification"

- objects: words in the objects of M (unbracketed)
- morphisms: evaluate words in M then take morphisms from M.

Question: How do we evaluate strict words in a weak monoidal category?

2. Warm-up: strictification of weak monoidal categories

Coherence: Every weak monoidal category is monoidal equivalent to a strict one.

Follows from: $F1$ is monoidal equivalent to the discrete $(\mathbb{N}, +, 0)$.

So
- $F1$ splits into connected components C_n "bracketed words of length n".
- Each $C_n \simeq 1$ so all bracketings are uniquely isomorphic "all diagrams commute".

Given a monoidal category M we construct $\mathbf{st}\, M$ "strictification"

- objects: words in the objects of M (unbracketed)
- morphisms: evaluate words in M then take morphisms from M.

Question: How do we evaluate strict words in a weak monoidal category?

Answer: Use cliques.

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\, M$.

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\,M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\,M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.
- You could take morphisms $a \otimes (b \otimes c) \longrightarrow y$.

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\,M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.
- You could take morphisms $a \otimes (b \otimes c) \longrightarrow y$.
- We could all take different ones by throwing in copies of I.

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\, M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.
- You could take morphisms $a \otimes (b \otimes c) \longrightarrow y$.
- We could all take different ones by throwing in copies of I.

Key: they're not *really* different.

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\, M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.
- You could take morphisms $a \otimes (b \otimes c) \longrightarrow y$.
- We could all take different ones by throwing in copies of I.

Key: they're not *really* different.

All bracketings of abc are uniquely isomorphic via coherence constraints.

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\, M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.
- You could take morphisms $a \otimes (b \otimes c) \longrightarrow y$.
- We could all take different ones by throwing in copies of I.

Key: they're not *really* different.

All bracketings of *abc* are uniquely isomorphic via coherence constraints.

We just have to agree that we've got
the same morphism if they only differ
by that unique isomorphism:

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\,M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.
- You could take morphisms $a \otimes (b \otimes c) \longrightarrow y$.
- We could all take different ones by throwing in copies of I.

Key: they're not *really* different.

All bracketings of *abc* are uniquely isomorphic via coherence constraints.

We just have to agree that we've got the same morphism if they only differ by that unique isomorphism:

$$(a \otimes b) \otimes c \xrightarrow{(f \otimes g) \otimes h} (a' \otimes b') \otimes c$$

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\, M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.
- You could take morphisms $a \otimes (b \otimes c) \longrightarrow y$.
- We could all take different ones by throwing in copies of I.

Key: they're not *really* different.

All bracketings of abc are uniquely isomorphic via coherence constraints.

We just have to agree that we've got the same morphism if they only differ by that unique isomorphism:

$$\begin{tikzcd} (a \otimes b) \otimes c \arrow[r, "(f \otimes g) \otimes h"] & (a' \otimes b') \otimes c \\ a \otimes (b \otimes c) \arrow[r, "f \otimes (g \otimes h)"'] & a' \otimes (b' \otimes c') \end{tikzcd}$$

Thought experiment

Suppose we're trying to define morphisms $abc \longrightarrow y$ in $\mathbf{st}\, M$.

- I could take morphisms $(a \otimes b) \otimes c \longrightarrow y$.
- You could take morphisms $a \otimes (b \otimes c) \longrightarrow y$.
- We could all take different ones by throwing in copies of I.

$$\boxed{\text{Key: they're not } \textit{really} \text{ different.}}$$

All bracketings of abc are uniquely isomorphic via coherence constraints.

We just have to agree that we've got the same morphism if they only differ by that unique isomorphism:

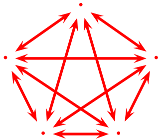
$$\begin{tikzcd}
(a \otimes b) \otimes c \arrow[r, "(f \otimes g) \otimes h"] \arrow[d, "\mathfrak{a}"', "\sim"] & (a' \otimes b') \otimes c \arrow[d, "\mathfrak{a}"', "\sim"] \\
a \otimes (b \otimes c) \arrow[r, "f \otimes (g \otimes h)"'] & a' \otimes (b' \otimes c')
\end{tikzcd}$$

Idea

A clique is essentially a collection of objects and unique isomorphisms between them.

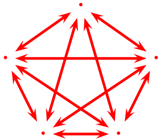
Idea

A clique is essentially a collection of objects and unique isomorphisms between them.



Idea

A clique is essentially a collection of objects and unique isomorphisms between them.

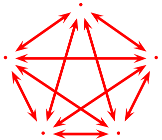


Precisely/technically

A clique in a category $\mathcal{C}$ is a functor $J \longrightarrow \mathcal{C}$ where $J \simeq 1$

Idea

A clique is essentially a collection of objects and unique isomorphisms between them.



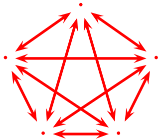
Precisely/technically

A clique in a category $\mathcal{C}$ is a functor $J \longrightarrow \mathcal{C}$ where $J \simeq 1$

— clique objects are indexed by J but can repeat.

Idea

A clique is essentially a collection of objects and unique isomorphisms between them.



Precisely/technically

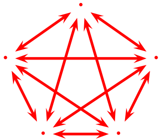
A clique in a category $\mathcal{C}$ is a functor $J \longrightarrow \mathcal{C}$ where $J \simeq 1$

— clique objects are indexed by J but can repeat.

The unique isomorphisms are called connecting isomorphisms.

Idea

A clique is essentially a collection of objects and unique isomorphisms between them.



Precisely/technically

A clique in a category $\mathcal{C}$ is a functor $J \longrightarrow \mathcal{C}$ where $J \simeq 1$

— clique objects are indexed by J but can repeat.

The unique isomorphisms are called connecting isomorphisms.

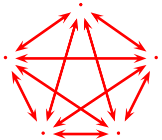
A clique map $\overline{x} \longrightarrow \overline{y}$

is a system of morphisms from each object of $\overline{x}$ to each object of $\overline{y}$

making everything commute.

Idea

A clique is essentially a collection of objects and unique isomorphisms between them.



Precisely/technically

A clique in a category $\mathcal{C}$ is a functor $J \longrightarrow \mathcal{C}$ where $J \simeq 1$

— clique objects are indexed by J but can repeat.

The unique isomorphisms are called connecting isomorphisms.

A clique map $\overline{x} \longrightarrow \overline{y}$

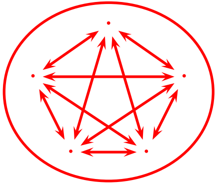
is a system of morphisms from each object of $\overline{x}$ to each object of $\overline{y}$

making everything commute.

We only have to specify one component to know what all the others are.

Idea

A clique is essentially a collection of objects and unique isomorphisms between them.

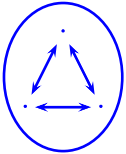


Precisely/technically

A clique in a category $\mathcal{C}$ is a functor $J \longrightarrow \mathcal{C}$ where $J \simeq 1$

— clique objects are indexed by J but can repeat.

The unique isomorphisms are called connecting isomorphisms.



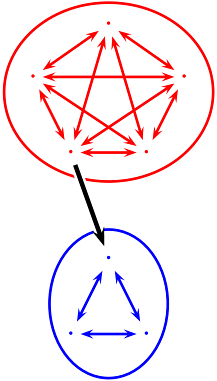
A clique map $\overline{x} \longrightarrow \overline{y}$

is a system of morphisms from each object of $\overline{x}$ to each object of $\overline{y}$ making everything commute.

We only have to specify one component to know what all the others are.

Idea

A clique is essentially a collection of objects and unique isomorphisms between them.



Precisely/technically

A clique in a category $\mathcal{C}$ is a functor $J \longrightarrow \mathcal{C}$ where $J \simeq 1$

— clique objects are indexed by J but can repeat.

The unique isomorphisms are called connecting isomorphisms.

A clique map $\overline{x} \longrightarrow \overline{y}$

is a system of morphisms from each object of $\overline{x}$ to each object of $\overline{y}$ making everything commute.

We only have to specify one component to know what all the others are.

Example

There is a clique in M of all bracketings of abc (evaluated)

Example

There is a clique in M of all bracketings of abc (evaluated)

$$(a \otimes b) \otimes c \xleftrightarrow{\sim} a \otimes (b \otimes c)$$

Example

There is a clique in M of all bracketings of abc (evaluated)

$$(a \otimes b) \otimes c \xleftrightarrow{\sim} a \otimes (b \otimes c)$$

$$(a' \otimes b') \otimes c \xleftrightarrow[\sim]{} a' \otimes (b' \otimes c')$$

2. Warm-up: strictification of weak monoidal categories

Example

There is a clique in M of all bracketings of abc (evaluated)

$$\begin{tikzcd}
(a \otimes b) \otimes c \arrow[r, leftrightarrow, "\sim"] \arrow[d, "(f \otimes g) \otimes h"'] & a \otimes (b \otimes c) \\
(a' \otimes b') \otimes c \arrow[r, leftrightarrow, "\sim"'] & a' \otimes (b' \otimes c')
\end{tikzcd}$$

Example

There is a clique in M of all bracketings of abc (evaluated)

$$
\begin{tikzcd}
(a \otimes b) \otimes c \arrow[rr, leftrightarrow, "\sim"] \arrow[d, "(f \otimes g) \otimes h"'] & & a \otimes (b \otimes c) \arrow[d, dashed, "f \otimes (g \otimes h)"] \\
(a' \otimes b') \otimes c \arrow[rr, leftrightarrow, "\sim"'] & & a' \otimes (b' \otimes c')
\end{tikzcd}
$$

$\dashrightarrow$

other components of the same clique map

Example

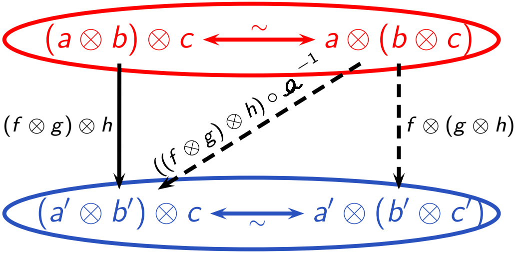
There is a clique in M of all bracketings of abc (evaluated)

$$\begin{tikzcd}
(a \otimes b) \otimes c \arrow[rr, leftrightarrow, "\sim"] \arrow[d, "(f \otimes g) \otimes h"'] \arrow[drr, dashed, "\mathfrak{a} \circ ((f \otimes g) \otimes h)"] & & a \otimes (b \otimes c) \arrow[d, dashed, "f \otimes (g \otimes h)"] \\
(a' \otimes b') \otimes c \arrow[rr, leftrightarrow, "\sim"'] & & a' \otimes (b' \otimes c')
\end{tikzcd}$$

$\dashrightarrow$ other components of the same clique map

Example

There is a clique in M of all bracketings of abc (evaluated)



other components of the same clique map

Example

There is a clique in M of all bracketings of abc (evaluated)

$$\begin{tikzcd}
(a \otimes b) \otimes c \arrow[r, leftrightarrow, "\sim"] \arrow[d, "(f \otimes g) \otimes h"'] & a \otimes (b \otimes c) \arrow[d, dashed, "f \otimes (g \otimes h)"] \arrow[dl, dashed, "((f \otimes g) \otimes h) \circ \alpha^{-1}"'] \\
(a' \otimes b') \otimes c \arrow[r, leftrightarrow, "\sim"'] & a' \otimes (b' \otimes c')
\end{tikzcd}$$

$$\dashrightarrow$$

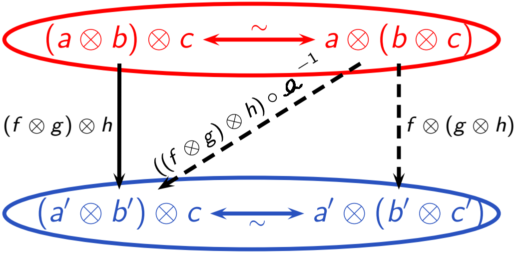
other components of the same clique map

We can compose components that don't look composable via connecting isos

$$\begin{tikzcd}
(a \otimes b) \otimes c \arrow[d] & \\
(a' \otimes b') \otimes c' & a' \otimes (b' \otimes c') \arrow[d] \\
& a'' \otimes (b'' \otimes c'')
\end{tikzcd}$$

Example

There is a clique in M of all bracketings of abc (evaluated)



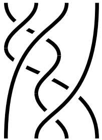
other components of the same clique map

We can compose components that don't look composable via connecting isos

$$
\begin{tikzcd}
(a \otimes b) \otimes c \arrow[r, leftrightarrow] \arrow[d] & a \otimes (b \otimes c) \\
(a' \otimes b') \otimes c' \arrow[r, leftrightarrow] & a' \otimes (b' \otimes c') \arrow[d] \\
(a'' \otimes b'') \otimes c'' \arrow[r, leftrightarrow] & a'' \otimes (b'' \otimes c'')
\end{tikzcd}
$$

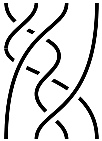
2. Warm-up: dots in boxes

Coherence for braided monoidal categories relates $F1$ to the braid category.

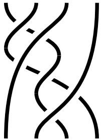


Coherence for braided monoidal categories relates $F1$ to the braid category.

Braids come from configurations of points in $\mathbb{R}^2$, and paths.



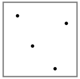
2. Warm-up: dots in boxes



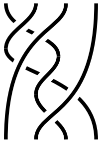
Coherence for braided monoidal categories relates $F1$ to the braid category.

Braids come from configurations of points in $\mathbb{R}^2$, and paths.

We use configurations of points in the interior of I^2

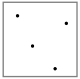


Coherence for braided monoidal categories relates $F1$ to the braid category.

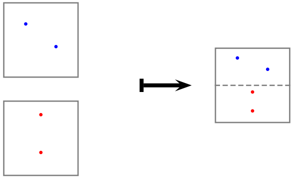


Braids come from configurations of points in $\mathbb{R}^2$, and paths.

We use configurations of points in the interior of I^2

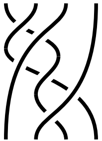


- Vertical composition

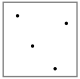


Coherence for braided monoidal categories relates $F1$ to the braid category.

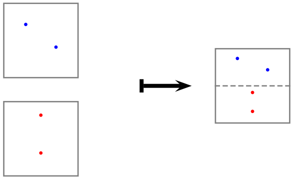
Braids come from configurations of points in $\mathbb{R}^2$, and paths.



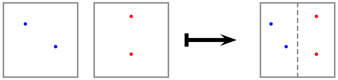
We use configurations of points in the interior of I^2



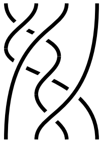
- Vertical composition



- Horizontal composition

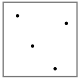


Coherence for braided monoidal categories relates $F1$ to the braid category.

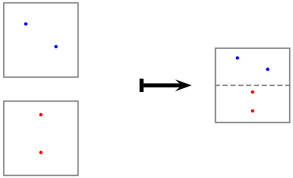


Braids come from configurations of points in $\mathbb{R}^2$, and paths.

We use configurations of points in the interior of I^2

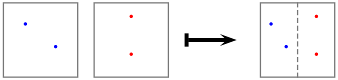


- Vertical composition

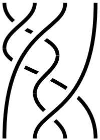


Problem: Both weak.

- Horizontal composition

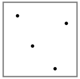


2. Warm-up: dots in boxes

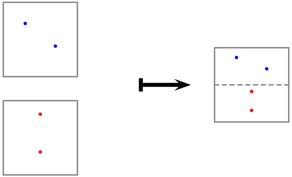


Coherence for braided monoidal categories relates $F1$ to the braid category.

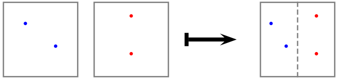
Braids come from configurations of points in $\mathbb{R}^2$, and paths.

We use configurations of points in the interior of I^2 

- Vertical composition

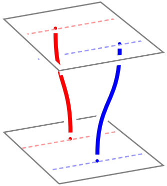


- Horizontal composition



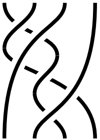
Problem: Both weak.

Solution: Take "horizontal path" classes
— paths that do not change any y coordinate



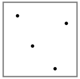
2. Warm-up: dots in boxes

Coherence for braided monoidal categories relates $F1$ to the braid category.

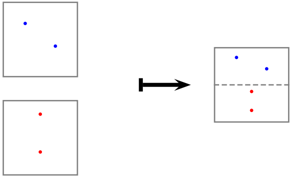


Braids come from configurations of points in $\mathbb{R}^2$, and paths.

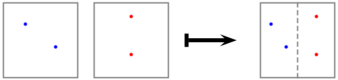
We use configurations of points in the interior of I^2



- Vertical composition

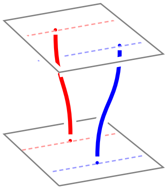


- Horizontal composition



Problem: Both weak.

Solution: Take "horizontal path" classes — paths that do not change any y coordinate



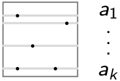
"strictification in the horizontal direction"

3. Construction of ΣB

Aim: construct a dd$\mathbf{Bicat}_s$-category ΣB from a braided monoidal category B

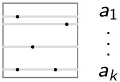
Aim: construct a dd$\mathbf{Bicat}_s$-category ΣB from a braided monoidal category B

Objects: horizontal path classes of points in I^2, labelled by objects of B



3. Construction of ΣB

Aim: construct a dd$\mathbf{Bicat}_s$-category ΣB from a braided monoidal category B

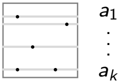


Objects: horizontal path classes of points in I^2, labelled by objects of B

Morphisms (cf strictification): start by evaluating the configuration as a word

3. Construction of ΣB

Aim: construct a dd$\mathbf{Bicat}_s$-category ΣB from a braided monoidal category B

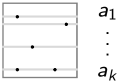


Objects: horizontal path classes of points in I^2, labelled by objects of B

Morphisms (cf strictification): start by evaluating the configuration as a word

- similarity: there are many different ways to do so
- difference: they are not uniquely isomorphic "not all diagrams commute"

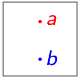
Aim: construct a dd**Bicat**$_s$-category ΣB from a braided monoidal category B



Objects: horizontal path classes of points in I^2, labelled by objects of B

Morphisms (cf strictification): start by evaluating the configuration as a word

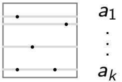
- similarity: there are many different ways to do so
- difference: they are not uniquely isomorphic "not all diagrams commute"



$\longmapsto a \otimes b$

3. Construction of ΣB

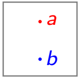
Aim: construct a dd$\mathbf{Bicat}_s$-category ΣB from a braided monoidal category B



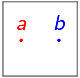
Objects: horizontal path classes of points in I^2, labelled by objects of B

Morphisms (cf strictification): start by evaluating the configuration as a word

- similarity: there are many different ways to do so
- difference: they are not uniquely isomorphic   “not all diagrams commute”



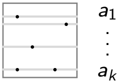
$\longmapsto a \otimes b$



$\longmapsto$?

3. Construction of ΣB

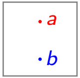
Aim: construct a dd**Bicat**$_s$-category ΣB from a braided monoidal category B



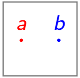
Objects: horizontal path classes of points in I^2, labelled by objects of B

Morphisms (cf strictification): start by evaluating the configuration as a word

- similarity: there are many different ways to do so
- difference: they are not uniquely isomorphic "not all diagrams commute"



$\longmapsto a \otimes b$



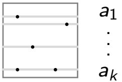
$\longmapsto$?

clockwise	$a \otimes b$
anti-clockwise	$b \otimes a$

3. Construction of ΣB

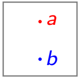
Aim: construct a dd**Bicat**$_s$-category ΣB from a braided monoidal category B

Objects: horizontal path classes of points in I^2, labelled by objects of B

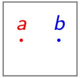


Morphisms (cf strictification): start by evaluating the configuration as a word

- similarity: there are many different ways to do so
- difference: they are not uniquely isomorphic   "not all diagrams commute"



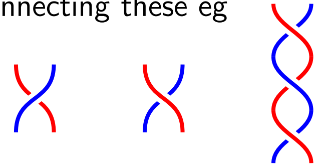
$\longmapsto a \otimes b$



$\longmapsto$?

clockwise	$a \otimes b$
anti-clockwise	$b \otimes a$

There are many isomorphisms connecting these eg



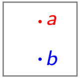
3. Construction of ΣB

Aim: construct a dd**Bicat**$_s$-category ΣB from a braided monoidal category B

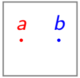
Objects: horizontal path classes of points in I^2, labelled by objects of B

Morphisms (cf strictification): start by evaluating the configuration as a word

- similarity: there are many different ways to do so
- difference: they are not uniquely isomorphic "not all diagrams commute"



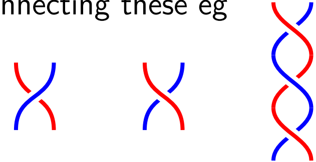
$\longmapsto a \otimes b$



$\longmapsto$?

clockwise	$a \otimes b$
anti-clockwise	$b \otimes a$

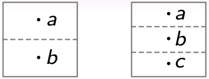
There are many isomorphisms connecting these eg



Solution: remember the journey, not just the destination.

3. Construction of ΣB

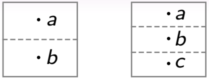
- The free braided monoidal category embeds vertically:



$a \otimes b$ $\qquad$ $a \otimes b \otimes c$ $\qquad \ldots$

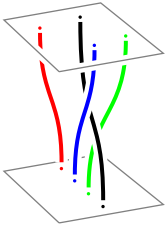
3. Construction of ΣB

- The free braided monoidal category embeds vertically:



$a \otimes b$ $\quad$ $a \otimes b \otimes c$ $\quad \ldots$

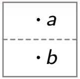
- We "flatten" our configuration to a canonical vertical one and remember what braid we used to do it.



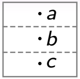
"flattening braid"

3. Construction of ΣB

- The free braided monoidal category embeds vertically:



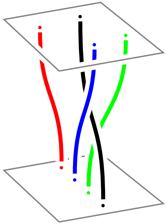
$a \otimes b$



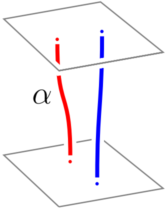
$a \otimes b \otimes c$

. . .

- We "flatten" our configuration to a canonical vertical one and remember what braid we used to do it.



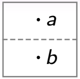
"flattening braid"



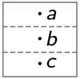
clockwise
$a \otimes b$

3. Construction of ΣB

- The free braided monoidal category embeds vertically:



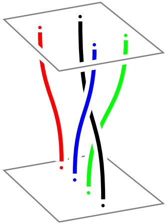
$a \otimes b$



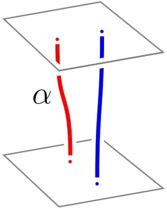
$a \otimes b \otimes c$

. . .

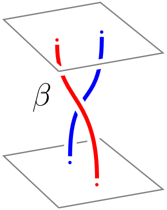
- We "flatten" our configuration to a canonical vertical one and remember what braid we used to do it.



"flattening braid"



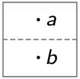
clockwise
$a \otimes b$



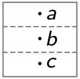
anti-clockwise
$b \otimes a$

3. Construction of ΣB

- The free braided monoidal category embeds vertically:



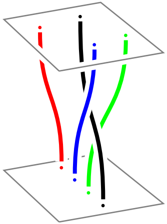
$a \otimes b$



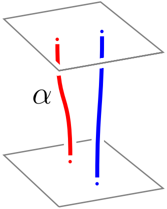
$a \otimes b \otimes c$

$\ldots$

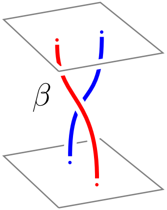
- We "flatten" our configuration to a canonical vertical one and remember what braid we used to do it.



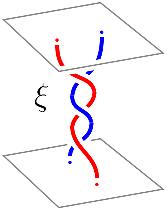
"flattening braid"



clockwise
$a \otimes b$



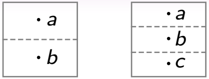
anti-clockwise
$b \otimes a$



hilarious
$b \otimes a$

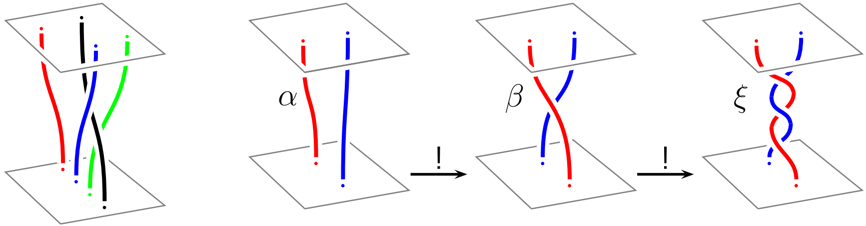
3. Construction of ΣB

- The free braided monoidal category embeds vertically:



$a \otimes b$ $\qquad$ $a \otimes b \otimes c$ $\qquad$ ...

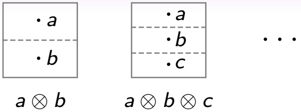
- We "flatten" our configuration to a canonical vertical one and remember what braid we used to do it.



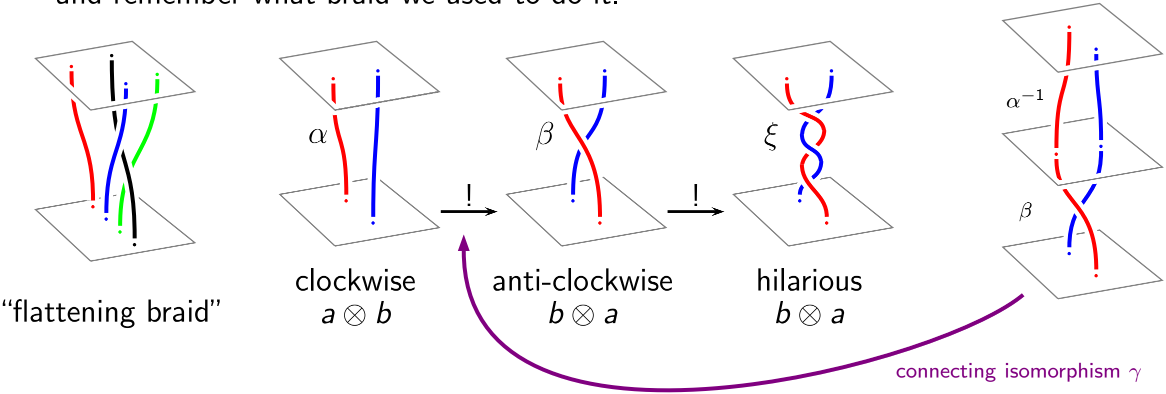
"flattening braid" $\qquad$ clockwise $a \otimes b$ $\qquad$ anti-clockwise $b \otimes a$ $\qquad$ hilarious $b \otimes a$

3. Construction of ΣB

- The free braided monoidal category embeds vertically:



- We "flatten" our configuration to a canonical vertical one and remember what braid we used to do it.



3. Construction of ΣB: morphisms

	$\mathbf{st}\,B$	ΣB
Objects:	strings of objects of B	configurations of points labelled by objects of B

	st B	ΣB
Objects:	strings of objects of B	configurations of points labelled by objects of B

Morphisms:

- find associated clique in FB

of bracketed words in B

of flattened configurations with flattening braids

3. Construction of ΣB: morphisms

	$\mathbf{st}\,B$	ΣB
Objects:	strings of objects of B	configurations of points labelled by objects of B

Morphisms:

- find associated clique in FB

  of bracketed words in B

  of flattened configurations with flattening braids

- evaluate it as a clique in B and take clique maps in B

3. Construction of ΣB: morphisms

	st B	ΣB
Objects:	strings of objects of B	configurations of points labelled by objects of B

Morphisms:

- find associated clique in FB

  of bracketed words in B | of flattened configurations with flattening braids

- evaluate it as a clique in B and take clique maps in B

these are the morphisms of ΣB

3. Construction of ΣB: morphisms

	$\mathbf{st}\,B$	ΣB
Objects:	strings of objects of B	configurations of points labelled by objects of B

Morphisms:

- find associated clique in FB

  of bracketed words in B — of flattened configurations with flattening braids

- evaluate it as a clique in B and take clique maps in B ← these are the morphisms of ΣB

- morphisms represent the same clique map if they differ only by

  coherence constraints — the correct factorising braid

3. Construction of ΣB: example morphism

$\mathbf{st}\, B$ $\qquad\qquad\qquad$ ΣB

$$abc \xrightarrow{?} abc$$

3. Construction of ΣB: example morphism

$\mathbf{st}\, B$ $\qquad\qquad$ ΣB

$$abc \xrightarrow{?} abc$$

rep by $\qquad (ab)c \xrightarrow{\mathrm{id}} (ab)c$

3. Construction of ΣB: example morphism

$\mathbf{st}\, B$ $\qquad\qquad\qquad\qquad$ ΣB

$$abc \xrightarrow{?} abc$$

rep by $\qquad (ab)c \xrightarrow{\mathrm{id}} (ab)c$

or $\qquad (ab)c \xrightarrow{a} a(bc)$

3. Construction of ΣB: example morphism

$\mathbf{st}\,B$ $\qquad\qquad\qquad$ ΣB

$$abc \xrightarrow{?} abc$$

$$\begin{tikzcd}
\text{rep by} & (ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \\
\text{or} & (ab)c \arrow[r, "a"] & a(bc)
\end{tikzcd}$$

3. Construction of ΣB: example morphism

$\mathbf{st}\,B$ $\qquad\qquad\qquad$ ΣB

$$abc \xrightarrow{?} abc$$

rep by

$$\begin{tikzcd}
(ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "a"] \\
(ab)c \arrow[r, "a"] & a(bc)
\end{tikzcd}$$

or

3. Construction of ΣB: example morphism

$\mathbf{st}\, B$ $\qquad\qquad\qquad\qquad$ ΣB

$$abc \xrightarrow{?} abc$$

$$\begin{tikzcd}
\text{rep by} & (ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "\mathfrak{a}"] \\
\text{or} & (ab)c \arrow[r, "\mathfrak{a}"] & a(bc)
\end{tikzcd}$$

— same clique map

3. Construction of ΣB: example morphism

st B

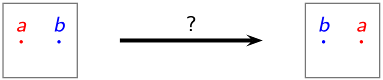
$$abc \xrightarrow{?} abc$$

rep by / or

$$\begin{tikzcd} (ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "a"] \\ (ab)c \arrow[r, "a"] & a(bc) \end{tikzcd}$$

— same clique map

ΣB



st B

$$abc \xrightarrow{?} abc$$

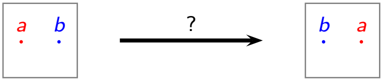
rep by

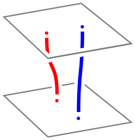
$$\begin{tikzcd} (ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "a"] \\ (ab)c \arrow[r, "a"] & a(bc) \end{tikzcd}$$

or

— same clique map

ΣB





$a \otimes b$

3. Construction of ΣB: example morphism

st B

$$abc \xrightarrow{?} abc$$

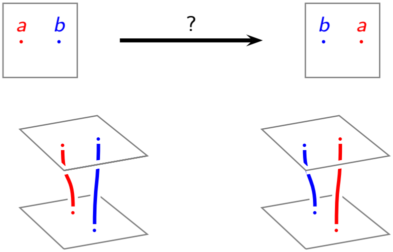
rep by

$$\begin{tikzcd}
(ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "\mathfrak{a}"] \\
(ab)c \arrow[r, "\mathfrak{a}"] & a(bc)
\end{tikzcd}$$

or

— same clique map

ΣB



$a \otimes b$ $\qquad$ $b \otimes a$

3. Construction of ΣB: example morphism

st B

$$abc \xrightarrow{?} abc$$

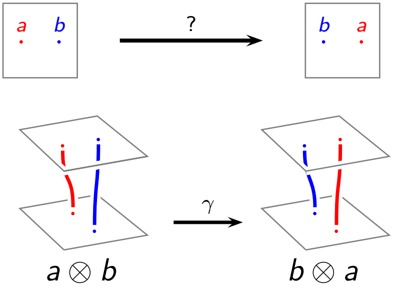
rep by

$$\begin{tikzcd} (ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "\mathfrak{a}"] \\ (ab)c \arrow[r, "\mathfrak{a}"] & a(bc) \end{tikzcd}$$

or

— same clique map

ΣB



3. Construction of ΣB: example morphism

st B

$$abc \xrightarrow{?} abc$$

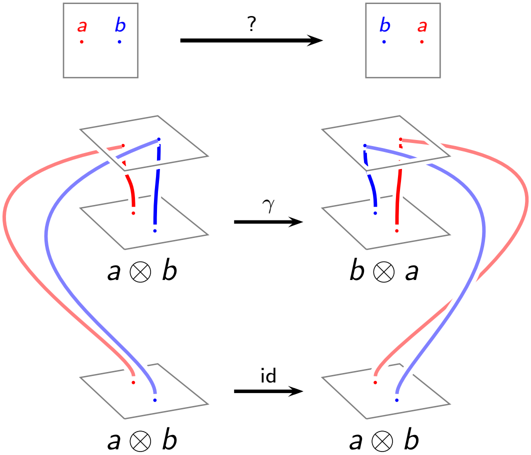
rep by

$$\begin{tikzcd} (ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "a"] \\ (ab)c \arrow[r, "a"] & a(bc) \end{tikzcd}$$

or

— same clique map

ΣB



3. Construction of ΣB: example morphism

$\mathbf{st}\, B$

$$abc \xrightarrow{?} abc$$

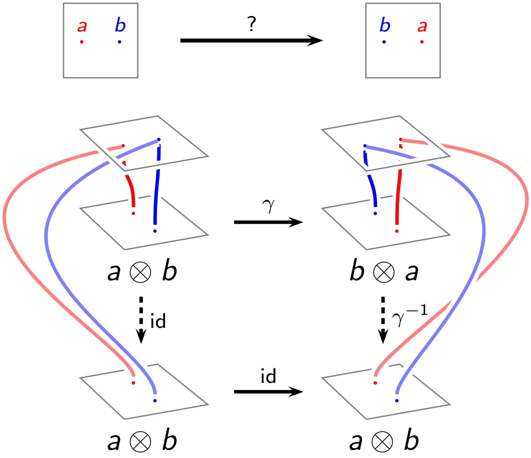
rep by

$$\begin{tikzcd}
(ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "\mathfrak{a}"] \\
(ab)c \arrow[r, "\mathfrak{a}"] & a(bc)
\end{tikzcd}$$

or

— same clique map

ΣB



— same clique map

3. Construction of ΣB: example morphism

st B

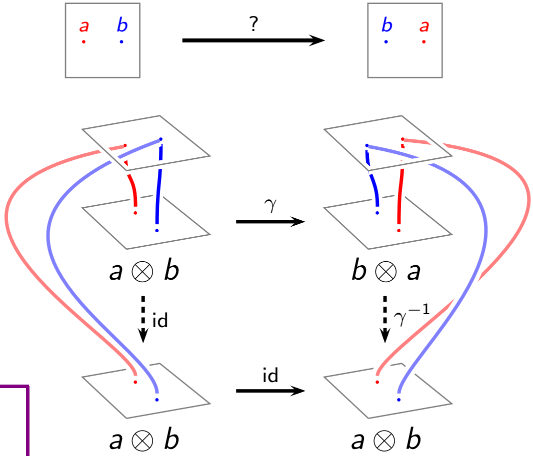
$$abc \xrightarrow{?} abc$$

rep by / or

$$\begin{tikzcd}
(ab)c \arrow[r, "\mathrm{id}"] \arrow[d, dashed, "\mathrm{id}"'] & (ab)c \arrow[d, dashed, "a"] \\
(ab)c \arrow[r, "a"] & a(bc)
\end{tikzcd}$$

— same clique map

ΣB



— same clique map

NB: identity in ΣB can be represented by a non-identity in B and vice versa

3. Construction of ΣB: technically

Write $\mathcal{O}$ for the objects of B, $\mathcal{F}\mathcal{O}$ for the free braided monoidal category on $\mathcal{O}$.

3. Construction of ΣB: technically

Write $\mathcal{O}$ for the objects of B, $\mathcal{F}\mathcal{O}$ for the free braided monoidal category on $\mathcal{O}$.

We have functors

$$\Pi_1 C(I^2, \mathcal{O}) \xleftarrow[\sim]{F} \mathcal{F}\mathcal{O}$$

3. Construction of ΣB: technically

Write $\mathcal{O}$ for the objects of B, $\mathcal{F}\mathcal{O}$ for the free braided monoidal category on $\mathcal{O}$.

We have functors

$$\Pi_1 C(I^2, \mathcal{O}) \xleftarrow[\sim]{F} \mathcal{F}\mathcal{O} \xrightarrow[\text{eval}]{G} B$$

3. Construction of ΣB: technically

Write $\mathcal{O}$ for the objects of B, $\mathcal{F}\mathcal{O}$ for the free braided monoidal category on $\mathcal{O}$.

We have functors

$$\Pi_1 C(I^2, \mathcal{O}) \xleftarrow[\sim]{F} \mathcal{F}\mathcal{O} \xrightarrow[\text{eval}]{G} B$$

inducing functors on clique categories

$$\Pi_1 \widetilde{C(I^2, \mathcal{O})} \xrightarrow{F^*} \widetilde{\mathcal{F}\mathcal{O}} \xrightarrow{G_!} \widetilde{B}$$

Write $\mathcal{O}$ for the objects of B, $\mathcal{F}\mathcal{O}$ for the free braided monoidal category on $\mathcal{O}$.

We have functors

$$\Pi_1 C(I^2, \mathcal{O}) \xleftarrow[\sim]{F} \mathcal{F}\mathcal{O} \xrightarrow[\text{eval}]{G} B$$

inducing functors on clique categories

$$\Pi_1 \widetilde{C(I^2, \mathcal{O})} \xrightarrow{F^*} \widetilde{\mathcal{F}\mathcal{O}} \xrightarrow{G_!} \widetilde{B}$$

ΣB is defined by

- objects: horizontal path cliques of $\Pi_1 C(I^2, \mathcal{O})$
- morphisms:

$$\Sigma B(\overline{X}, \overline{Y}) := \widetilde{B}(G_! F^* \overline{X}, G_! F^* \overline{Y})$$

3. Construction of ΣB: "horizontal and vertical composition are both $\otimes$"

3. Construction of ΣB: "horizontal and vertical composition are both $\otimes$"

Given $\overline{a} \xrightarrow{\overline{f}} \overline{a}'$ and $\overline{b} \xrightarrow{\overline{g}} \overline{b}'$

- for any components f and g respectively, take $f \otimes g$

3. Construction of ΣB: "horizontal and vertical composition are both $\otimes$"

Given $\overline{a} \xrightarrow{\overline{f}} \overline{a}'$ and $\overline{b} \xrightarrow{\overline{g}} \overline{b}'$

- for any components f and g respectively, take $f \otimes g$
- we need to specify what component of the clique map it is
  i.e. what flattening braids it refers to

3. Construction of ΣB: "horizontal and vertical composition are both $\otimes$"

Given $\overline{a} \xrightarrow{\overline{f}} \overline{a}'$ and $\overline{b} \xrightarrow{\overline{g}} \overline{b}'$

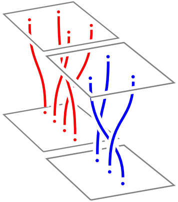
- for any components f and g respectively, take $f \otimes g$
- we need to specify what component of the clique map it is
  i.e. what flattening braids it refers to
- we know which flattening braids are referred to by f and g

Given $\overline{a} \xrightarrow{\overline{f}} \overline{a}'$ and $\overline{b} \xrightarrow{\overline{g}} \overline{b}'$

- for any components f and g respectively, take $f \otimes g$
- we need to specify what component of the clique map it is i.e. what flattening braids it refers to
- we know which flattening braids are referred to by f and g

Vertical composition:

stack braids vertically



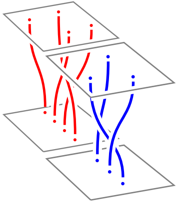
3. Construction of ΣB: "horizontal and vertical composition are both $\otimes$"

Given $\overline{a} \xrightarrow{\overline{f}} \overline{a}'$ and $\overline{b} \xrightarrow{\overline{g}} \overline{b}'$

- for any components f and g respectively, take $f \otimes g$
- we need to specify what component of the clique map it is i.e. what flattening braids it refers to
- we know which flattening braids are referred to by f and g

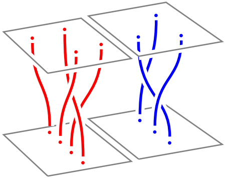
Vertical composition:

stack braids vertically



Horizonal composition:

stack braids horizontally...



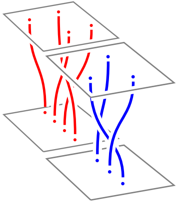
3. Construction of ΣB: "horizontal and vertical composition are both $\otimes$"

Given $\overline{a} \xrightarrow{\overline{f}} \overline{a}'$ and $\overline{b} \xrightarrow{\overline{g}} \overline{b}'$

- for any components f and g respectively, take $f \otimes g$
- we need to specify what component of the clique map it is i.e. what flattening braids it refers to
- we know which flattening braids are referred to by f and g

Vertical composition:

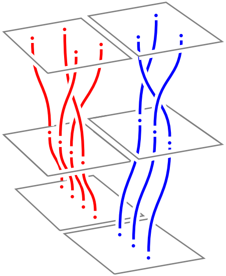
stack braids vertically



Horizonal composition:

stack braids horizontally. . .

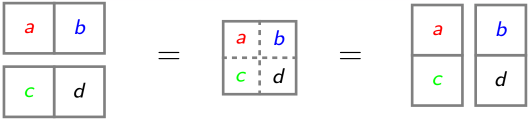
. . .and twist



3. Construction of ΣB: interchange

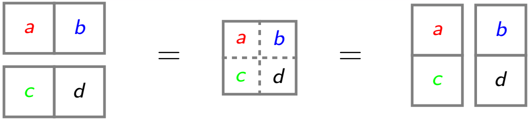
3. Construction of ΣB: interchange

On objects:

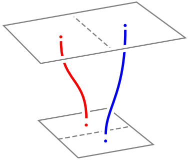


3. Construction of ΣB: interchange

On objects:

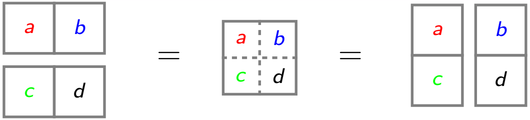


On morphisms: by the method above we get different representatives

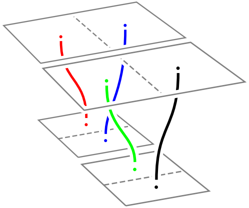


3. Construction of ΣB: interchange

On objects:

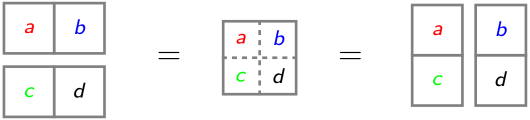


On morphisms: by the method above we get different representatives

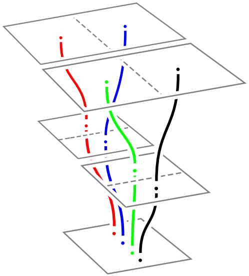


3. Construction of ΣB: interchange

On objects:

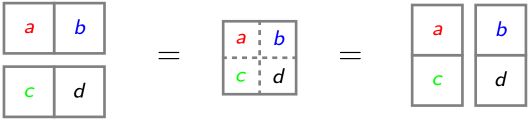


On morphisms: by the method above we get different representatives

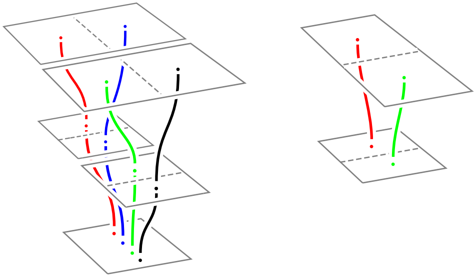


3. Construction of ΣB: interchange

On objects:

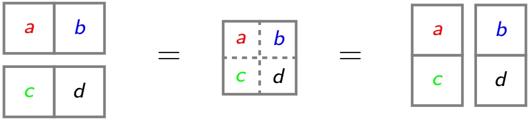


On morphisms: by the method above we get different representatives

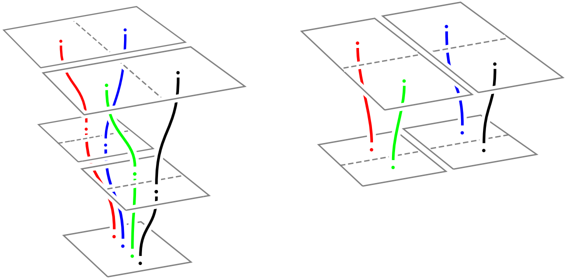


3. Construction of ΣB: interchange

On objects:

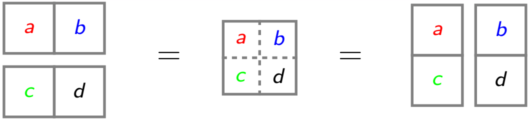


On morphisms: by the method above we get different representatives

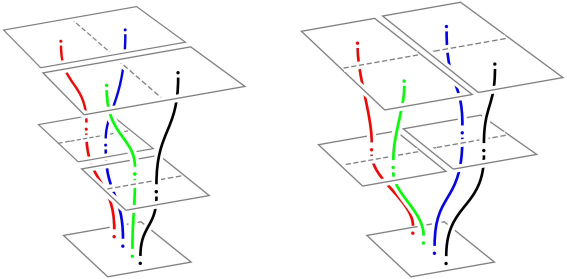


3. Construction of ΣB: interchange

On objects:

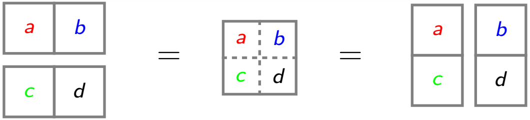


On morphisms: by the method above we get different representatives

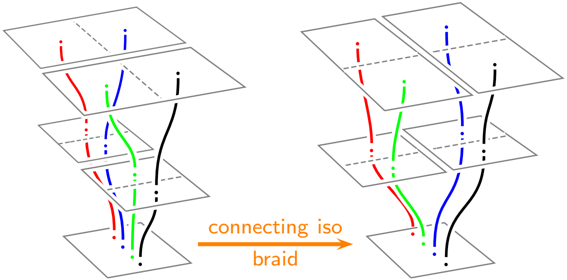


3. Construction of ΣB: interchange

On objects:

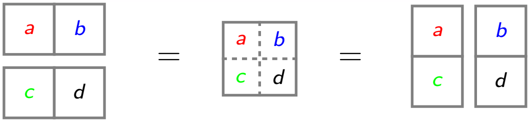


On morphisms: by the method above we get different representatives

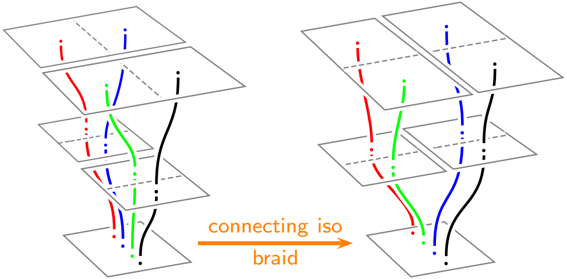


3. Construction of ΣB: interchange

On objects:



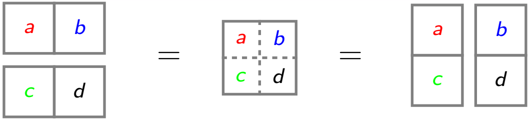
On morphisms: by the method above we get different representatives



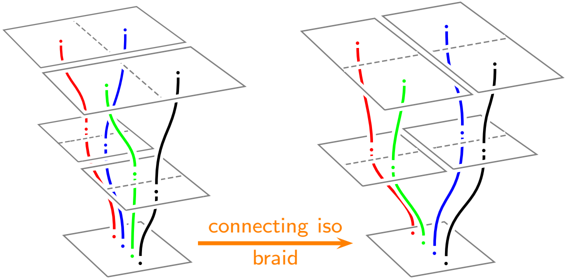
$$
\begin{array}{c}
a \otimes b \otimes c \otimes d \\
\Big\downarrow {\scriptstyle f \otimes g \otimes h \otimes i} \\
a' \otimes b' \otimes c' \otimes d'
\end{array}
$$

3. Construction of ΣB: interchange

On objects:



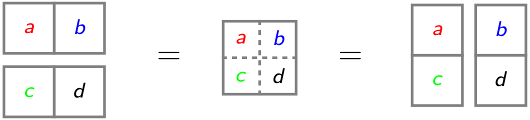
On morphisms: by the method above we get different representatives



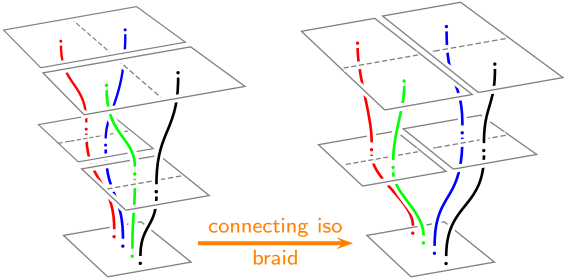
$$\begin{tikzcd} a \otimes b \otimes c \otimes d \arrow[d, "f \otimes g \otimes h \otimes i"'] \\ a' \otimes b' \otimes c' \otimes d' \end{tikzcd} \qquad \begin{tikzcd} a \otimes c \otimes b \otimes d \arrow[d, "f \otimes h \otimes g \otimes i"] \\ a' \otimes c' \otimes b' \otimes d' \end{tikzcd}$$

3. Construction of ΣB: interchange

On objects:



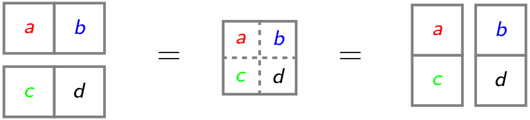
On morphisms: by the method above we get different representatives



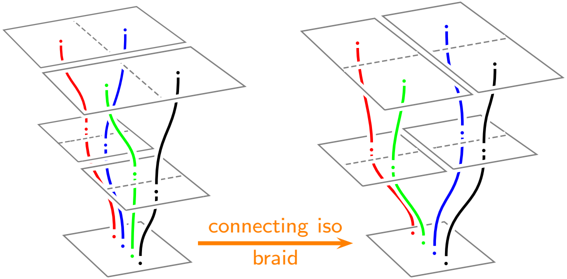
$$
\begin{tikzcd}
a \otimes b \otimes c \otimes d \arrow[r, "1 \otimes \gamma \otimes 1"] \arrow[d, "f \otimes g \otimes h \otimes i"'] & a \otimes c \otimes b \otimes d \arrow[d, "f \otimes h \otimes g \otimes i"] \\
a' \otimes b' \otimes c' \otimes d' \arrow[r, "1 \otimes \gamma \otimes 1"'] & a' \otimes c' \otimes b' \otimes d'
\end{tikzcd}
$$

— same clique map

On objects:



On morphisms: by the method above we get different representatives



$$
\begin{tikzcd}
a \otimes b \otimes c \otimes d \arrow[r, "1 \otimes \gamma \otimes 1"] \arrow[d, "f \otimes g \otimes h \otimes i"'] & a \otimes c \otimes b \otimes d \arrow[d, "f \otimes h \otimes g \otimes i"] \\
a' \otimes b' \otimes c' \otimes d' \arrow[r, "1 \otimes \gamma \otimes 1"'] & a' \otimes c' \otimes b' \otimes d'
\end{tikzcd}
$$

— same clique map

Interchange is strict but still comes from the braiding.

4. Braided monoidal equivalence $B \xrightarrow{\sim} U\Sigma B$

1. Define functor

3. Monoidal equivalence

2. Equivalence of categories

4. Braided monoidal equivalence $B \xrightarrow{\sim} U\Sigma B$

1. Define functor

$$a \longmapsto \boxed{\bullet a}$$

$$\begin{array}{c} a \\ \downarrow \\ b \end{array} \longmapsto \begin{array}{c} \text{clique map} \\ \text{represented by} \\ a \longrightarrow b \end{array}$$

3. Monoidal equivalence

2. Equivalence of categories

4. Braided monoidal equivalence $B \xrightarrow{\sim} U\Sigma B$

1. Define functor

$$a \longmapsto \boxed{\bullet a}$$

$$\begin{array}{c} a \\ \downarrow \\ b \end{array} \longmapsto \begin{array}{c} \text{clique map} \\ \text{represented by} \\ a \longrightarrow b \end{array}$$

3. Monoidal equivalence

2. Equivalence of categories

- full and faithful by construction

1. Define functor

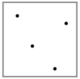
$a \longmapsto$ 

$$\begin{array}{c} a \\ \downarrow \\ b \end{array} \longmapsto$$ clique map represented by $a \longrightarrow b$

3. Monoidal equivalence

2. Equivalence of categories

- full and faithful by construction
- essentially surjective on objects:

Given  $\begin{array}{c} a_1 \\ \vdots \\ a_k \end{array} \in U\Sigma B$

1. Define functor

$$a \longmapsto \boxed{\bullet_a}$$

$$\begin{array}{c} a \\ \downarrow \\ b \end{array} \longmapsto \text{clique map represented by } a \longrightarrow b$$

3. Monoidal equivalence

2. Equivalence of categories

- full and faithful by construction
- essentially surjective on objects:

Given $\boxed{\cdot\ \cdot\ \cdot\ \cdot}\ \begin{array}{c} a_1 \\ \vdots \\ a_k \end{array} \in U\Sigma B$

$$a_1 \otimes \cdots \otimes a_k \in B \longmapsto \boxed{\bullet_{a_1 \otimes \cdots \otimes a_k}}$$

1. Define functor

$$a \longmapsto \boxed{\bullet a}$$

$$\begin{array}{c} a \\ \downarrow \\ b \end{array} \longmapsto \begin{array}{c} \text{clique map} \\ \text{represented by} \\ a \longrightarrow b \end{array}$$

3. Monoidal equivalence

2. Equivalence of categories

- full and faithful by construction
- essentially surjective on objects:

Given $\boxed{\cdot \ \cdot \ \cdot \ \cdot}\ \begin{array}{c} a_1 \\ \vdots \\ a_k \end{array} \in U\Sigma B$

$$a_1 \otimes \cdots \otimes a_k \in B \longmapsto \boxed{\bullet}\ a_1 \otimes \cdots \otimes a_k$$

and we have

$$\boxed{\cdot \ \cdot \ \cdot \ \cdot}\ \begin{array}{c} a_1 \\ \vdots \\ a_k \end{array} \xrightarrow{\sim} \boxed{\bullet}\ a_k \otimes \cdots \otimes a_k$$

1. Define functor

$a \longmapsto \boxed{\bullet a}$

$a \to b \longmapsto$ clique map represented by $a \longrightarrow b$

3. Monoidal equivalence

2. Equivalence of categories

- full and faithful by construction
- essentially surjective on objects:

Given $a_1, \ldots, a_k \in U\Sigma B$

$a_1 \otimes \cdots \otimes a_k \in B \longmapsto \boxed{\bullet}\ a_1 \otimes \cdots \otimes a_k$

and we have

$a_1, \ldots, a_k \xrightarrow{\sim} \boxed{\bullet}\ a_k \otimes \cdots \otimes a_k$

This isomorphism is a clique map represented by an identity.

1. Define functor

$a \longmapsto$ $\bullet a$

$a \downarrow b \longmapsto$ clique map represented by $a \longrightarrow b$

3. Monoidal equivalence

Need

$\cdot a$ / $\cdot b$ $\xrightarrow{\sim}$ $\bullet$ $a \otimes b$

2. Equivalence of categories

- full and faithful by construction
- essentially surjective on objects:

Given $a_1, \ldots, a_k$ $\in U\Sigma B$

$a_1 \otimes \cdots \otimes a_k \in B \longmapsto$ $\bullet$ $a_1 \otimes \cdots \otimes a_k$

and we have

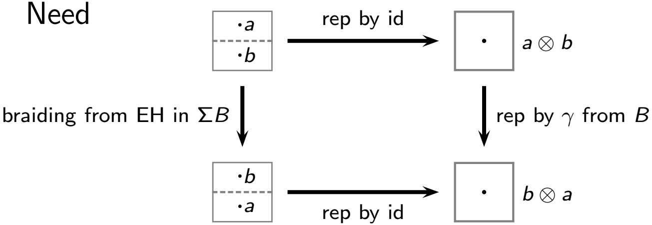
$a_1, \ldots, a_k$ $\xrightarrow{\sim}$ $\bullet$ $a_k \otimes \cdots \otimes a_k$

This isomorphism is a clique map represented by an identity.

4. Braided monoidal equivalence

4. Braided monoidal equivalence

Need



4. Braided monoidal equivalence

Need

$$\begin{tikzcd}[column sep=huge, row sep=large]
\begin{array}{|c|}\hline \cdot a \\ \hdashline \cdot b \\ \hline\end{array} \arrow[r, "\text{rep by id}"] \arrow[d, "\text{braiding from EH in } \Sigma B"'] & \boxed{\cdot}\ a \otimes b \arrow[d, "\text{rep by } \gamma \text{ from } B"] \\
\begin{array}{|c|}\hline \cdot b \\ \hdashline \cdot a \\ \hline\end{array} \arrow[r, "\text{rep by id}"'] & \boxed{\cdot}\ b \otimes a
\end{tikzcd}$$

— commuting diagram of clique maps

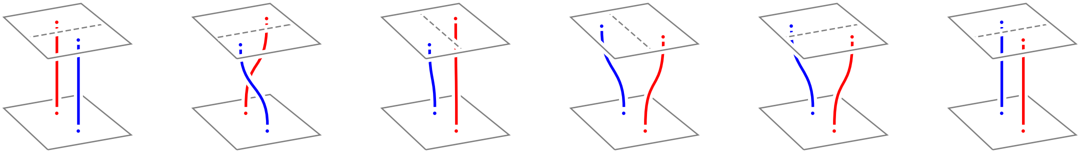
4. Braided monoidal equivalence

Need

$$\begin{tikzcd}
\boxed{\begin{smallmatrix}\cdot a \\ \cdot b\end{smallmatrix}} \arrow[rr, "\text{rep by id}"] \arrow[d, "\text{braiding from EH in } \Sigma B"'] & & \boxed{\cdot}\ a \otimes b \arrow[d, "\text{rep by } \gamma \text{ from } B"] \\
\boxed{\begin{smallmatrix}\cdot b \\ \cdot a\end{smallmatrix}} \arrow[rr, "\text{rep by id}"'] & & \boxed{\cdot}\ b \otimes a
\end{tikzcd}$$

— commuting diagram of clique maps

EH in ΣB:



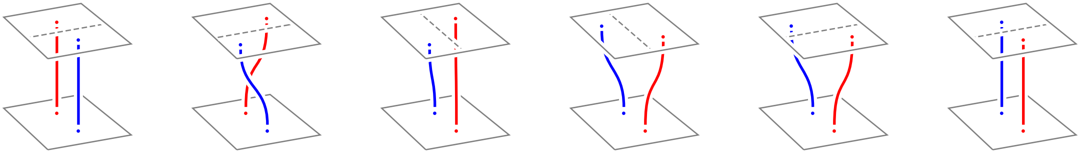
4. Braided monoidal equivalence

Need

$$\begin{tikzcd}
\boxed{\begin{matrix}\cdot a\\ \cdot b\end{matrix}} \arrow[r, "\text{rep by id}"] \arrow[d, "\text{braiding from EH in } \Sigma B"'] & \boxed{\cdot}\ a \otimes b \arrow[d, "\text{rep by } \gamma \text{ from } B"] \\
\boxed{\begin{matrix}\cdot b\\ \cdot a\end{matrix}} \arrow[r, "\text{rep by id}"'] & \boxed{\cdot}\ b \otimes a
\end{tikzcd}$$

— commuting diagram of clique maps

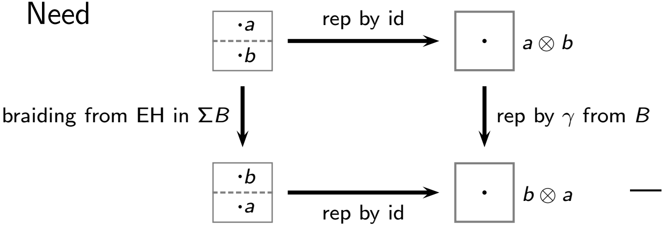
EH in ΣB:



clique maps: $\xrightarrow{\text{id}}$

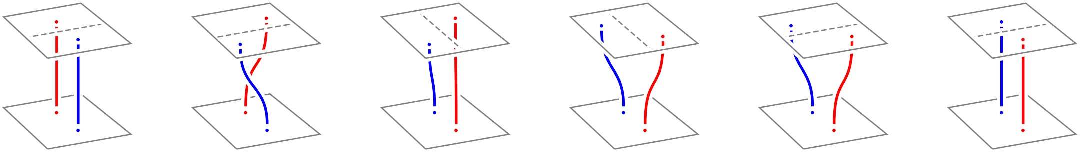
4. Braided monoidal equivalence

Need



— commuting diagram of clique maps

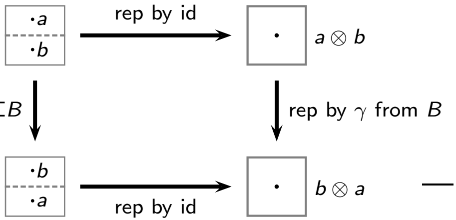
EH in ΣB:



clique maps: $\xrightarrow{\text{id}}$ $\xrightarrow{\text{id}}$

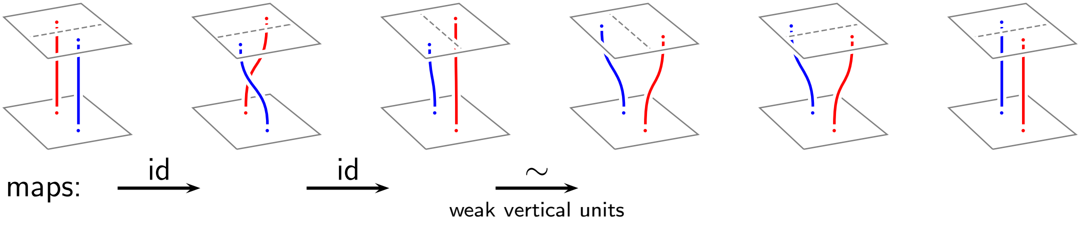
4. Braided monoidal equivalence

Need



— commuting diagram of clique maps

EH in ΣB:



clique maps:

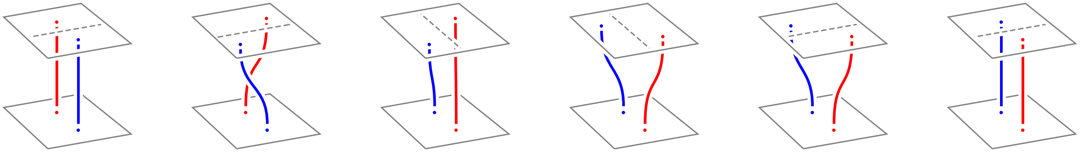
4. Braided monoidal equivalence

Need

$$\begin{tikzcd}
\boxed{\begin{smallmatrix}\cdot a \\ \hdashline \cdot b\end{smallmatrix}} \arrow[r, "\text{rep by id}"] \arrow[d, "\text{braiding from EH in } \Sigma B"'] & \boxed{\cdot}\ a \otimes b \arrow[d, "\text{rep by } \gamma \text{ from } B"] \\
\boxed{\begin{smallmatrix}\cdot b \\ \hdashline \cdot a\end{smallmatrix}} \arrow[r, "\text{rep by id}"'] & \boxed{\cdot}\ b \otimes a
\end{tikzcd}$$

— commuting diagram of clique maps

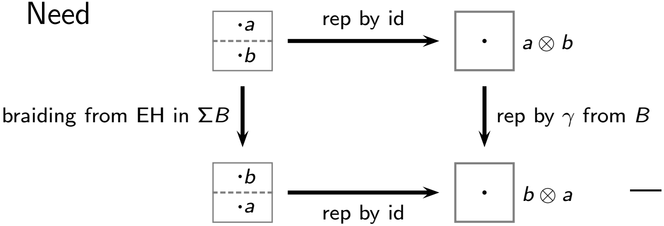
EH in ΣB:



clique maps: $\xrightarrow{\text{id}}$ $\xrightarrow{\text{id}}$ $\underset{\text{weak vertical units}}{\xrightarrow{\sim}}$ $\xrightarrow{\text{id}}$

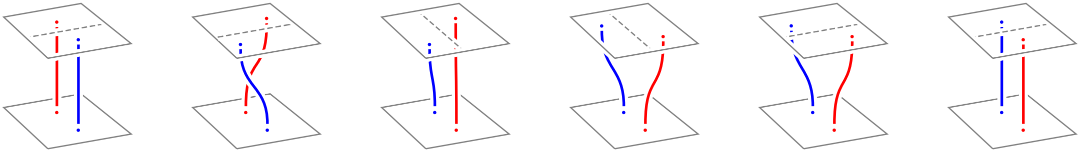
4. Braided monoidal equivalence

Need



— commuting diagram of clique maps

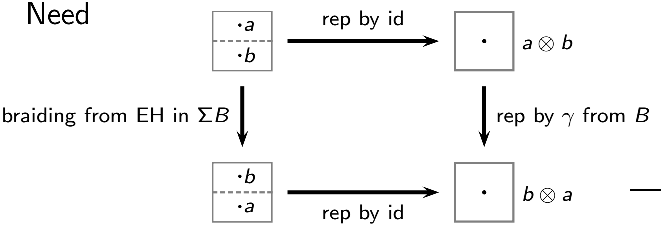
EH in ΣB:



clique maps: $\xrightarrow{\text{id}}$ $\xrightarrow{\text{id}}$ $\xrightarrow[\text{weak vertical units}]{\sim}$ $\xrightarrow{\text{id}}$ $\xrightarrow{\text{id}}$

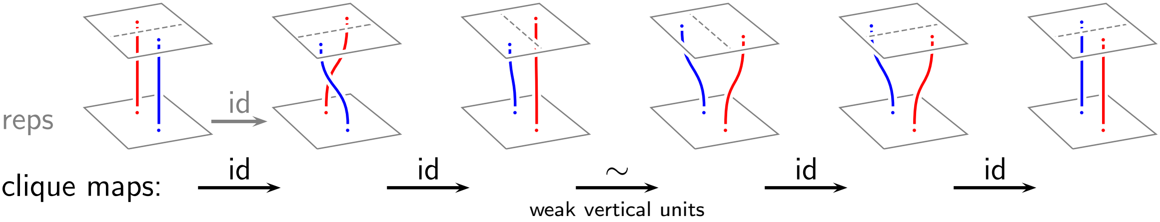
4. Braided monoidal equivalence

Need



— commuting diagram of clique maps

EH in ΣB:



4. Braided monoidal equivalence

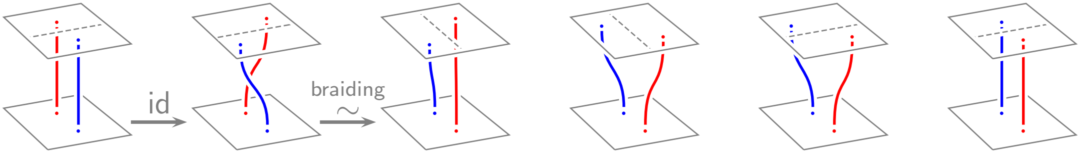
Need

$$\begin{tikzcd}
\boxed{\begin{array}{c}\cdot a \\ \hdashline \cdot b\end{array}} \arrow[r, "\text{rep by id}"] \arrow[d, "\text{braiding from EH in } \Sigma B"'] & \boxed{\cdot}\ a \otimes b \arrow[d, "\text{rep by } \gamma \text{ from } B"] \\
\boxed{\begin{array}{c}\cdot b \\ \hdashline \cdot a\end{array}} \arrow[r, "\text{rep by id}"'] & \boxed{\cdot}\ b \otimes a
\end{tikzcd}$$

— commuting diagram of clique maps

EH in ΣB:

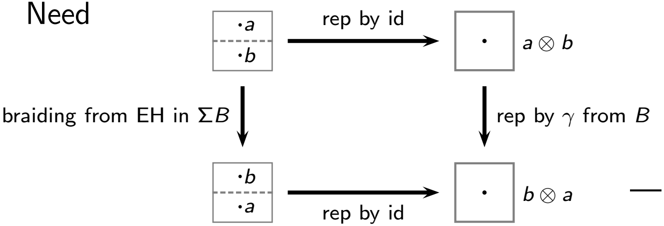
reps



clique maps: $\xrightarrow{\text{id}} \quad \xrightarrow{\text{id}} \quad \xrightarrow[\text{weak vertical units}]{\sim} \quad \xrightarrow{\text{id}} \quad \xrightarrow{\text{id}}$

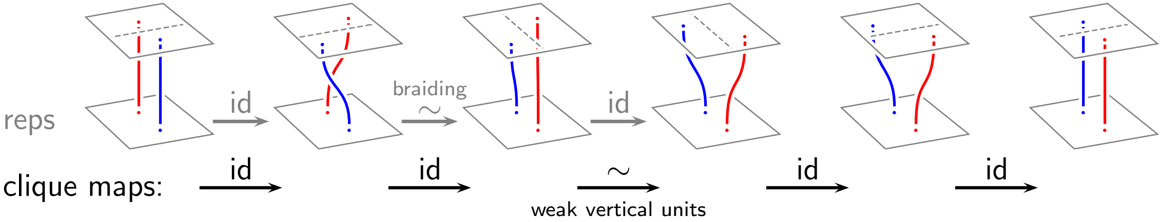
4. Braided monoidal equivalence

Need



— commuting diagram of clique maps

EH in ΣB:



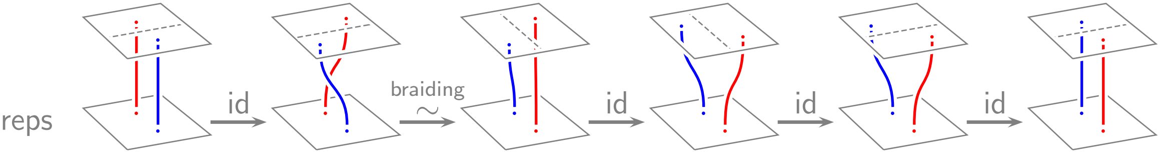
4. Braided monoidal equivalence

Need

$$\begin{tikzcd}
\boxed{\begin{smallmatrix}\cdot a \\ \hline \cdot b\end{smallmatrix}} \arrow[rr, "\text{rep by id}"] \arrow[d, "\text{braiding from EH in } \Sigma B"'] & & \boxed{\cdot}\ a \otimes b \arrow[d, "\text{rep by } \gamma \text{ from } B"] \\
\boxed{\begin{smallmatrix}\cdot b \\ \hline \cdot a\end{smallmatrix}} \arrow[rr, "\text{rep by id}"'] & & \boxed{\cdot}\ b \otimes a
\end{tikzcd}$$

— commuting diagram of clique maps

EH in ΣB:



clique maps: $\xrightarrow{\text{id}}$ $\xrightarrow{\text{id}}$ $\xrightarrow[\text{weak vertical units}]{\sim}$ $\xrightarrow{\text{id}}$ $\xrightarrow{\text{id}}$

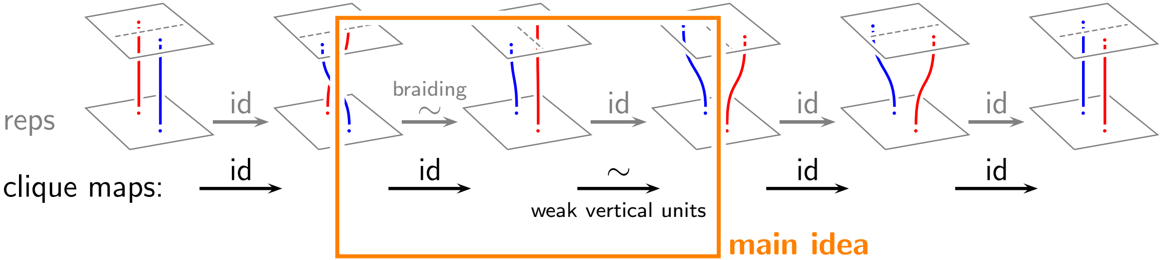
4. Braided monoidal equivalence

Need

$$\begin{tikzcd}
\boxed{\begin{smallmatrix}\cdot a\\ \hdashline \cdot b\end{smallmatrix}} \arrow[r, "\text{rep by id}"] \arrow[d, "\text{braiding from EH in }\Sigma B"'] & \boxed{\cdot}\ a\otimes b \arrow[d, "\text{rep by }\gamma\text{ from }B"] \\
\boxed{\begin{smallmatrix}\cdot b\\ \hdashline \cdot a\end{smallmatrix}} \arrow[r, "\text{rep by id}"'] & \boxed{\cdot}\ b\otimes a
\end{tikzcd}$$

— commuting diagram of clique maps

EH in ΣB:



Main ideas

- Putting points in boxes gives us enough control.
- Using cliques exchanges the roles of units and interchange.

Conclusion

Main ideas

- Putting points in boxes gives us enough control.
- Using cliques exchanges the roles of units and interchange.

Main results

- We have Σ on 0-cells.

$$\begin{tikzcd} \mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \arrow[r, bend left, "U"] & \mathrm{w}\mathbf{BrMonCat} \arrow[l, bend left, dashed, "\Sigma"] \end{tikzcd}$$

- It is biessentially surjective on objects.

Main ideas

- Putting points in boxes gives us enough control.
- Using cliques exchanges the roles of units and interchange.

Main results

- We have Σ on 0-cells.

$$\begin{tikzcd} \mathrm{dd}\mathbf{Bicat}_s\text{-}\mathbf{Cat} \arrow[r, bend left, "U"] & \mathrm{w}\mathbf{BrMonCat} \arrow[l, bend left, dashed, "\Sigma"] \end{tikzcd}$$

- It is biessentially surjective on objects.

$\boxed{\text{Weak vertical composition is enough to produce braidings.}}$

Done but no space in talk:

- Define weak functors of dd$\mathbf{Bicat}_s$-categories using abstract EH (CT18).
- Assemble these into a 2-category with icon-like transformations.
- Extend Σ to a pseudo-functor of 2-categories.
- Show that we have a biequivalence of 2-categories.
- Analogous results for Trimble 3-categories.

Future:

- Rotate and get weak horizontal composition and strict vertical.
- Produce free doubly-degenerate structures by composing adjunctions.
- The non-degenerate case.
- Higher dimensions.