Computational Descriptions of Higher Categories

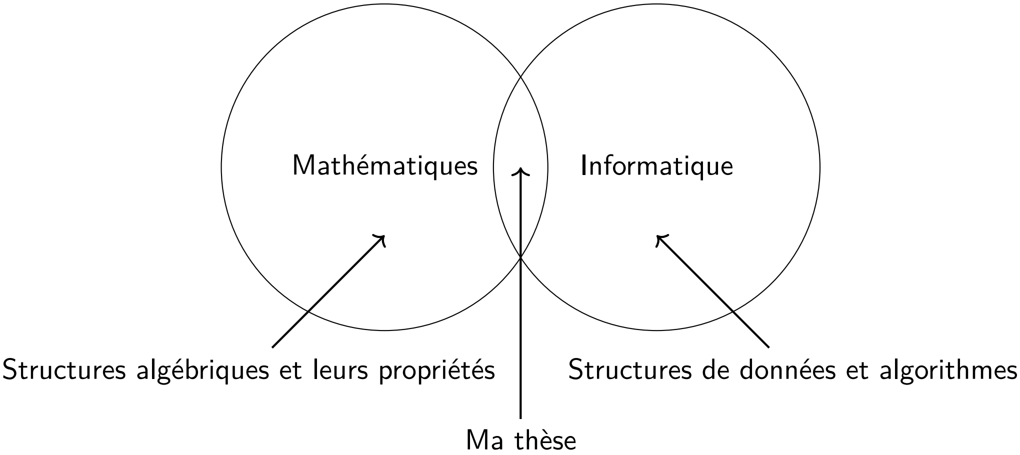
PhD defense

Simon Forest

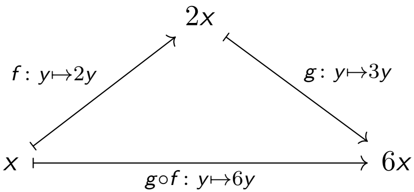
École Polytechnique, Université de Paris

January 8, 2021

Ma thèse en une slide



Composition de fonctions



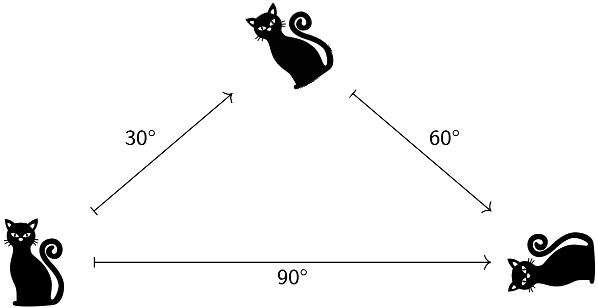
Composition de matrices

$$\mathbb{R}^3 \xrightarrow{M} \mathbb{R}^2 \xrightarrow{N} \mathbb{R}^2, \qquad \mathbb{R}^3 \xrightarrow{N \circ M} \mathbb{R}^2$$

$$M = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 3 \end{pmatrix} \qquad N = \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}$$

$$N \circ M = \begin{pmatrix} 5 & 1 & 6 \\ 6 & 0 & 9 \end{pmatrix}$$

Composition de transformations



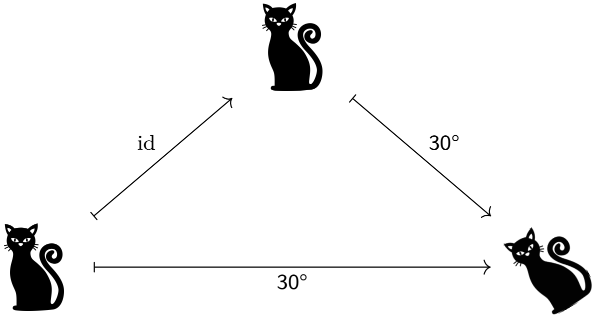
Composition de transformations : identité

Transformation identité id

id :  $\mapsto$ 

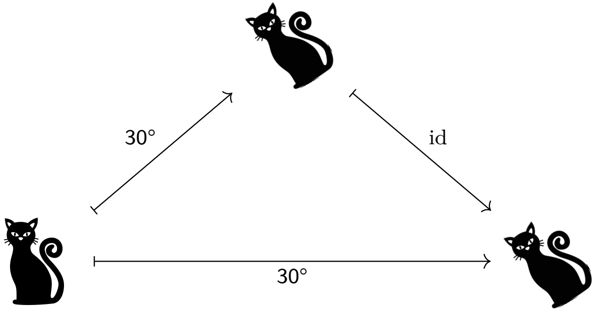
Composition de transformations : identité

Transformation identité id

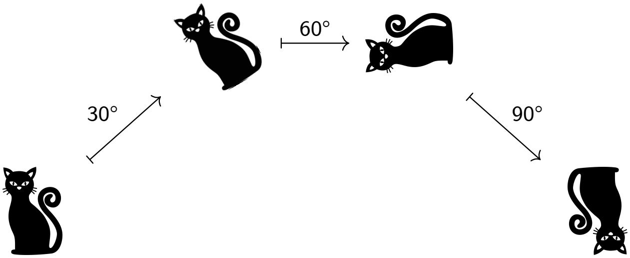


Composition de transformations : identité

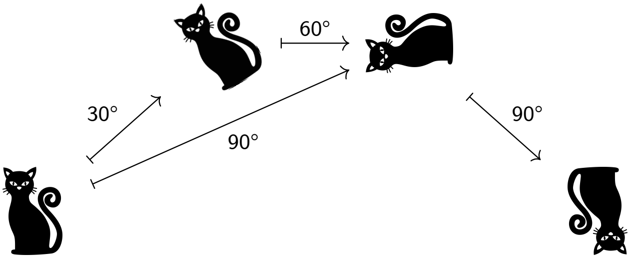
Transformation identité id



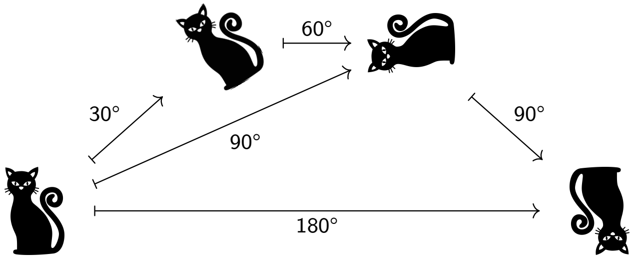
Composition de transformations : associativité



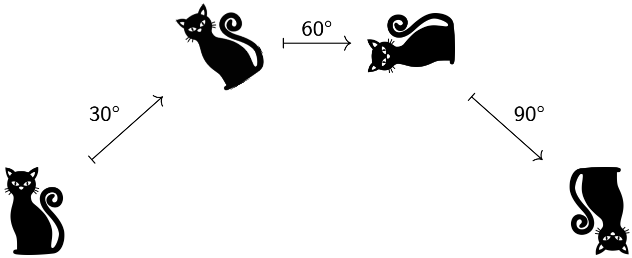
Composition de transformations : associativité



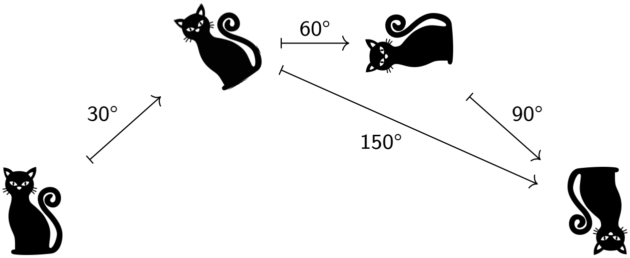
Composition de transformations : associativité



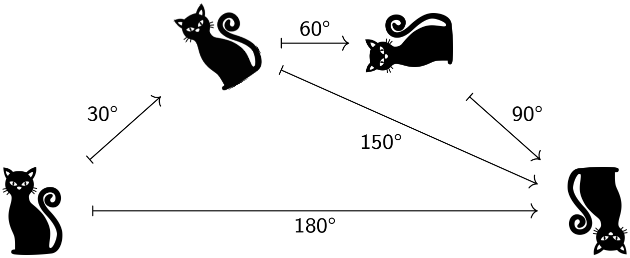
Composition de transformations : associativité



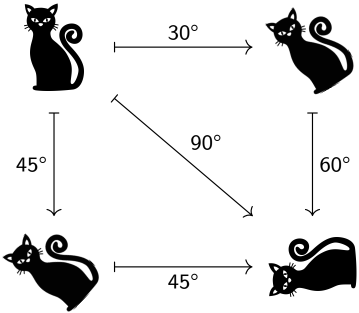
Composition de transformations : associativité



Composition de transformations : associativité



Composition de transformations : relations



Monoïdes

Un monoïde est la donnée d'un ensemble M et

$$e \in M \qquad \bullet \colon M \times M \to M$$

tels que

$$e \bullet x = x \qquad x \bullet e = x \qquad (x \bullet y) \bullet z = x \bullet (y \bullet z).$$

On souhaite pouvoir faire des calculs sur ces structures.

Monoïdes libres

$S = \{a, b, c\}$. Monoïde libre sur S ?

Il est constitué de tous les termes formels :

$$e, \quad a \bullet b, \quad (a \bullet c) \bullet (b \bullet e), \quad etc.$$

Monoïdes libres

$S = \{a, b, c\}$. Monoïde libre sur S ?

Il est constitué de tous les termes formels :

$$e, \quad a \bullet b, \quad (a \bullet c) \bullet (b \bullet e), \quad etc.$$

Cela se code très simplement :

```
type s = A | B | C
type terme =
| Gen of s
| E
| Mult of (terme * terme)
```

Non-unicité

Axiomes des monoïdes :

$$e \bullet x = x \qquad x \bullet e = x \qquad (x \bullet y) \bullet z = x \bullet (y \bullet z)$$

Non-unicité

Axiomes des monoïdes :

$$e \bullet x = x \qquad x \bullet e = x \qquad (x \bullet y) \bullet z = x \bullet (y \bullet z)$$

```
          Mult (E,Gen A)   "="  Gen A
          Mult (Gen A,E)   "="  Gen A
Mult (Mult (Gen A,Gen B),Gen C)   "="  Mult (Gen A,Mult {Gen B,Gen C})
                   ...
```

Comment tester l'égalité entre les termes ?

Réduction

Avec les nombres, comment tester l'égalité de deux expressions ? Par exemple

$$(3+4)+(0+4) \qquad 1+((2+3)+4)\ ?$$

Réduction

Avec les nombres, comment tester l'égalité de deux expressions ? Par exemple

$$(3+4)+(0+4) \qquad 1+((2+3)+4)\,?$$

Il suffit de réduire !

$$(3+4)+(0+3) \quad \rightsquigarrow \quad 7+(0+3) \quad \rightsquigarrow \quad 7+3 \quad \rightsquigarrow \quad 10$$

$$1+((2+3)+4) \quad \rightsquigarrow \quad 1+(5+4) \quad \rightsquigarrow \quad 1+9 \quad \rightsquigarrow \quad 10$$

Réduction

Pour les monoïdes, on va se donner également des règles de réduction

$$e \bullet x \rightsquigarrow x \qquad x \bullet e \rightsquigarrow x \qquad (x \bullet y) \bullet z \rightsquigarrow x \bullet (y \bullet z)$$

Réduction

Pour les monoïdes, on va se donner également des règles de réduction

$$e \bullet x \rightsquigarrow x \qquad x \bullet e \rightsquigarrow x \qquad (x \bullet y) \bullet z \rightsquigarrow x \bullet (y \bullet z)$$

et on peut ainsi tester l'égalité en réduisant

$$(b \bullet a) \bullet (e \bullet c) = ((e \bullet b) \bullet a) \bullet c\,?$$

$$\begin{array}{ccccc} (b \bullet a) \bullet (e \bullet c) & \rightsquigarrow & (b \bullet a) \bullet c & \rightsquigarrow & b \bullet (a \bullet c) \\ ((e \bullet b) \bullet a) \bullet c & \rightsquigarrow & (b \bullet a) \bullet c & \rightsquigarrow & b \bullet (a \bullet c) \end{array}$$

Réduction

Pour les monoïdes, on va se donner également des règles de réduction

$$e \bullet x \rightsquigarrow x \qquad x \bullet e \rightsquigarrow x \qquad (x \bullet y) \bullet z \rightsquigarrow x \bullet (y \bullet z)$$

On s'aperçoit que tout terme peut se réduire vers un terme de la forme

$$x_1 \bullet (x_2 \bullet \cdots \bullet (x_{n-1} \bullet x_n) \cdots)$$

qui peut se voir comme une **liste**

$$[x_1;\ x_2;\ \cdots;\ x_{n-1};\ x_n]$$

Réduction

Pour les monoïdes, on va se donner également des règles de réduction

$$e \bullet x \rightsquigarrow x \qquad x \bullet e \rightsquigarrow x \qquad (x \bullet y) \bullet z \rightsquigarrow x \bullet (y \bullet z)$$

On s'aperçoit que tout terme peut se réduire vers un terme de la forme

$$x_1 \bullet (x_2 \bullet \cdots \bullet (x_{n-1} \bullet x_n) \cdots)$$

... et cela s'implémente très bien

```
type s = A | B | C
type monS = s list

let exemple = [C; A; B]
```

Réduction

Pour les monoïdes, on va se donner également des règles de réduction

$$e \bullet x \rightsquigarrow x \qquad x \bullet e \rightsquigarrow x \qquad (x \bullet y) \bullet z \rightsquigarrow x \bullet (y \bullet z)$$

On s'aperçoit que tout terme peut se réduire vers un terme de la forme

$$x_1 \bullet (x_2 \bullet \cdots \bullet (x_{n-1} \bullet x_n) \cdots)$$

... et cela s'implémente très bien

```
type s = A | B | C
type monS = s list

let exemple = [C; A; B]
```

Meilleure représentation que les termes car **unique** !

Catégories

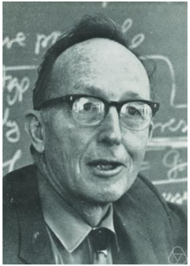
On ne compose pas toujours comme on veut

$$M = \begin{pmatrix} 2 & 0 \\ 1 & 3 \\ 0 & 1 \\ 2 & 2 \end{pmatrix} \qquad N = \begin{pmatrix} 3 & 0 & 1 \\ 4 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

$$N \circ M = \texttt{undefined}$$

Catégories

- introduction des **catégories** par Eilenberg et MacLane dans les années 40



Catégories

- introduction des **catégories** par Eilenberg et MacLane dans les années 40
- structure simple : des objets et des flèches composables entre ces objets

$$x \xrightarrow{f} y \xrightarrow{g} z, \qquad x \xrightarrow{g \circ f} z$$

Exemple avec les matrices

$$M = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 3 \end{pmatrix} \qquad N = \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix} \qquad \rightsquigarrow \qquad 3 \xrightarrow{M} 2 \xrightarrow{N} 2, \quad 3 \xrightarrow{N \circ M} 2$$

Catégories

- introduction des **catégories** par Eilenberg et MacLane dans les années 40
- structure simple : des objets et des flèches composables entre ces objets

$$x \xrightarrow{f} y \xrightarrow{g} z \qquad x \xrightarrow{g \circ f} z$$

Exemple avec les matrices

$$M = \begin{pmatrix} 2 & 0 \\ 1 & 3 \\ 0 & 1 \\ 2 & 2 \end{pmatrix} \qquad N = \begin{pmatrix} 3 & 0 & 1 \\ 4 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \qquad \rightsquigarrow \qquad 2 \xrightarrow{M} 4 \qquad 3 \xrightarrow{N} 3$$

Catégories

- introduction des **catégories** par Eilenberg et MacLane dans les années 40
- structure simple : des objets et des flèches composables entre ces objets

$$x \xrightarrow{f} y \xrightarrow{g} z \qquad x \xrightarrow{g \circ f} z$$

- utilisées ailleurs : algèbre, théorie de la représentation, logique, sémantique, *etc.*
  - catégorie des ensembles
  - catégorie des groupes
  - catégorie des anneaux
  - catégorie syntaxique
  - *etc.*

Mon travail

Développer des outils informatiques pour des généralisations des catégories en dimensions supérieures.

Back to English...

Outline

Higher categories

Word problem on strict categories

Pasting diagrams

Coherence for Gray categories

Outline

Higher categories

Word problem on strict categories

Pasting diagrams

Coherence for Gray categories

Higher categories: why?

Example in topology

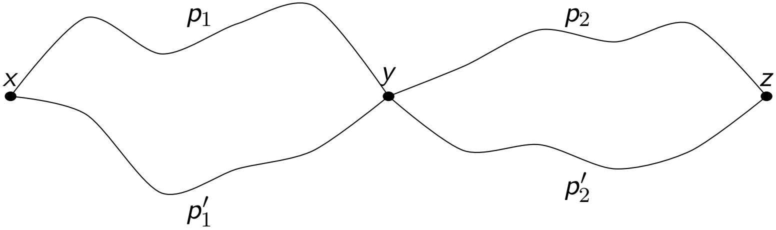
Higher categories: why?

Example in topology

- points

Higher categories: why?

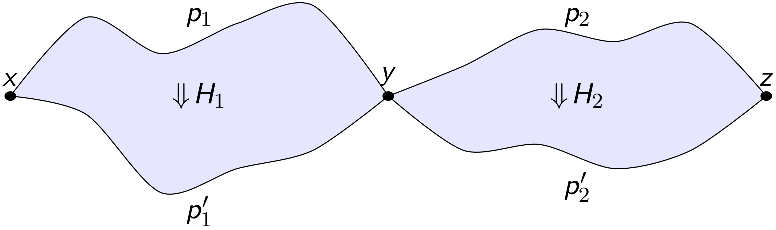
Example in topology



- points
- paths between points

Higher categories: why?

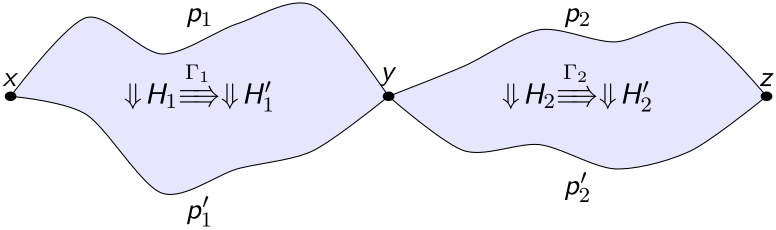
Example in topology



- points
- paths between points
- homotopies between paths

Higher categories: why?

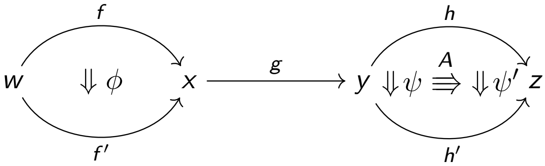
Example in topology



- points
- paths between points
- homotopies between paths
- homotopies between homotopies

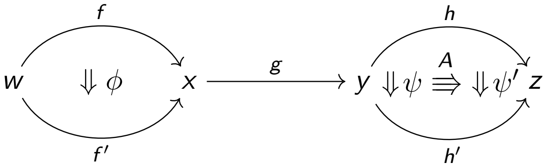
Higher categories: why?

Higher categories: categories with higher cells

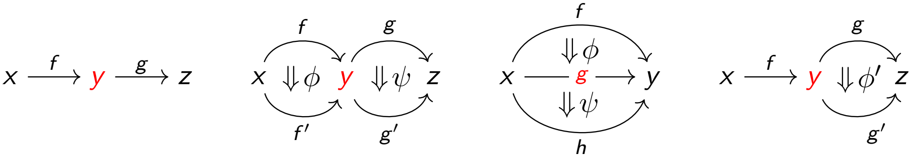


Higher categories: why?

Higher categories: categories with higher cells

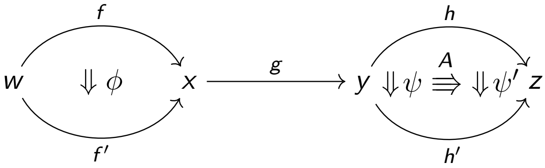


Cells can be combined with different operations



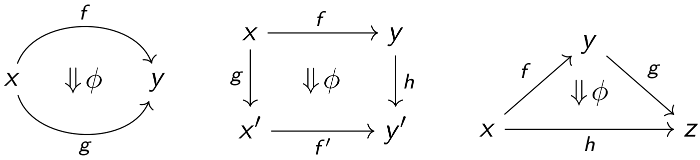
Higher categories: why?

Higher categories: categories with higher cells



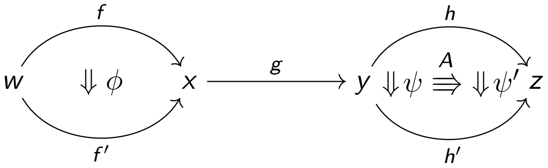
Different possible flavors

- with regard the shape of the cells: globular, cubical, simplicial, *etc*.



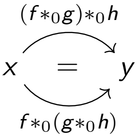
Higher categories: why?

Higher categories: categories with higher cells

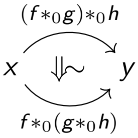


Different possible flavors

- with regard the shape of the cells: globular, cubical, simplicial, *etc.*
- with regard to the axioms enforced, **strict** or **weak**

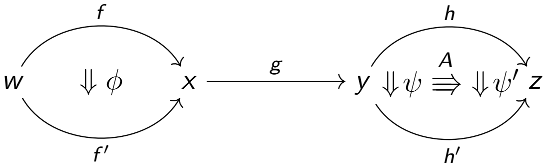


vs



Higher categories: why?

Higher categories: categories with higher cells



Different possible flavors

- with regard the shape of the cells: globular, cubical, simplicial, *etc.*
- with regard to the axioms enforced, **strict** or **weak**

For example in dimension 3:

- strict 3-categories
- tricategories
- Gray categories

A common perspective

How to consider all the higher categories in a unified way?

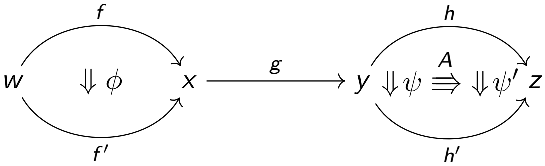
Definition (Batanin,98)

A higher category is a T-algebra for a monad T on globular sets.

Globular sets

***n*-globular sets**: graphs in higher dimensions with cells up to dimension n

Example of a 3-globular set X

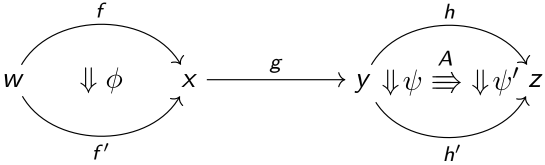


$$X_0 = \{w, x, y, z\}$$
$$X_1 = \{f, f', g, h, h'\}$$
$$X_2 = \{\phi, \psi, \psi'\}$$
$$X_3 = \{A\}$$

Globular sets

***n*-globular sets**: graphs in higher dimensions with cells up to dimension n

Example of a 3-globular set X



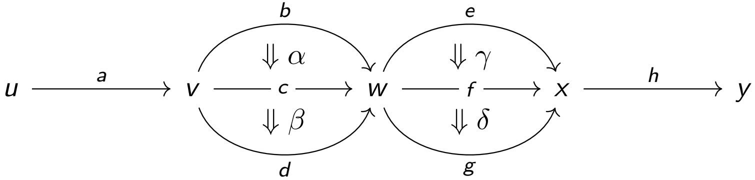
$$\begin{aligned}
X_0 &= \{w, x, y, z\} \\
X_1 &= \{f, f'\colon w \to x, \quad g\colon x \to y, \quad h, h'\colon y \to z\} \\
X_2 &= \{\phi\colon f \Rightarrow f', \quad \psi, \psi'\colon h \Rightarrow h'\} \\
X_3 &= \{A\colon \psi \Rrightarrow \psi'\}
\end{aligned}$$

General form: $X_0 \underset{\partial_0^+}{\overset{\partial_0^-}{\leftleftarrows}} X_1 \underset{\partial_1^+}{\overset{\partial_1^-}{\leftleftarrows}} X_2 \underset{\partial_2^+}{\overset{\partial_2^-}{\leftleftarrows}} \cdots \underset{\partial_{n-1}^+}{\overset{\partial_{n-1}^-}{\leftleftarrows}} X_n$

Free higher categories

We saw that free monoids could be generated on sets

Similarly, we can generate free categories on globular sets

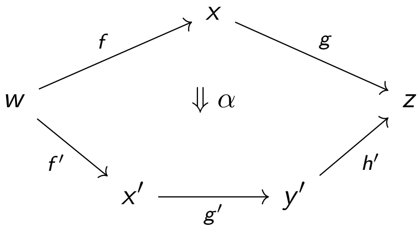


$$a *_0 b, \qquad d *_0 (e *_0 h), \qquad \alpha *_1 \beta, \qquad (\alpha *_0 \gamma) *_1 (\beta *_0 \delta), \qquad \textit{etc.}$$

Free higher categories

We saw that free monoids could be generated on sets

Similarly, we can generate free categories on ~~globular sets~~

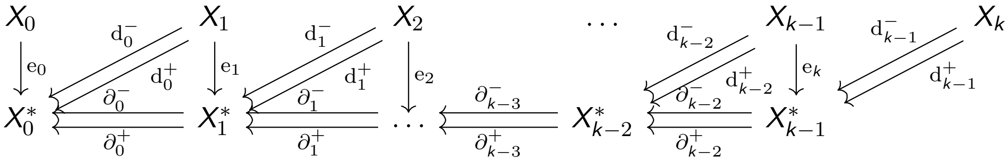


Polygraphs

$$X_0 \underset{\partial_0^+}{\overset{\partial_0^-}{\longleftleftarrows}} X_1 \underset{\partial_1^+}{\overset{\partial_1^-}{\longleftleftarrows}} \cdots \underset{\partial_{k-3}^+}{\overset{\partial_{k-3}^-}{\longleftleftarrows}} X_{k-2} \underset{\partial_{k-2}^+}{\overset{\partial_{k-2}^-}{\longleftleftarrows}} X_{k-1} \underset{\partial_{k-1}^+}{\overset{\partial_{k-1}^-}{\longleftleftarrows}} X_k$$

We need a more complex structure than globular sets

Polygraphs



We need a more complex structure than globular sets: **polygraphs** [Street, Burroni]

Batanin revisited

Theorem (Batanin)

Polygraphs and free categories on polygraphs are well-defined for globular algebras.

Batanin revisited

Theorem (Batanin, F.)

Polygraphs and free categories on polygraphs are well-defined for globular algebras.

Another proof which highlights some interesting intermediate constructions.

Abstract criterion

Globular monads are not readily available in general. Can we get rid of them?

Abstract criterion

Globular monads are not readily available in general. Can we get rid of them?

Theorem

Given functors

$$\mathcal{T}\colon C \to C' \qquad U\colon C \to \mathbf{Glob}_n \qquad U'\colon C' \to \mathbf{Glob}_n$$

such that (...), there exists equivalence of categories

$$H\colon C \to \mathbf{Glob}_n \qquad H'\colon C' \to \mathbf{Glob}_k$$

such that the following diagram commutes

$$\begin{array}{ccc} C & \xrightarrow{H} & \mathbf{Alg}_n \\ {\scriptstyle \mathcal{T}}\downarrow & & \downarrow{\scriptstyle (-)^{\mathrm{Alg}}_{\leq k}} \\ C' & \xrightarrow[H']{} & \mathbf{Alg}_k \end{array}$$

The word problem on strict categories

We can easily solve the word problem for the theory of monoids.

$$(b \bullet a) \bullet (e \bullet c) = ((e \bullet b) \bullet a) \bullet c \quad ?$$

$$(b \bullet a) \bullet (e \bullet c) \quad \rightsquigarrow \quad (b \bullet a) \bullet c \quad \rightsquigarrow \quad b \bullet (a \bullet c)$$

$$((e \bullet b) \bullet a) \bullet c \quad \rightsquigarrow \quad (b \bullet a) \bullet c \quad \rightsquigarrow \quad b \bullet (a \bullet c)$$

The word problem on strict categories

We can easily solve the word problem for the theory of monoids.

Higher strict categories are generalized monoids. Word problem for them?

The word problem on strict categories

We can easily solve the word problem for the theory of monoids.

Higher strict categories are generalized monoids. Word problem for them?

Example: 2-polygraph P with

$$\mathsf{P}_0 = \{*\} \qquad \mathsf{P}_1 = \{a\colon * \to *\} \qquad \mathsf{P}_2 = \{\alpha\colon a \to a\}$$

Word problem instances

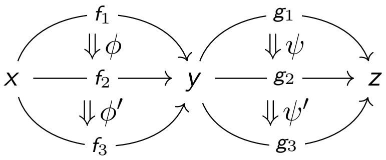
- $\alpha = \alpha *_0 (\alpha *_0 \alpha)$? no
- $(\alpha *_0 \mathrm{id}_a) *_1 (\mathrm{id}_a *_0 \alpha) = \alpha *_0 \alpha$? yes

Solution based on rewriting?

The word problem for the theory of monoids was solved using **rewriting**

We only had to find a suitable **orientation** of the axioms

We cannot orient the **exchange law** of strict categories adequately



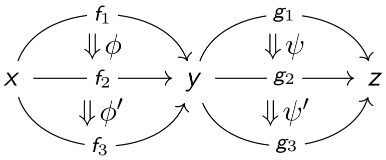
$$(\phi *_1 \phi') *_0 (\psi *_1 \psi') = (\phi *_0 \psi) *_1 (\phi' *_0 \psi')$$

Solution based on rewriting?

The word problem for the theory of monoids was solved using **rewriting**

We only had to find a suitable **orientation** of the axioms

We cannot orient the **exchange law** of strict categories adequately



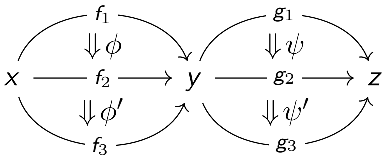
$$(\phi *_1 \phi') *_0 (\psi *_1 \psi') \to (\phi *_0 \psi) *_1 (\phi' *_0 \psi') \rightsquigarrow \text{no}$$

Solution based on rewriting?

The word problem for the theory of monoids was solved using **rewriting**

We only had to find a suitable **orientation** of the axioms

We cannot orient the **exchange law** of strict categories adequately



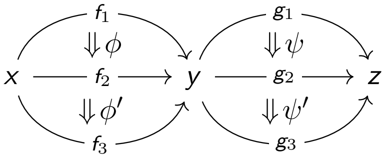
$$(\phi *_1 \phi') *_0 (\psi *_1 \psi') \leftarrow (\phi *_0 \psi) *_1 (\phi' *_0 \psi') \rightsquigarrow \text{no}$$

Solution based on rewriting?

The word problem for the theory of monoids was solved using **rewriting**

We only had to find a suitable **orientation** of the axioms

We cannot orient the **exchange law** of strict categories adequately



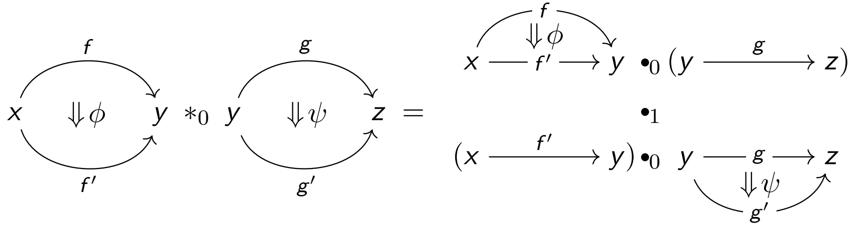
$$(\phi *_1 \phi') *_0 (\psi *_1 \psi') = (\phi *_0 \psi) *_1 (\phi' *_0 \psi')$$

$\rightsquigarrow$ no easy and efficient solution based on rewriting

Other operations

A solution to this problem was nevertheless found by Makkai.

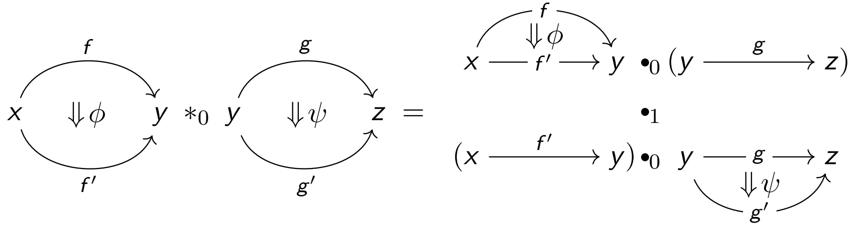
Simplification: introduce other operations for defining strict categories



Other operations

A solution to this problem was nevertheless found by Makkai.

Simplification: introduce other operations for defining strict categories



***n*-precategory**: an *n*-category with composition operations of the form

$$\bullet_{k,l} \colon C_k \times_{\min(k,l)-1} C_l \to C_{\max(k,l)} \text{ for } 0 < k, l \leq n$$

satisfying conditions.

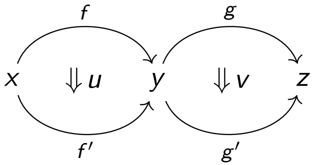
Other operations

Theorem (Makkai,05)

There is an isomorphism of categories between $\mathbf{Cat}_n$ *and* $\mathbf{PCat}_n^{(\mathrm{E})}$.

Definition (The exchange condition (E))

Given

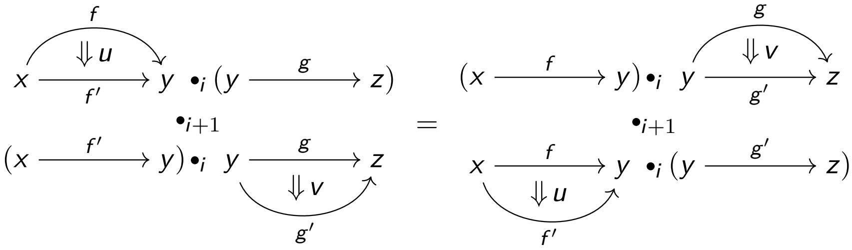


Other operations

Theorem (Makkai,05)

There is an isomorphism of categories between $\mathbf{Cat}_n$ *and* $\mathbf{PCat}_n^{(\mathrm{E})}$.

Definition (The exchange condition (E))

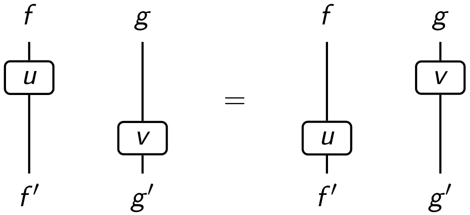


Other operations

Theorem (Makkai,05)

There is an isomorphism of categories between $\mathbf{Cat}_n$ *and* $\mathbf{PCat}_n^{(\mathrm{E})}$.

Definition (The exchange condition (E))



Canonical forms

It is now possible to orient the axioms of precategories to recover normal forms

Canonical forms

It is now possible to orient the axioms of precategories to recover normal forms

Example for a 3-polygraph P: given a 3-cell $u \in \mathsf{P}^*$ with one maximal generator $\in \mathsf{P}_3$

$$u = l_2 \bullet_1 (l_1 \bullet_0 g \bullet_0 r_1) \bullet_1 r_2$$

for some $g \in \mathsf{P}_3$, $l_1, r_1 \in \mathsf{P}_1^*$ and $l_2, r_2 \in \mathsf{P}_2^*$

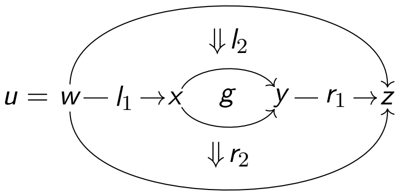
Canonical forms

It is now possible to orient the axioms of precategories to recover normal forms

Example for a 3-polygraph P: given a 3-cell $u \in \mathrm{P}^*$ with one maximal generator $\in \mathrm{P}_3$

$$u = l_2 \bullet_1 (l_1 \bullet_0 g \bullet_0 r_1) \bullet_1 r_2$$

for some $g \in \mathrm{P}_3$, $l_1, r_1 \in \mathrm{P}_1^*$ and $l_2, r_2 \in \mathrm{P}_2^*$ $\rightsquigarrow$ u is a **whiskered generator**



Canonical forms

It is now possible to orient the axioms of precategories to recover normal forms

Example for a 3-polygraph P: given a general 3-cell $u \in \mathsf{P}^*$

$$u = w_1 \bullet_2 \cdots \bullet_2 w_k$$

where the w_i's are whiskered generators.

Canonical forms

It is now possible to orient the axioms of precategories to recover normal forms

Example for a 3-polygraph P: given a general 3-cell $u \in \mathrm{P}^*$

$$u = w_1 \bullet_2 \cdots \bullet_2 w_k$$

where the w_i's are whiskered generators.

For precategories: **unique** such normal forms for cells.

Canonical forms

It is now possible to orient the axioms of precategories to recover normal forms

Example for a 3-polygraph P: given a general 3-cell $u \in \mathsf{P}^*$

$$u = w_1 \bullet_2 \cdots \bullet_2 w_k$$

where the w_i's are whiskered generators.

For precategories: **unique** such normal forms for cells.

For strict categories: several possible normal forms by (E): **canonical forms**

Makkai's algorithm

Problem: Given an n-polygraph P of **strict categories** and two words w_1 and w_2 on P,

how to decide whether w_1 and w_2 represent the same cell of P^*?

Solution:

- compute canonical forms w_1', w_2' for w_1 and w_2,
- compute the set S of canonical forms that are equivalent to w_1' w.r.t. (E),
- check whether $w_2' \in S$.

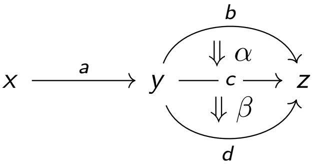
My work

What I did on this subject

- a **computability formalism** for higher categories
- some **enhancements** on Makkai's procedure
- an **implementation** of the solution in OCaml: `cateq`

cateq

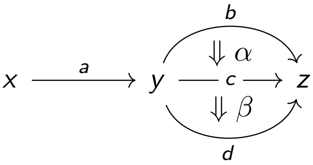
A brief overview of cateq



$$\mathrm{id}_a *_0 (\alpha *_1 \beta) = (\mathrm{id}_a *_0 \alpha) *_1 (\mathrm{id}_a *_0 \beta) \ ?$$

cateq

A brief overview of cateq

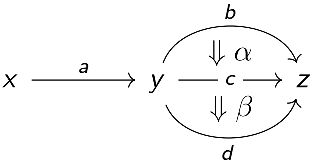


$$\mathrm{id}_a *_0 (\alpha *_1 \beta) = (\mathrm{id}_a *_0 \alpha) *_1 (\mathrm{id}_a *_0 \beta) \ ?$$

```
# x,y,z := gen *
# a := gen x -> y
# b,c,d := gen y -> z
# alpha := gen b -> c
# beta := gen c -> d
```

cateq

A brief overview of cateq

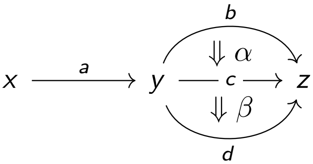


$$\mathrm{id}_a *_0 (\alpha *_1 \beta) = (\mathrm{id}_a *_0 \alpha) *_1 (\mathrm{id}_a *_0 \beta) \ ?$$

```
# lhs := id2 a *0 (alpha *1 beta)
# rhs := (id2 a *0 alpha) *1 (id2 a *0 beta)
```

cateq

A brief overview of `cateq`



$$\mathrm{id}_a *_0 (\alpha *_1 \beta) = (\mathrm{id}_a *_0 \alpha) *_1 (\mathrm{id}_a *_0 \beta) \text{ ?}$$

```
# lhs = rhs
```

and `cateq` answers `true`.

Problem

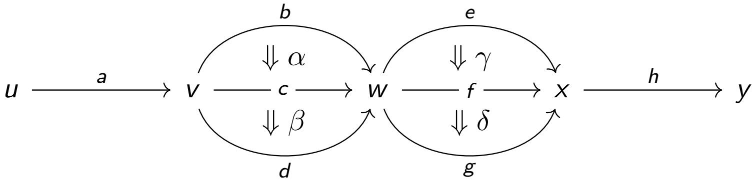
strict categories: the word problem is solved and the solver improved, **but**

- solving can still be expensive (worst case is at least factorial)
- formal expressions (words) remain non-intuitive

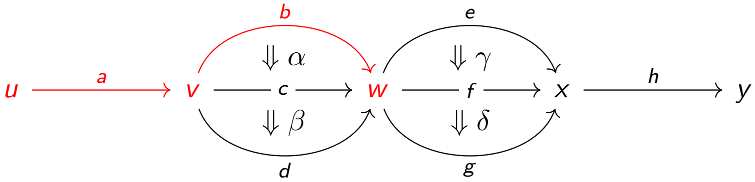
$$\mathrm{id}^2_a *_0 (\alpha *_1 \beta) *_0 ((\gamma *_0 \mathrm{id}^2_h) *_1 (\delta *_0 \mathrm{id}^2_h))???$$
$$(\mathrm{id}^2_a *_0 \alpha *_0 \mathrm{id}^2_e *_0 \mathrm{id}^2_h) *_1 (\mathrm{id}^2_a *_0 \mathrm{id}^2_c *_0 \gamma *_0 \mathrm{id}^2_h) *_1 (\mathrm{id}^2_a *_0 \beta *_0 \delta *_0 \mathrm{id}^2_h)???$$

Can we find a nicer representation? Even a **partial** one?

Cells as sets

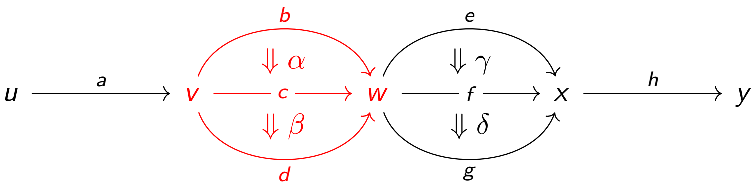


Cells as sets



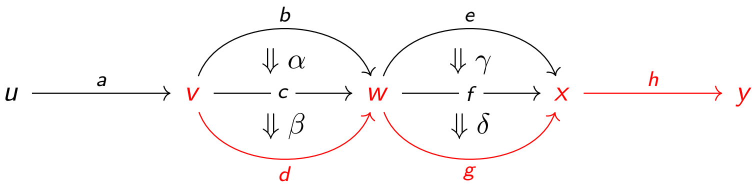
$$\{u, v, w, a, b\} \rightsquigarrow a *_0 b$$

Cells as sets



$$\{v, w, b, c, d, \alpha, \beta\} \rightsquigarrow \alpha *_1 \beta$$

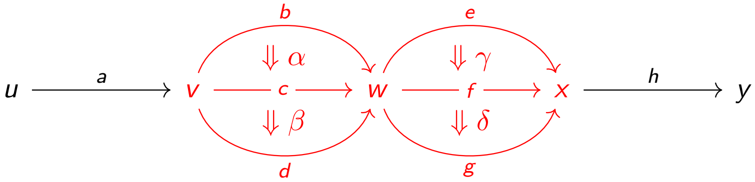
Cells as sets



Possibly several expressions but they are equivalent

$$\{v, w, x, y, d, g, h\} \rightsquigarrow (d *_0 g) *_0 h \quad \text{or} \quad d *_0 (g *_0 h)$$

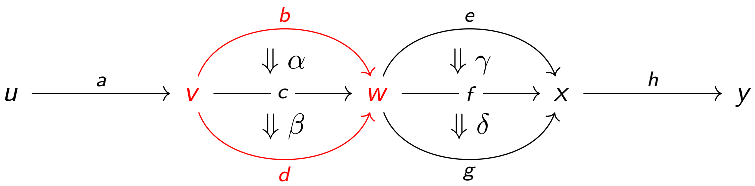
Cells as sets



Possibly several expressions but they are equivalent

$$\{v, w, x, b, c, d, e, f, g, \alpha, \beta, \gamma, \delta\} \rightsquigarrow (\alpha *_1 \beta) *_0 (\gamma *_1 \delta), \quad (\alpha *_0 \gamma) *_1 (\beta *_0 \delta), \quad \textit{etc.}$$

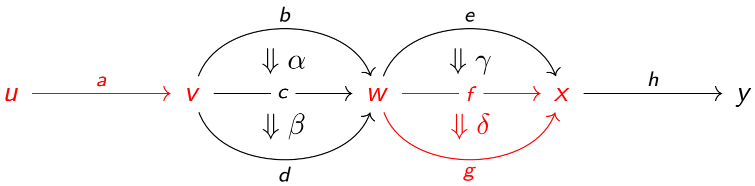
Cells as sets



Some sets make no sense

$$\{v, w, b, d\} \rightsquigarrow \texttt{undefined}$$

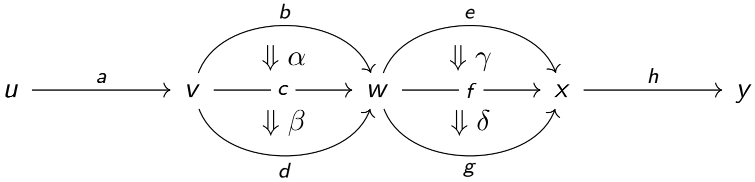
Cells as sets



Some sets make no sense

$$\{u, v, w, x, a, f, g, \delta\} \rightsquigarrow \texttt{undefined}$$

Cells as sets



Sets allows for an efficient representation of cells.

Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad \textit{etc.}$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

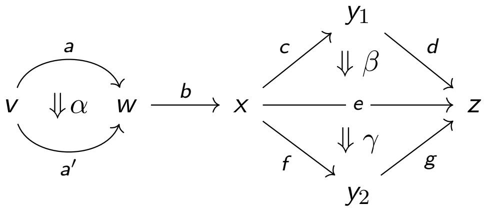
Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad \textit{etc.}$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Example



$$P_0 = \{v, w, x, y_1, y_2, z\} \qquad P_1 = \{a, a', b, c, d, e, f, g\} \qquad P_2 = \{\alpha, \beta, \gamma\}$$

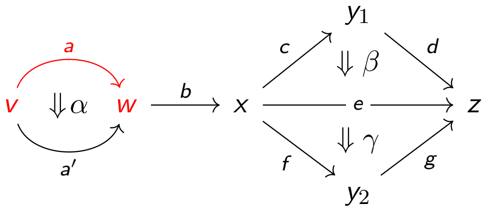
Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad etc.$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Example



$a^- = \{v\}$, $a^+ = \{w\}$

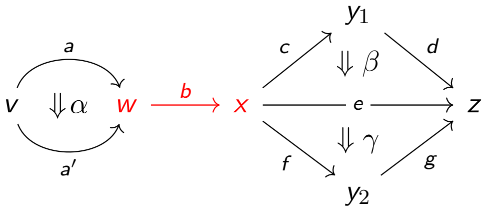
Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad etc.$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Example



$a^- = \{v\}$, $a^+ = \{w\}$, $b^- = \{w\}$, $b^+ = \{x\}$

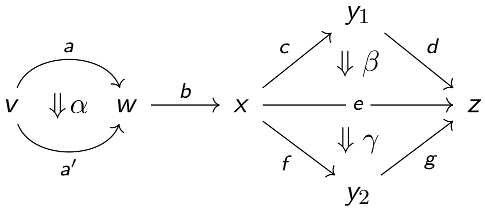
Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad etc.$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Example



$a^- = \{v\}$, $a^+ = \{w\}$, $b^- = \{w\}$, $b^+ = \{x\}$, *etc.*

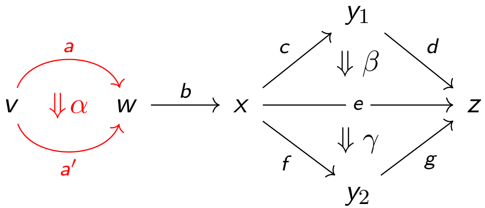
Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad \textit{etc.}$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Example



$\alpha^- = \{a\}$, $\alpha^+ = \{a'\}$

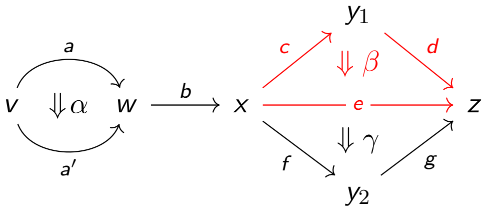
Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad etc.$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Example



$\alpha^- = \{a\}$, $\alpha^+ = \{a'\}$, $\beta^- = \{c, d\}$, $\beta^+ = \{e\}$

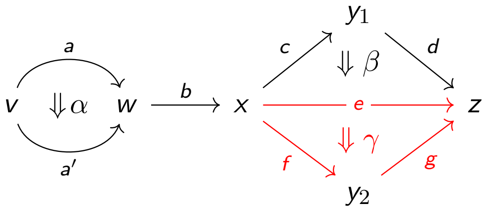
Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad etc.$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Example



$\alpha^- = \{a\}$, $\alpha^+ = \{a'\}$, $\beta^- = \{c, d\}$, $\beta^+ = \{e\}$, $\gamma^- = \{e\}$, $\gamma^+ = \{f, g\}$

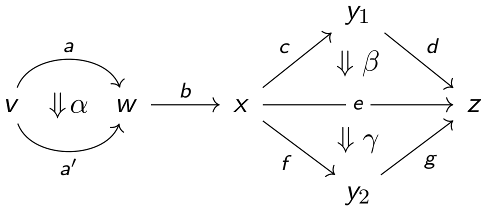
Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad \textit{etc.}$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Example



$\alpha^- = \{a\}$, $\alpha^+ = \{a'\}$, $\beta^- = \{c, d\}$, $\beta^+ = \{e\}$, $\gamma^- = \{e\}$, $\gamma^+ = \{f, g\}$, *etc.*

Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad \textit{etc.}$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

Definition

A subset of a diagram is a **pasting diagram** when its elements define a cell in an unambiguous way.

Formal diagrams

A **formal diagram** (a.k.a ω-hypergraph) is a sequence of sets

$$P_0, \qquad P_1, \qquad P_2, \qquad \textit{etc.}$$

with **source and target sets** $u^-, u^+ \subseteq P_i$ for each $u \in P_{i+1}$.

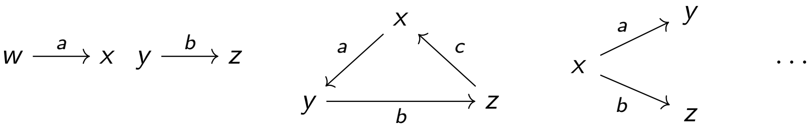
Definition

A subset of a diagram is a **pasting diagram** when its elements define a cell in an unambiguous way.

But which diagrams are pasting diagrams?

Problematic diagrams

In dimension 1, we easily see what are the “problematic diagrams”



Problematic diagrams

In dimension 1, we easily see what are the "problematic diagrams"

Easy characterization of pasting diagrams in dimension 1:

they are the finite connected linear diagrams.

$$x_0 \xrightarrow{a_1} x_1 \xrightarrow{a_2} x_2 \xrightarrow{a_3} \cdots \xrightarrow{a_n} x_n$$

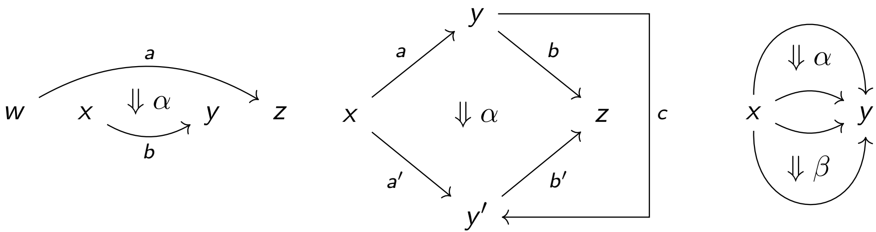
Problematic diagrams

In dimension 1, we easily see what are the "problematic diagrams"

Easy characterization of pasting diagrams in dimension 1:

they are the finite connected linear diagrams.

But the "problematic" examples are harder to characterize in higher dimensions



etc.

Pasting diagram formalisms

Several formalisms where proposed with such sets of conditions:

- Street's **parity complexes**
- Johnson's **pasting schemes**
- Steiner's **augmented directed complexes**

Pasting diagram formalisms

Several formalisms where proposed with such sets of conditions:

- Street's **parity complexes**
- Johnson's **pasting schemes**
- Steiner's **augmented directed complexes**

Theorem (Street / Johnson / Steiner)

A diagram which is a parity complex / pasting scheme / augmented directed complex is a pasting diagram.

Pasting diagram formalisms

Several formalisms where proposed with such sets of conditions:

- Street's **parity complexes**
- Johnson's **pasting schemes**
- Steiner's **augmented directed complexes**

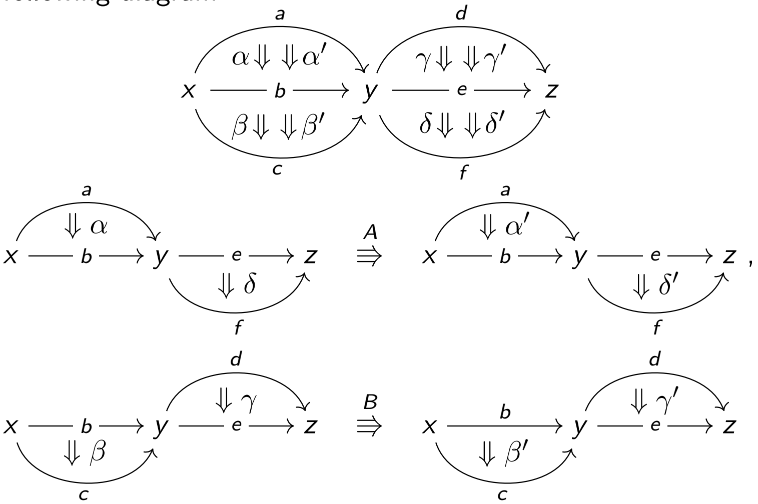
Theorem (Street / Johnson / Steiner)

A diagram which is a parity complex / pasting scheme / augmented directed complex is a pasting diagram.

Actually, some condition is missing for parity complexes and pasting schemes...

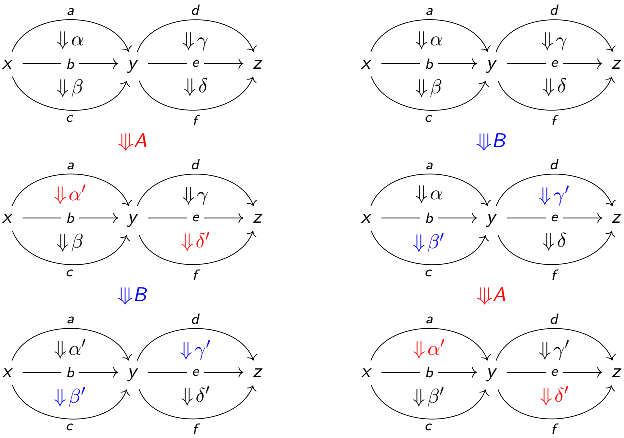
Counter-example

Consider the following diagram



The counter-example

There are two composites of all the generators of P.



The counter-example

There are two composites of all the generators of P.

Formally, they correspond to two 3-cells H_1 and H_2

$$H_1 = ((a *_0 \gamma) *_1 A *_1 (\beta *_0 f)) *_2 ((\alpha' *_0 d) *_1 B *_1 (c *_0 \delta'))$$
$$H_2 = ((\alpha *_0 d) *_1 B *_1 (c *_0 \delta)) *_2 ((a *_0 \gamma') *_1 A *_1 (\beta' *_0 f))$$

both associated to the same set

$$\{x, y, z, a, b, c, d, e, f, \alpha, \alpha', \beta, \beta', \gamma, \gamma', \delta, \delta', A, B\}$$

The counter-example

There are two composites of all the generators of P.

Formally, they correspond to two 3-cells H_1 and H_2 .

But $\boldsymbol{H_1 \neq H_2}$, disproving the pasting diagram property:

Theorem

Given the parity complex / pasting scheme P above, its associated polygraph Q, and the evaluation function

$$\mathrm{eval}\colon \mathsf{Q}^* \to \mathcal{P}(P)$$

from formal cells to subsets of P, we have

$$H_1 \neq H_2 \in \mathsf{Q}^* \qquad \textit{and} \qquad \mathrm{eval}(H_1) = \mathrm{eval}(H_2) = P$$

*so eval is **not** an isomorphism.*

The counter-example

There are two composites of all the generators of P.

Formally, they correspond to two 3-cells H_1 and H_2

In order to prove the inequality $H_1 \neq H_2$, cateq can be used

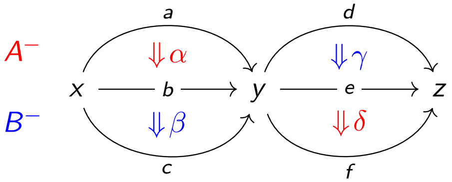
```
# H1 := ((id3 a *0 id3 gamma) *1 A *1 (id3 beta *0 id3 f))
    *2 ((id3 alpha' *0 id3 d) *1 B *1 (id3 c *0 id3 delta'))
# H2 := ((id3 alpha *0 id3 d) *1 B *1 (id3 c *0 id3 delta))
    *2 ((id3 a *0 id3 gamma') *1 A *1 (id3 beta' *0 id3 f))
# H1 = H2
```

to which cateq answers

```
false
```

Fix

Consider again the counter-example



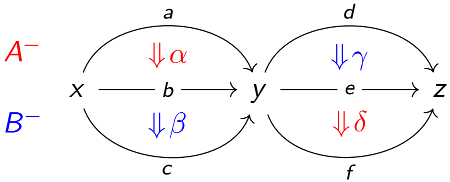
$$A\colon \{\alpha, \delta\} \Rrightarrow \{\alpha', \delta'\}$$

$$B\colon \{\beta, \gamma\} \Rrightarrow \{\beta', \gamma'\}$$

The elements of A^- are before and after the elements of B^-: they are **in torsion**.

Fix

Consider again the counter-example



$$A\colon \{\alpha, \delta\} \Rrightarrow \{\alpha', \delta'\}$$

$$B\colon \{\beta, \gamma\} \Rrightarrow \{\beta', \gamma'\}$$

Definition (Torsion-freeness)

A diagram is **torsion-free** when there are no pair of generators in torsion.

Torsion-free complexes

Theorem (Pasting diagram property)

A diagram which is a torsion-free complex is a pasting diagram.

Theorem (Embeddings)

Parity complexes and pasting schemes (satisfying torsion-freeness) and augmented directed complexes are embedded in torsion-free complexes.

Implementation of a library in OCaml which extends `cateq`

```
# A := gen {alpha, delta} -> {alpha', delta'}
# B := gen {beta,  gamma} -> {beta',  gamma'}
```

Higher categories

Word problem on strict categories

Pasting diagrams

Coherence for Gray categories

Coherence

Coherence in higher categories:

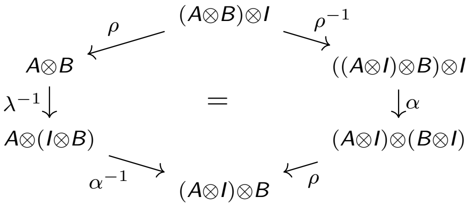
all parallel cells are equal.

Coherence

Coherence in higher categories:

all parallel cells are equal.

Classical example: **MacLane's coherence theorem** for monoidal categories.



Theorem (MacLane's coherence property for monoidal categories)

*All morphisms made of λ, ρ, α **and their inverses** between two objects are equal.*

Coherence tiles

Coherence tiles: the axioms allowing the coherence property

$$
\begin{array}{ccccc}
((A \otimes B) \otimes C) \otimes D & \xrightarrow{\alpha} & (A \otimes (B \otimes C)) \otimes D & \xrightarrow{\alpha} & A \otimes ((B \otimes C) \otimes D) \\
\alpha \downarrow & & = & & \downarrow \alpha \\
(A \otimes B) \otimes (C \otimes D) & & \xrightarrow[\alpha]{\hspace{6em}} & & A \otimes (B \otimes (C \otimes D))
\end{array}
$$

$$
\begin{array}{ccc}
(A \otimes I) \otimes B & \xrightarrow{\alpha} & A \otimes (I \otimes B) \\
& \searrow^{\lambda} \quad = \quad {}^{\rho}\swarrow & \\
& A \otimes B &
\end{array}
$$

Coherence tiles

Coherence tiles: the axioms allowing the coherence property

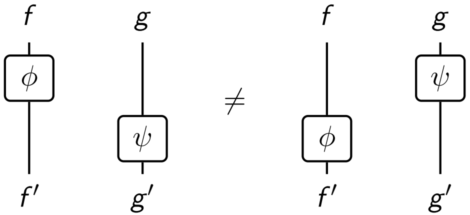
$$\begin{array}{ccccc} ((w\bullet x)\bullet y)\bullet z & \xrightarrow{\alpha} & (w\bullet(x\bullet y))\bullet z & \xrightarrow{\alpha} & w\bullet((x\bullet y)\bullet z) \\ {\scriptstyle\alpha}\downarrow & & & & \downarrow{\scriptstyle\alpha} \\ (w\bullet x)\bullet(y\bullet z) & & \xrightarrow[\alpha]{\qquad\qquad} & & w\bullet(x\bullet(y\bullet z)) \end{array}$$

$$\begin{array}{ccc} (w\bullet e)\bullet x & \xrightarrow{\alpha} & w\bullet(e\bullet x) \\ & = & \\ {\scriptstyle\lambda}\searrow & & \swarrow{\scriptstyle\rho} \\ & w\bullet x & \end{array}$$

Observation [Street,Burroni,Guiraud,Malbos]: coherence tiles can be found using rewriting theory.

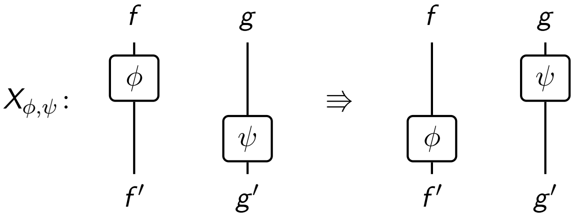
Gray categories

This work: adaptation to **Gray categories** (3-dimensional categories)



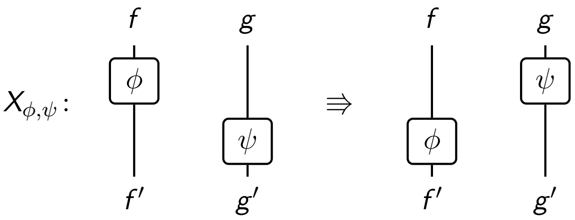
Gray categories

This work: adaptation to **Gray categories** (3-dimensional categories)



Gray categories

This work: adaptation to **Gray categories** (3-dimensional categories)



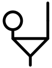
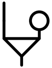
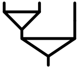
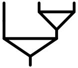
Theorem (Mimram, F.)

The coherence tiles for Gray categories can be computed using with rewriting systems.

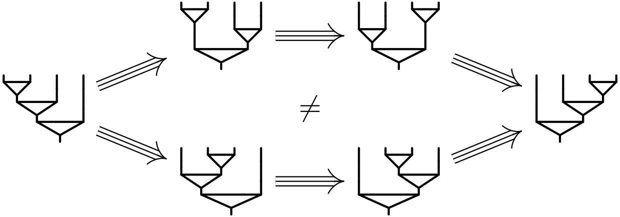
Example

3-polygraph P for pseudomonoids

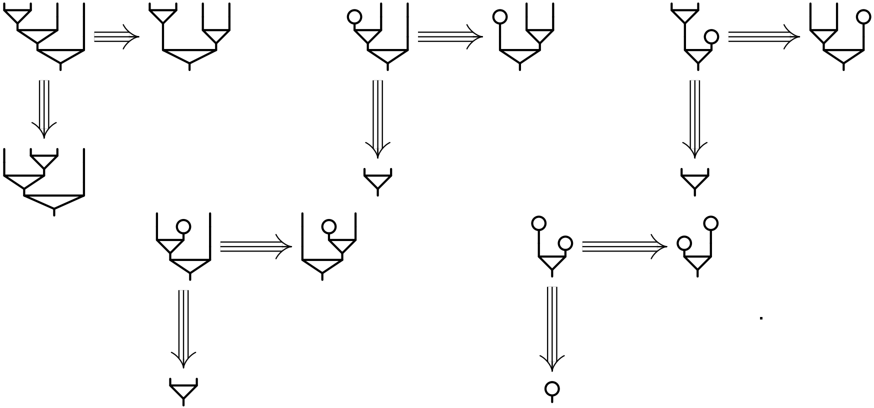
$\mathsf{P}_2 = \{$ $: \bar{0} \Rightarrow 1,$ $: \bar{2} \Rightarrow \bar{1}$ $\}$

$\mathsf{P}_3 = \{$ $L:$  $\Rrightarrow$, $R:$  $\Rrightarrow$, $A:$  $\Rrightarrow$  $\}$

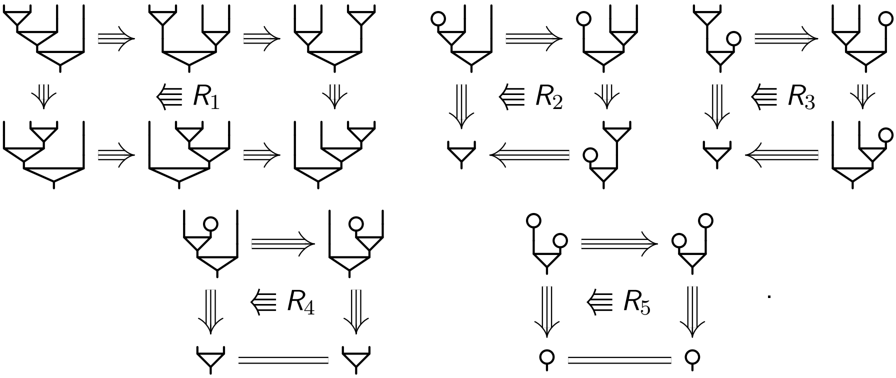
As is, it induces a $(3,2)$-Gray category which is **not** coherent.



Example



Example



Theorem

$R_1, \ldots, R_5$ are adequate coherence tiles for pseudomonoids.

The end

Thank you!