

Categorical coherence from term rewriting systems

Samuel Mimram

CoREACT meeting / 19 March 2023

École polytechnique

The coherence theorem for monoidal categories

A monoidal category $(\mathcal{C}, \otimes, \mathbf{e}, \alpha, \lambda, \rho)$ comes equipped with

$$\alpha_{x,y,z} : (x \otimes y) \otimes z \xrightarrow{\sim} x \otimes (y \otimes z) \quad \lambda_x : \mathbf{e} \otimes x \xrightarrow{\sim} x \quad \rho_x : x \otimes \mathbf{e} \xrightarrow{\sim} x$$

satisfying axioms.

$$\begin{array}{ccccc} ((x \otimes y) \otimes z) \otimes w & \longrightarrow & (x \otimes (y \otimes z)) \otimes w & \longrightarrow & x \otimes ((y \otimes z) \otimes w) \\ \downarrow & & & & \downarrow \\ (x \otimes y) \otimes (z \otimes w) & \longrightarrow & & \longrightarrow & x \otimes (y \otimes (z \otimes w)) \end{array}$$

$$\begin{array}{ccc} (x \otimes \mathbf{e}) \otimes y & \longrightarrow & x \otimes (\mathbf{e} \otimes y) \\ & \searrow & \swarrow \\ & x \otimes y & \end{array}$$

The coherence theorem for monoidal categories

A **monoidal category** $(\mathcal{C}, \otimes, e, \alpha, \lambda, \rho)$ comes equipped with

$$\alpha_{x,y,z} : (x \otimes y) \otimes z \xrightarrow{\sim} x \otimes (y \otimes z) \quad \lambda_x : e \otimes x \xrightarrow{\sim} x \quad \rho_x : x \otimes e \xrightarrow{\sim} x$$

satisfying axioms.

The **coherence theorem** for monoidal categories states that every diagram whose morphisms are composites of α , λ and ρ commutes:

$$\begin{array}{ccc} (e \otimes x) \otimes y & \longrightarrow & e \otimes (x \otimes y) \\ \downarrow & & \searrow \\ & & e \otimes (x \otimes (y \otimes e)) \\ & & \downarrow \\ x \otimes y & \longrightarrow & x \otimes (y \otimes e) \end{array}$$

The coherence theorems for monoidal categories

In fact, there are various ways of formulating the coherence theorem:

1. Coherence:

every diagram in a free monoidal category made up of α , λ and ρ commutes.

2. Coherence:

every diagram in a monoidal category made up of α , λ and ρ commutes.

3. Strictification:

every monoidal category is monoidally equivalent to a strict monoidal category.

4. Global strictification:

the forgetful 2-functor from strict monoidal categories to monoidal categories has a left adjoint and the components of the unit are equivalences.

The coherence theorems for symmetric monoidal categories

A monoidal category is **symmetric** when equipped with

$$\gamma_{x,y} : x \otimes y \rightarrow y \otimes x$$

satisfying axioms.

Similar coherence theorems hold but they are more subtle:

- in 2. we have to restrict to “generic” diagrams, e.g. the following diagram does not commute:

$$\begin{array}{ccc} & \xrightarrow{\gamma_{x,x}} & \\ x \otimes x & & x \otimes x \\ & \xrightarrow{\text{id}_{x \otimes x}} & \end{array}$$

- in 4., for a strict symmetric monoidal category, we suppose that α , λ and ρ are strict but not γ
- (global) strictification is only shown for free categories

A generic framework for coherence

Here, we investigate general coherence theorems where

- coherence holds with respect to part of the structure (e.g. α , λ and ρ but not γ)
- structural morphisms can erase or duplicate variables:

$$\delta_{x,y,z} : x \otimes (y \oplus z) \rightarrow (x \otimes y) \oplus (x \otimes z)$$

- we use rewriting theory.

A generic framework for coherence

Here, we investigate general coherence theorems where

- coherence holds with respect to part of the structure (e.g. α , λ and ρ but not γ)
- structural morphisms can erase or duplicate variables:

$$\delta_{x,y,z} : x \otimes (y \oplus z) \rightarrow (x \otimes y) \oplus (x \otimes z)$$

- we use rewriting theory.

We begin by studying the situation in an abstract setting.

Part I

Abstract coherence

An abstract setting

Fix a category \mathcal{C} which we think of as describing an **algebraic structure**.

For instance, we have a theory of symmetric monoidal categories:

- the objects of \mathcal{C} are formal tensor expressions

$$\mathbf{e} \otimes ((\mathbf{x} \otimes \mathbf{e}) \otimes \mathbf{y})$$

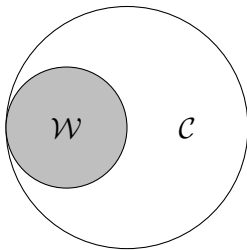
- morphisms are composites of α , λ , ρ and γ modulo axioms.

An abstract setting

Fix a category \mathcal{C} which we think of as describing an **algebraic structure**.

We suppose fixed a subgroupoid $\mathcal{W} \subseteq \mathcal{C}$ with the same objects, which we are interested in strictifying.

(for SMC, \mathcal{W} would be the groupoid of composites of α , λ and ρ , but not γ)



Quotient of categories

The **quotient** \mathcal{C}/\mathcal{W} is the universal way of making the elements of \mathcal{W} identities

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{F} & \mathcal{D} \\ \downarrow & \nearrow \tilde{F} & \\ \mathcal{C}/\mathcal{W} & & \end{array}$$

Question

When is the quotient functor $\mathcal{C} \rightarrow \mathcal{C}/\mathcal{W}$ an equivalence of categories?

Quotient of categories

The **quotient** \mathcal{C}/\mathcal{W} is the universal way of making the elements of \mathcal{W} identities

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{F} & \mathcal{D} \\ \downarrow & \nearrow \tilde{F} & \\ \mathcal{C}/\mathcal{W} & & \end{array}$$

Question

When is the quotient functor $\mathcal{C} \rightarrow \mathcal{C}/\mathcal{W}$ an equivalence of categories?

Intuitively, when \mathcal{W} does not contain non-trivial information!

Rigid groupoids

A groupoid \mathcal{W} is **rigid** when either

- (i) any two parallel morphisms $f, g : x \rightarrow y$ are equal
- (ii) any automorphism $f : x \rightarrow x$ is an identity
- (iii) \mathcal{W} is equivalent to $\bigsqcup_x 1$

Quotienting by rigid groupoids

When \mathcal{W} is rigid the quotient \mathcal{C}/\mathcal{W} has a simple description:

- objects are eq. classes of objects with $[x] = [y]$ when there is $w : x \rightarrow y$ in \mathcal{W} ,

Quotienting by rigid groupoids

When \mathcal{W} is rigid the quotient \mathcal{C}/\mathcal{W} has a simple description:

- objects are eq. classes of objects with $[x] = [y]$ when there is $w : x \rightarrow y$ in \mathcal{W} ,
- morphisms are eq. classes of morphisms with $[f] = [g]$ when there is v and w in \mathcal{W} such that

$$\begin{array}{ccc} x & \xrightarrow{f} & y \\ \textcolor{red}{\mathcal{W} \ni v} \downarrow & & \downarrow \textcolor{red}{w \in \mathcal{W}} \\ x' & \xrightarrow{g} & y' \end{array}$$

Quotienting by rigid groupoids

When \mathcal{W} is rigid the quotient \mathcal{C}/\mathcal{W} has a simple description:

- objects are eq. classes of objects with $[x] = [y]$ when there is $w : x \rightarrow y$ in \mathcal{W} ,
- morphisms are eq. classes of morphisms with $[f] = [g]$ when there is v and w in \mathcal{W} such that

$$\begin{array}{ccc} x & \xrightarrow{f} & y \\ \textcolor{red}{\mathcal{W} \ni v} \downarrow & & \downarrow \textcolor{red}{w \in \mathcal{W}} \\ x' & \xrightarrow{g} & y' \end{array}$$

- we compose $[f] : [x] \rightarrow [y]$ and $[g] : [y] \rightarrow [z]$ as

$$x \xrightarrow{f} y$$

$$y \xrightarrow{g} z$$

Quotienting by rigid groupoids

When \mathcal{W} is rigid the quotient \mathcal{C}/\mathcal{W} has a simple description:

- objects are eq. classes of objects with $[x] = [y]$ when there is $w : x \rightarrow y$ in \mathcal{W} ,
- morphisms are eq. classes of morphisms with $[f] = [g]$ when there is v and w in \mathcal{W} such that

$$\begin{array}{ccc} x & \xrightarrow{f} & y \\ \textcolor{red}{\mathcal{W} \ni v} \downarrow & & \downarrow \textcolor{red}{w \in \mathcal{W}} \\ x' & \xrightarrow{g} & y' \end{array}$$

- we compose $[f] : [x] \rightarrow [y]$ and $[g] : [y] \rightarrow [z]$ as

$$\begin{array}{ccc} x & \xrightarrow{f} & y \\ & \downarrow \textcolor{red}{\in \mathcal{W}} & \\ & y & \xrightarrow{g} z \end{array}$$

Quotienting by rigid groupoids

When \mathcal{W} is rigid the quotient \mathcal{C}/\mathcal{W} has a simple description:

- objects are eq. classes of objects with $[x] = [y]$ when there is $w : x \rightarrow y$ in \mathcal{W} ,
- morphisms are eq. classes of morphisms with $[f] = [g]$ when there is v and w in \mathcal{W} such that

$$\begin{array}{ccc} x & \xrightarrow{f} & y \\ \textcolor{red}{\mathcal{W} \ni v} \downarrow & & \downarrow \textcolor{red}{w \in \mathcal{W}} \\ x' & \xrightarrow{g} & y' \end{array}$$

- we compose $[f] : [x] \rightarrow [y]$ and $[g] : [y] \rightarrow [z]$ as

$$\begin{array}{ccc} x & \xrightarrow{f} & y \\ \textcolor{red}{\mathcal{W} \ni} \downarrow & \textcolor{red}{\downarrow \in \mathcal{W}} & \\ y & \xrightarrow{g} & z \end{array}$$

Rigidification

The **rigidification** $\mathcal{C} // \mathcal{W}$ of \mathcal{W} in \mathcal{C} is obtained from \mathcal{C} by identifying any two parallel morphisms in \mathcal{W} .

Proposition

The quotient can be obtained in two steps: $\mathcal{C} / \mathcal{W} = (\mathcal{C} // \mathcal{W}) / \mathcal{W}$

Coherence for quotients

Theorem

The quotient functor $\mathcal{C} \rightarrow \mathcal{C}/\mathcal{W}$ is an equivalence of categories if and only if \mathcal{W} is rigid.

\mathcal{C}	\mathcal{C}/\mathcal{W}
$x \xrightarrow[f]{g} y$	$x \curvearrowright f$
$x \xrightarrow[g]{f} y$	x

Coherence for quotients

Theorem

The quotient functor $\mathcal{C} \rightarrow \mathcal{C}/\mathcal{W}$ is an equivalence of categories if and only if \mathcal{W} is rigid.

Proof.

The quotient functor $\mathcal{C} \rightarrow \mathcal{C}/\mathcal{W} \rightarrow \mathcal{C}$ is surjective on objects and full.

We need to show that it is faithful iff \mathcal{W} is rigid.

- If the quotient functor is faithful, given $\mathbf{w}, \mathbf{w}' : \mathbf{x} \rightarrow \mathbf{y}$, we have $[\mathbf{w}] = [\mathbf{w}']$ and thus $\mathbf{w} = \mathbf{w}'$.
- If \mathcal{W} is rigid, given $\mathbf{f}, \mathbf{g} : \mathbf{x} \rightarrow \mathbf{y}$ such that $[\mathbf{f}] = [\mathbf{g}]$, we have

$$\begin{array}{ccc} \mathbf{x} & \xrightarrow{\mathbf{f}} & \mathbf{y} \\ \textcolor{red}{\mathcal{W} \ni \mathbf{v}} \downarrow & & \downarrow \textcolor{red}{\mathbf{w} \in \mathcal{W}} \\ \mathbf{x} & \xrightarrow{\mathbf{g}} & \mathbf{y} \end{array}$$

By rigidity, $\mathbf{v} = \text{id}_{\mathbf{x}}$ and $\mathbf{w} = \text{id}_{\mathbf{y}}$.

Coherence for algebras

An **algebra** for \mathcal{C} in \mathcal{D} is a functor $\mathcal{C} \rightarrow \mathcal{D}$, we write $\text{Alg}(\mathcal{C}, \mathcal{D})$ for the category of algebras. In particular, we are interested in $\text{Alg}(\mathcal{C}) = \text{Alg}(\mathcal{C}, \mathbf{Cat})$.

Coherence for algebras

An **algebra** for \mathcal{C} in \mathcal{D} is a functor $\mathcal{C} \rightarrow \mathcal{D}$, we write $\text{Alg}(\mathcal{C}, \mathcal{D})$ for the category of algebras. In particular, we are interested in $\text{Alg}(\mathcal{C}) = \text{Alg}(\mathcal{C}, \mathbf{Cat})$.

Theorem

A functor $F : \mathcal{C} \rightarrow \mathcal{C}'$ is an equivalence iff $\text{Alg}(F, \mathcal{D}) : \text{Alg}(\mathcal{C}, \mathcal{D}) \rightarrow \text{Alg}(\mathcal{C}', \mathcal{D})$ is an equivalence natural in \mathcal{D} .

Proof.

Given a 2-category \mathcal{K} , the Yoneda functor

$$Y_{\mathcal{K}} : \mathcal{K}^{\text{op}} \rightarrow [\mathcal{K}, \mathbf{Cat}]$$

$$\mathcal{C} \mapsto \mathcal{K}(\mathcal{C}, -)$$

is a local isomorphism. In particular, with $\mathcal{K} = \mathbf{Cat}$, we have $Y_{\mathbf{Cat}}\mathcal{C} = \text{Alg}(\mathcal{C}, -)$. \square

Coherence for algebras

An **algebra** for \mathcal{C} in \mathcal{D} is a functor $\mathcal{C} \rightarrow \mathcal{D}$, we write $\text{Alg}(\mathcal{C}, \mathcal{D})$ for the category of algebras. In particular, we are interested in $\text{Alg}(\mathcal{C}) = \text{Alg}(\mathcal{C}, \mathbf{Cat})$.

Theorem

A functor $F : \mathcal{C} \rightarrow \mathcal{C}'$ is an equivalence iff $\text{Alg}(F, \mathcal{D}) : \text{Alg}(\mathcal{C}, \mathcal{D}) \rightarrow \text{Alg}(\mathcal{C}', \mathcal{D})$ is an equivalence natural in \mathcal{D} .

Proof.

Given a 2-category \mathcal{K} , the Yoneda functor

$$Y_{\mathcal{K}} : \mathcal{K}^{\text{op}} \rightarrow [\mathcal{K}, \mathbf{Cat}]$$

$$\mathcal{C} \mapsto \mathcal{K}(\mathcal{C}, -)$$

is a local isomorphism. In particular, with $\mathcal{K} = \mathbf{Cat}$, we have $Y_{\mathbf{Cat}}\mathcal{C} = \text{Alg}(\mathcal{C}, -)$. \square

Conjecture (?)

The canonical functor $\text{Alg}(\mathcal{C}/\mathcal{W}) \rightarrow \text{Alg}(\mathcal{C})$ is an equivalence iff \mathcal{W} is rigid.

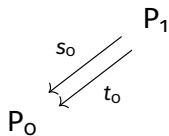
Question

How do we show rigidity in practice?

In the following, we are interested in the case where \mathcal{C} is a groupoid.

Abstract rewriting systems

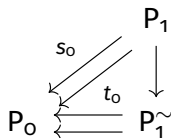
An **abstract rewriting system** P is a graph



$$P = x \begin{array}{c} \xrightarrow{f} \\ \xleftarrow{g} \end{array} y \xleftarrow{h} z$$

Abstract rewriting systems

An **abstract rewriting system** P is a graph

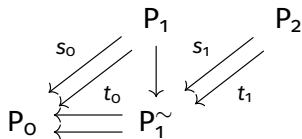


It generates a groupoid with P_1^{\sim} as set of morphisms.

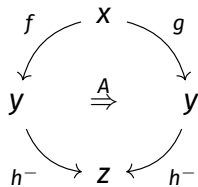
$$\begin{array}{c}
 P \quad = \quad x \begin{array}{c} \xrightarrow{f} y \\ \xleftarrow{g} y \end{array} \xleftarrow{h} z \\
 \\
 x \xrightarrow{f} y \xrightarrow{g} x \xrightarrow{f} y \xrightarrow{h^-} z
 \end{array}$$

Abstract rewriting systems

An **extended abstract rewriting system** \mathbf{P} is a graph

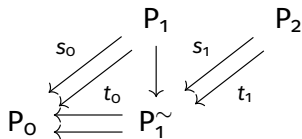


together with a set of **2-cells**

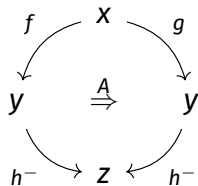


Abstract rewriting systems

An **extended abstract rewriting system** \mathbf{P} is a graph



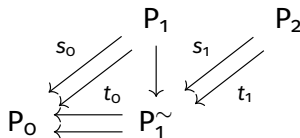
together with a set of **2-cells**



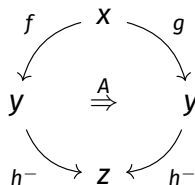
It **presents** a groupoid $\bar{\mathbf{P}} = \mathbf{P}^{\sim} / \sim$.

Abstract rewriting systems: Tietze equivalences

In a situation such as



with



- if A can be derived from other elements P_2 , we can remove it,
- if we remove $f \in P_1$ and $A \in P_2$ the presented groupoid is the same.

Abstract rewriting systems

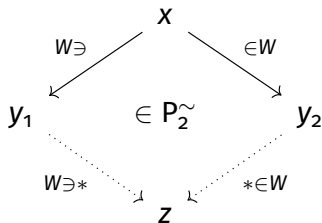
Suppose given an extended ARS \mathbf{P} together with $W \subseteq P_1$.

We say that \mathbf{P} is **W -convergent** when it has

- termination: there is no infinite sequence of morphisms in W

$$x_0 \xrightarrow{f_0} x_1 \xrightarrow{f_1} x_2 \xrightarrow{f_2} \dots$$

- local confluence:

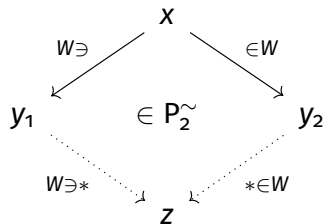


Abstract rewriting systems

By adapting standard rewriting techniques,

Lemma (“Newman”)

If P is W -convergent then it is W -confluent:



Abstract rewriting systems

By adapting standard rewriting techniques,

Lemma (“Newman”)

If P is W -convergent then it is W -confluent:

Lemma (“Church-Rosser”)

If P is W -convergent then for any two parallel W -morphisms in \bar{P} are equal.

Proof.

$$\begin{array}{ccccccccccc} x & \xrightarrow{p_1^-} & y_1 & \xrightarrow{q_1^+} & x_2 & \longrightarrow & \cdots & \longrightarrow & x_n & \xrightarrow{p_n^-} & y_n & \xrightarrow{q_n^-} & y \\ \downarrow n_x & & \downarrow n_{y_1} & & \downarrow n_{x_2} & & & & \downarrow n_{x_n} & & \downarrow n_{y_n} & & \downarrow n_y \\ \hat{x} & \equiv & \hat{x} & \equiv & \hat{x} & \equiv & \cdots & \equiv & \hat{x} & \equiv & \hat{x} & \equiv & \hat{x} \end{array}$$

□

Abstract rewriting systems

Theorem

If \mathbf{P} is \mathbf{W} -convergent then the groupoid generated by \mathbf{W} in $\overline{\mathbf{P}}$ is rigid.

Abstract rewriting systems

Theorem

If \mathbf{P} is \mathbf{W} -convergent then the groupoid generated by \mathbf{W} in $\overline{\mathbf{P}}$ is rigid.

Writing $\mathbf{N}(\overline{\mathbf{P}})$ for the full subcategory of $\overline{\mathbf{P}}$ whose objects are normal forms (are not the source of a morphism in \mathbf{W}),

Theorem

If (\mathbf{P}, \mathbf{W}) is \mathbf{W} -convergent then $\overline{\mathbf{P}}/\mathbf{W} \cong \mathbf{N}(\overline{\mathbf{P}})$.

A concrete description of normal forms

We have the intuition that the groupoid $N(\overline{P})$ is presented by the extended ars $P \setminus W$ obtained by “restricting P to normal forms”:

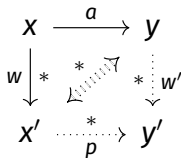
- $(P \setminus W)_0$: the objects of $P \setminus W$ are the those of P in W -normal form,
- $(P \setminus W)_1$: the rewriting rules of $P \setminus W$ are those of P whose source and target are both in $(P \setminus W)_0$ (in particular, it does not contain any element of W , thus the notation),
- $(P \setminus W)_2$: the coherence relations are those of P_2 whose source and target both belong to $(P \setminus W)_1$.

A concrete description of normal forms

Theorem

Suppose that

1. \mathbf{P} is \mathbf{W} -convergent,
2. every rule $\mathbf{a} : \mathbf{x} \rightarrow \mathbf{y}$ in \mathbf{P}_1 with \mathbf{x} is \mathbf{W} -normal also has a \mathbf{W} -normal target \mathbf{y} ,
3. for every coinital rule $\mathbf{a} : \mathbf{x} \rightarrow \mathbf{y}$ in \mathbf{P}_1 and path $\mathbf{w} : \mathbf{x} \xrightarrow{*} \mathbf{x}'$ in \mathbf{W}^* , there are paths $\mathbf{p} : \mathbf{x}' \xrightarrow{*} \mathbf{y}'$ in \mathbf{P}_1^* and $\mathbf{w}' : \mathbf{y} \xrightarrow{*} \mathbf{y}' \in \mathbf{W}^*$ such that $\mathbf{a} \cdot \mathbf{w}' \stackrel{*}{\Leftrightarrow} \mathbf{w} \cdot \mathbf{p}$:



4. for every coherence relation ...

Then $\overline{\mathbf{P}}$ is isomorphic to $\overline{\mathbf{P} \setminus \mathbf{W}}$.

Summary

Given (\mathbf{P}, \mathbf{W}) , we have shown that the following definitions of **coherence** of \mathbf{P} wrt \mathbf{W} are equivalent:

- (i) Every parallel zig-zags with edges in \mathbf{W} are equal
(i.e. the subgroupoid of $\bar{\mathbf{P}}$ generated by \mathbf{W} is rigid).
- (ii) The quotient map $\bar{\mathbf{P}} \rightarrow \bar{\mathbf{P}}/\mathbf{W}$ is an equivalence of categories.
- (iii) The inclusion $\text{Alg}(\bar{\mathbf{P}}/\mathbf{W}, -) \rightarrow \text{Alg}(\bar{\mathbf{P}}, -)$ is an equivalence of categories.
- (iv) The canonical morphism $\mathbf{N}(\mathbf{P}) \rightarrow \bar{\mathbf{P}}$ is an equivalence.

Part II

Coherence from term rewriting systems

From ARS to TRS

In order to obtain result about actual categorical structures,
we need to go from ARS to term rewriting systems!

Term rewriting systems

A term rewriting system P consists of

- P_1 : operations with arities
- P_2 : equations between generated terms

Example

The TRS **Mon** for monoids is

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \middle| \begin{array}{l} \alpha : m(m(x, y), z) = m(x, m(y, z)) \\ \lambda : m(e, x) = x \\ \rho : m(x, e) = x \end{array} \right\rangle$$

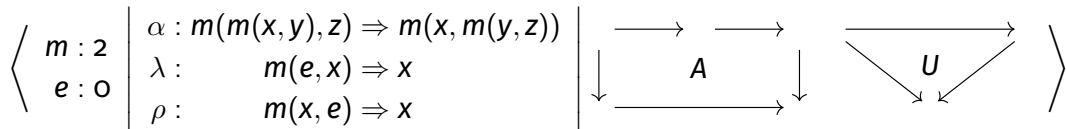
Term rewriting systems

An **extended term rewriting system** **P** consists of

- P_1 : operations with arities
- P_2 : 2-generators between generated terms
- P_3 : equations between 2-generators

Example

The 2-TRS **Mon** for **monoids** is



Term rewriting systems

An **extended term rewriting system** \mathbf{P} consists of

- \mathbf{P}_1 : operations with arities
- \mathbf{P}_2 : 2-generators between generated terms
- \mathbf{P}_3 : equations between 2-generators

Example

The 2-TRS **Mon** for **monoids** is

$$\left\langle \begin{array}{l|l} m : 2 & \alpha : m(m(x, y), z) \Rightarrow m(x, m(y, z)) \\ e : 0 & \lambda : m(e, x) \Rightarrow x \\ & \rho : m(x, e) \Rightarrow x \end{array} \right| \begin{array}{ccc} \longrightarrow & \longrightarrow & \\ \downarrow & A & \downarrow \\ \longrightarrow & & \end{array} \begin{array}{ccc} \longrightarrow & & \\ \swarrow U & & \searrow \\ & & \end{array} \right\rangle$$

NB: fixing m and n , \mathbf{P} induces an abstract rewriting systems on terms $m \rightarrow n$.

Lawvere theories

A **Lawvere theory** \mathcal{T} is a cartesian category such that objects are integers with cartesian product given by addition.

Lawvere theories

A **2-Lawvere theory** \mathcal{T} is a cartesian 2-category with invertible 2-cells such that objects are integers with cartesian product given by addition.

Lawvere theories

A **2-Lawvere theory** \mathcal{T} is a cartesian 2-category with invertible 2-cells such that objects are integers with cartesian product given by addition.

Any 2-TRS \mathbf{P} induces a 2-LT $\overline{\mathbf{P}}$ with

- morphisms $\langle t_1, \dots, t_n \rangle : m \rightarrow n$ are n -tuples of terms with m variables
- 2-cells are generated by 2-generators, quotiented by equations

Lawvere theories

A **2-Lawvere theory** \mathcal{T} is a cartesian 2-category with invertible 2-cells such that objects are integers with cartesian product given by addition.

Any 2-TRS \mathbf{P} induces a 2-LT $\overline{\mathbf{P}}$ with

- morphisms $\langle t_1, \dots, t_n \rangle : m \rightarrow n$ are n -tuples of terms with m variables
- 2-cells are generated by 2-generators, quotiented by equations

An **algebra** for \mathcal{T} is a product-preserving 2-functor $\mathcal{T} \rightarrow \mathbf{Cat}$.

Example

An algebra for $\overline{\mathbf{Mon}}$ is a monoidal category.

Algebras for Mon

With **Mon** being

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \middle| \begin{array}{l} \alpha : m(m(x, y), z) = m(x, m(y, z)) \\ \lambda : m(e, x) = x \\ \rho : m(x, e) = x \end{array} \right. \left| \begin{array}{c} \xrightarrow{\quad} \xrightarrow{\quad} \\ \downarrow \quad \quad \downarrow \\ \xrightarrow{\quad} \xrightarrow{\quad} \end{array} \right. \begin{array}{c} \xrightarrow{\quad} \\ \searrow \quad \swarrow \\ \quad U \end{array} \right\rangle$$

A

A functor $F : \overline{\mathbf{Mon}} \rightarrow \mathbf{Cat}$ consists of

- a category $\mathcal{C} = F1$

Algebras for Mon

With **Mon** being

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \middle| \begin{array}{l} \alpha : m(m(x, y), z) \Rightarrow m(x, m(y, z)) \\ \lambda : m(e, x) \Rightarrow x \\ \rho : m(x, e) \Rightarrow x \end{array} \right| \begin{array}{c} \xrightarrow{\quad} \xrightarrow{\quad} \\ \downarrow \quad \quad \downarrow \\ \xrightarrow{\quad} \xrightarrow{\quad} \end{array} \begin{array}{c} \xrightarrow{\quad} \\ \searrow \quad \swarrow \\ \quad U \end{array} \right\rangle$$

A functor $F : \overline{\mathbf{Mon}} \rightarrow \mathbf{Cat}$ consists of

- a category $\mathcal{C} = F1$
- thus $Fn = \mathcal{C}^n$

Algebras for Mon

With **Mon** being

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \middle| \begin{array}{l} \alpha : m(m(x, y), z) \Rightarrow m(x, m(y, z)) \\ \lambda : m(e, x) \Rightarrow x \\ \rho : m(x, e) \Rightarrow x \end{array} \right| \begin{array}{c} \xrightarrow{\quad} \xrightarrow{\quad} \\ \downarrow \quad \quad \downarrow \\ \xrightarrow{\quad} \xrightarrow{\quad} \end{array} \begin{array}{c} \xrightarrow{\quad} \\ \searrow \quad \swarrow \\ \quad \quad \quad \end{array} \right\rangle$$

A U

A functor $F : \overline{\mathbf{Mon}} \rightarrow \mathbf{Cat}$ consists of

- a category $\mathcal{C} = F1$
- thus $Fn = \mathcal{C}^n$
- two functors

$$\otimes = Fm : \mathcal{C}^2 \rightarrow \mathcal{C}$$

$$I = Fe : 1 \rightarrow \mathcal{C}$$

Algebras for Mon

With **Mon** being

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \left| \begin{array}{l} \alpha : m(m(x, y), z) \Rightarrow m(x, m(y, z)) \\ \lambda : m(e, x) \Rightarrow x \\ \rho : m(x, e) \Rightarrow x \end{array} \right. \right| \begin{array}{c} \xrightarrow{\quad} \xrightarrow{\quad} \\ \downarrow \quad \quad \downarrow \\ \xrightarrow{\quad} \xrightarrow{\quad} \end{array} \begin{array}{c} \xrightarrow{\quad} \\ \searrow \quad \swarrow \\ \quad U \end{array} \right\rangle$$

The diagram shows a square labeled A with horizontal arrows at the top and bottom, and vertical arrows on the left and right. To the right of the square is a triangle labeled U with a horizontal arrow at the top and two diagonal arrows meeting at a point below.

A functor $F : \overline{\mathbf{Mon}} \rightarrow \mathbf{Cat}$ consists of

- a category $\mathcal{C} = F1$
- thus $Fn = \mathcal{C}^n$
- two functors

$$\otimes = Fm : \mathcal{C}^2 \rightarrow \mathcal{C}$$

$$I = Fe : 1 \rightarrow \mathcal{C}$$

- satisfying the axioms of monoidal categories

Quotienting by rigid subgroupoids

Fix a 2-TRS \mathbf{P} with a subset $\mathbf{W} \subseteq \mathbf{P}_2$ of 2-generators generating a $(\mathbf{2}, \mathbf{1})$ -category \mathcal{W} .

Quotienting by rigid subgroupoids

Fix a 2-TRS \mathbf{P} with a subset $\mathbf{W} \subseteq \mathbf{P}_2$ of 2-generators generating a $(\mathbf{2}, \mathbf{1})$ -category \mathcal{W} .

\mathcal{W} is **2-rigid** when any parallel 2-cells are equal.

Quotienting by rigid subgroupoids

Fix a 2-TRS \mathbf{P} with a subset $\mathbf{W} \subseteq \mathbf{P}_2$ of 2-generators generating a $(\mathbf{2}, \mathbf{1})$ -category \mathcal{W} .

\mathcal{W} is **2-rigid** when any parallel 2-cells are equal.

Theorem

The quotient 2-functor $\bar{\mathbf{P}} \rightarrow \bar{\mathbf{P}}/\mathbf{W}$ is a local equivalence iff \mathcal{W} is 2-rigid.

Quotienting by rigid subgroupoids

Fix a 2-TRS \mathbf{P} with a subset $\mathbf{W} \subseteq \mathbf{P}_2$ of 2-generators generating a $(\mathbf{2}, \mathbf{1})$ -category \mathcal{W} .

\mathcal{W} is **2-rigid** when any parallel 2-cells are equal.

Theorem

The quotient 2-functor $\bar{\mathbf{P}} \rightarrow \bar{\mathbf{P}}/\mathbf{W}$ is a local equivalence iff \mathcal{W} is 2-rigid.

Theorem

If \mathbf{P} is \mathbf{W} -convergent then \mathcal{W} is 2-rigid.

Quotienting by rigid subgroupoids

Fix a 2-TRS \mathbf{P} with a subset $\mathbf{W} \subseteq \mathbf{P}_2$ of 2-generators generating a $(\mathbf{2}, \mathbf{1})$ -category \mathcal{W} .

\mathcal{W} is **2-rigid** when any parallel 2-cells are equal.

Theorem

The quotient 2-functor $\bar{\mathbf{P}} \rightarrow \bar{\mathbf{P}}/\mathbf{W}$ is a local equivalence iff \mathcal{W} is 2-rigid.

Theorem

If \mathbf{P} is \mathbf{W} -convergent then \mathcal{W} is 2-rigid.

Example

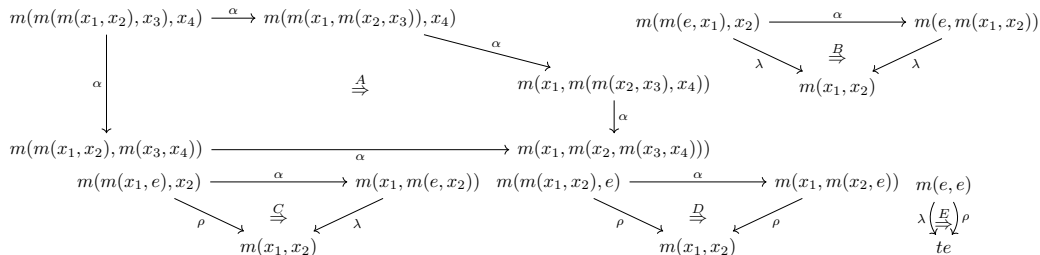
The theory **Mon** is \mathbf{W} -convergent with \mathbf{W} = all 2-cells.

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \left| \begin{array}{l} \alpha : m(m(x, y), z) \Rightarrow m(x, m(y, z)) \\ \lambda : m(e, x) \Rightarrow x \\ \rho : m(x, e) \Rightarrow x \end{array} \right. \right. \quad \begin{array}{c} \xrightarrow{\quad} \xrightarrow{\quad} \\ \downarrow \quad \quad \downarrow \\ \xrightarrow{\quad} \end{array} \quad \begin{array}{c} \xrightarrow{\quad} \\ \searrow \quad \swarrow \\ \quad \quad \quad \end{array} \quad \right\rangle$$

A U

Confluence of Mon

Note that there are 5 critical branchings:

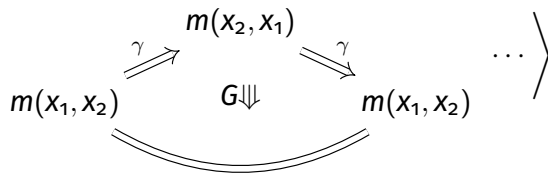


but 3 are derivable...

The case of symmetric monoidal categories

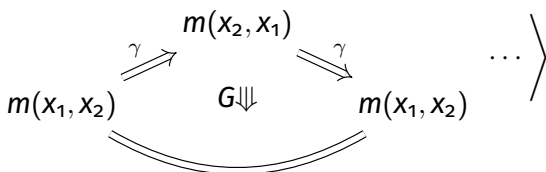
The theory of commutative monoids is

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \right| \begin{array}{l} \alpha : m(m(x_1, x_2), x_3) \Rightarrow m(x_1, m(x_2, x_3)) \\ \lambda : m(e, x_1) \Rightarrow x \\ \rho : m(x_1, e) \Rightarrow x \\ \gamma : m(x_1, x_2) \Rightarrow m(x_2, x_1) \end{array} \right|$$



The case of symmetric monoidal categories

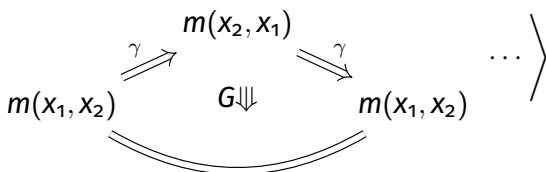
The theory of commutative monoids is

$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \right.$	$\left \begin{array}{l} \alpha : m(m(x_1, x_2), x_3) \Rightarrow m(x_1, m(x_2, x_3)) \\ \lambda : m(e, x_1) \Rightarrow x \\ \rho : m(x_1, e) \Rightarrow x \\ \gamma : m(x_1, x_2) \Rightarrow m(x_2, x_1) \end{array} \right $	
--	---	--

- if we take \mathcal{W} generated by α, λ, ρ , we are convergent:
every symmetric monoidal category is equivalent to a strict one
- but we can do more!

The case of symmetric monoidal categories

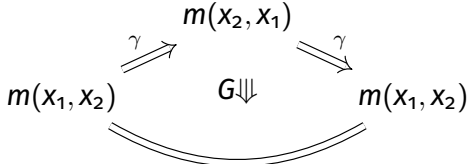
The theory of commutative monoids is

$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \right.$	$\left \begin{array}{l} \alpha : m(m(x_1, x_2), x_3) \Rightarrow m(x_1, m(x_2, x_3)) \\ \lambda : m(e, x_1) \Rightarrow x \\ \rho : m(x_1, e) \Rightarrow x \\ \gamma : m(x_1, x_2) \Rightarrow m(x_2, x_1) \\ \delta : m(x_1, m(x_2, x_3)) \Rightarrow m(x_2, m(x_1, x_3)) \end{array} \right $	
--	---	--

- it can be completed as a locally confluent presentation by adding a generator δ and a bunch of coherence relations

The case of symmetric monoidal categories

The theory of commutative monoids is

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \right| \begin{array}{l} \alpha : m(m(x_1, x_2), x_3) \Rightarrow m(x_1, m(x_2, x_3)) \\ \lambda : m(e, x_1) \Rightarrow x \\ \rho : m(x_1, e) \Rightarrow x \\ \gamma : m(x_1, x_2) \Rightarrow m(x_2, x_1) \\ \delta : m(x_1, m(x_2, x_3)) \Rightarrow m(x_2, m(x_1, x_3)) \end{array} \right.$$


- it is not terminating otherwise we could show “full coherence” including

$$\begin{array}{ccc}
 & \xrightarrow{\gamma_{x,x}} & \\
 x \otimes x & & x \otimes x \\
 & \xleftarrow{\text{id}_{x \otimes x}} &
 \end{array}$$

The case of symmetric monoidal categories

The theory of commutative monoids is

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \right| \begin{array}{l} \alpha : m(m(x_1, x_2), x_3) \Rightarrow m(x_1, m(x_2, x_3)) \\ \lambda : m(e, x_1) \Rightarrow x \\ \rho : m(x_1, e) \Rightarrow x \\ \gamma : m(x_1, x_2) \Rightarrow m(x_2, x_1) \\ \delta : m(x_1, m(x_2, x_3)) \Rightarrow m(x_2, m(x_1, x_3)) \end{array} \right.$$

- restricting to affine terms (without repeated variables is not enough):

$$m(x_1, x_2) \xrightarrow{\gamma(x_1, x_2)} m(x_2, x_1) \xrightarrow{\gamma(x_2, x_1)} m(x_1, x_2)$$

- but we don't need both $m(x_1, x_2) \Rightarrow m(x_2, x_1)$ and $m(x_2, x_1) \Rightarrow m(x_1, x_2)$!

The case of symmetric monoidal categories

The theory of commutative monoids is

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \right| \begin{array}{l} \alpha : m(m(x_1, x_2), x_3) \Rightarrow m(x_1, m(x_2, x_3)) \\ \lambda : m(e, x_1) \Rightarrow x \\ \rho : m(x_1, e) \Rightarrow x \\ \gamma : m(x_1, x_2) \Rightarrow m(x_2, x_1) \\ \delta : m(x_1, m(x_2, x_3)) \Rightarrow m(x_2, m(x_1, x_3)) \end{array} \right.$$

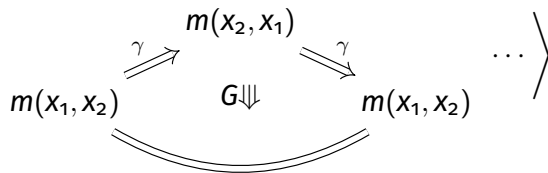
- if we only keep morphisms “sorting variables”, we are almost terminating excepting for situations such as $m(e, e) \Rightarrow m(e, e)$ which can be removed:

$$\begin{array}{ccc} m(e, e) & \xRightarrow{\quad} & m(e, e) \\ & \searrow \quad \swarrow & \\ & e & \end{array}$$

The case of symmetric monoidal categories

The theory of commutative monoids is

$$\left\langle \begin{array}{l} m : 2 \\ e : 0 \end{array} \right| \begin{array}{l} \alpha : m(m(x_1, x_2), x_3) \Rightarrow m(x_1, m(x_2, x_3)) \\ \lambda : m(e, x_1) \Rightarrow x \\ \rho : m(x_1, e) \Rightarrow x \\ \gamma : m(x_1, x_2) \Rightarrow m(x_2, x_1) \\ \delta : m(x_1, m(x_2, x_3)) \Rightarrow m(x_2, m(x_1, x_3)) \end{array} \right.$$



Theorem

In a symmetric monoidal category, every diagram whose source is a tensor product of distinct objects commutes.

Part III

Conclusion

Rigidity!

A quotient of (2-)category by a subgroupoid \mathcal{W} is **coherent** when \mathcal{W} is **rigid**.

This is the case when \mathcal{W} is generated by a **convergent rewriting system**.

This also explains situations such as coherence for rig categories:

$$\begin{array}{ccc} & (a + b)(c + d) & \\ \swarrow & & \searrow \\ a(c + d) + b(c + d) & & (a + b)c + (a + b)d \\ \downarrow & & \downarrow \\ ac + ad + bc + bd & \xrightarrow{\sim} & ac + bc + ad + bd \end{array}$$

Thanks!

Questions?