CONVERGENT PRESENTATIONS OF MONOIDAL CATEGORIES

Samuel Mimram

École Polytechnique

9th International School on Rewriting

July 6, 2017

Presentation

Here, the goal is to build **presentations** of algebraic “objects” (such as *monoids*):

- these provide small descriptions of the objects: they can be finite even though the object is not
- computations can be performed directly on those: homology, generating series, etc.
- rewriting theory can help!

In this course

1. we detail presentations of monoids,
2. generalize to presentations of monoidal categories,

n. this is the starting point of a general pattern.

Some references

- 1993: Albert Burroni
  Higher-dimensional word problems
  with applications to equational logic

- 2003: Yves Lafont
  Towards an algebraic theory of Boolean circuits

- 2014: Samuel Mimram
  Towards 3-dimensional rewriting theory

- 2016: Yves Guiraud, Philippe Malbos
  Polygraphs of finite derivation type

PRESENTATIONS
OF
MONOIDS

Monoids

A **monoid** $(M, \cdot, 1)$ consists of

- a set M,
- a *multiplication* $\cdot : M \times M \to M$,
- a *unit* $1 \in M$,

such that

- multiplication is associative

$$(a \cdot b) \cdot c \quad = \quad a \cdot (b \cdot c)\,,$$

- unit is a neutral element

$$1 \cdot a \quad = \quad a \quad = \quad a \cdot 1\,.$$

Monoids

Examples

- $(\mathbb{N}, +, 0)$
- $(\mathbb{N}, \times, 1)$
- matrices of size $n \times n$
- every group is a monoid:
  - $(\mathbb{Z}, +, 0)$, $(\mathbb{Z}/n\mathbb{Z}, +, 0)$, $(\mathbb{Q}, +, 0)$, $(\mathbb{Q}, \times, 1)$, …
  - S_n: group of permutations of n elements
  - etc.
- etc.

Morphisms of monoids

A **morphism**

$$f \quad : \quad M \quad \to \quad N$$

between monoids $(M, \times_M, 1_M)$ and $(N, \times_N, 1_N)$ is a function

$$f \quad : \quad M \quad \to \quad N$$

which

- preserves product: for $u, v \in M$,

$$f(u \times_M v) \quad = \quad f(u) \times_N f(v)\,,$$

- preserves unit:

$$f(1_M) \quad = \quad 1_N\,.$$

The free monoid

Given a set G, the **free monoid** $(G^*, \cdot, 1)$ has

- the set G^* of words over G as elements,
- the concatenation $\cdot$ as multiplication,
- the empty word 1 as unit.

The free monoid

Given a set G, the **free monoid** $(G^*, \cdot, 1)$ has

- the set G^* of words over G as elements,
- the concatenation $\cdot$ as multiplication,
- the empty word 1 as unit.

Proposition

Given a monoid $(M, \times, 1)$ any function

$$f \quad : \quad G \quad \to \quad M$$

extends uniquely as a morphism of monoids

$$f^* \quad : \quad G^* \quad \to \quad M\,.$$

The free monoid

Given a set G, the **free monoid** $(G^*, \cdot, 1)$ has

- the set G^* of words over G as elements,
- the concatenation $\cdot$ as multiplication,
- the empty word 1 as unit.

Proposition

Given a monoid $(M, \times, 1)$ any function

$$f \quad : \quad G \quad \to \quad M$$

extends uniquely as a morphism of monoids

$$f^* \quad : \quad G^* \quad \to \quad M\,.$$

Proof.

Given a word $a_1 \dots a_n \in G^*$, we had to define

$$f^*(a_1 \dots a_n) \quad = \quad f^*(a_1) \times \dots \times f^*(a_n) \quad = \quad f(a_1) \times \dots \times f(a_n)\,. \square$$

Isomorphisms of monoids

A morphism of monoids

$$f \quad : \quad M \quad \to \quad N$$

is an **isomorphism** when there exists a morphism

$$g \quad : \quad N \quad \to \quad M$$

such that

$$g \circ f = \mathrm{id}_M \qquad \text{and} \qquad f \circ g = \mathrm{id}_N \,.$$

This means that M and N are the same monoid up to renaming elements.

Isomorphisms of monoids

Example

The function

$$f \quad : \quad \mathbb{N} \rightarrow \{a\}^* \\ n \mapsto a^n$$

is a morphism

$$f(m) + f(n) = a^m \cdot a^n = a^{m+n} = f(m+n)$$

$$f(0) = a^0 = 1$$

Isomorphisms of monoids

Example

The function

$$\begin{array}{rcrcl} f & : & \mathbb{N} & \to & \{a\}^* \\ & & n & \mapsto & a^n \end{array}$$

is a morphism

$$f(m) + f(n) = a^m \cdot a^n = a^{m+n} = f(m+n)$$

$$f(0) = a^0 = 1$$

which is an isomorphism whose inverse is

$$\begin{array}{rcrcl} g & : & \{a\}^* & \to & \mathbb{N} \\ & & a^n & \mapsto & n\,. \end{array}$$

Isomorphisms of monoids

Lemma

A morphism of monoids

$$f \quad : \quad M \quad \to \quad N$$

which is invertible *<u>as a function</u>* *is an isomorphism.*

Isomorphisms of monoids

Lemma

A morphism of monoids

$$f \quad : \quad M \quad \to \quad N$$

which is invertible as a function is an isomorphism.

Proof.

We show that the inverse function

$$g \quad : \quad N \quad \to \quad M$$

is a morphism of monoids:

$$g(u \cdot v) = g(u) \cdot g(v)$$

Isomorphisms of monoids

Lemma

A morphism of monoids

$$f \quad : \quad M \quad \to \quad N$$

which is invertible as a function is an isomorphism.

Proof.

We show that the inverse function

$$g \quad : \quad N \quad \to \quad M$$

is a morphism of monoids:

$$g(u \cdot v) = g(f(g(u)) \cdot f(g(v))) \qquad = g(u) \cdot g(v)$$

Isomorphisms of monoids

Lemma

A morphism of monoids

$$f \quad : \quad M \quad \rightarrow \quad N$$

which is invertible as a function is an isomorphism.

Proof.

We show that the inverse function

$$g \quad : \quad N \quad \rightarrow \quad M$$

is a morphism of monoids:

$$g(u \cdot v) = g(f(g(u)) \cdot f(g(v))) = g(f(g(u) \cdot g(v))) = g(u) \cdot g(v)$$

Isomorphisms of monoids

Lemma

A morphism of monoids

$$f \quad : \quad M \quad \to \quad N$$

which is invertible as a function is an isomorphism.

Proof.

We show that the inverse function

$$g \quad : \quad N \quad \to \quad M$$

is a morphism of monoids:

$$g(u \cdot v) = g(f(g(u)) \cdot f(g(v))) = g(f(g(u) \cdot g(v))) = g(u) \cdot g(v)$$

and

$$g(1_N) = g(f(1_M)) = 1_M \,.$$

□

Congruence on a monoid

A **congruence** $\approx$ on a monoid $(M, \cdot, 1)$ is an equivalence relation on M such that

$$v \approx v' \qquad \text{implies} \qquad u \cdot v \cdot w \approx u \cdot v' \cdot w\,.$$

Congruence on a monoid

A **congruence** $\approx$ on a monoid $(M, \cdot, 1)$ is an equivalence relation on M such that

$$v \approx v' \qquad \text{implies} \qquad u \cdot v \cdot w \approx u \cdot v' \cdot w\,.$$

We recall that an *equivalence relation* is

- reflexive:
$$u \approx u$$
- symmetric:
$$u \approx v \qquad \text{implies} \qquad v \approx u$$
- transitive:
$$u \approx v \quad \text{and} \quad v \approx w \qquad \text{implies} \qquad u \approx w$$

Congruence on a monoid

A **congruence** $\approx$ on a monoid $(M, \cdot, 1)$ is an equivalence relation on M such that

$$v \approx v' \qquad \text{implies} \qquad u \cdot v \cdot w \approx u \cdot v' \cdot w\,.$$

In this case, one can define a **quotient monoid**

$$M/\!\approx$$

where

- an element $[u]$ is the equivalence class of some $u \in M$,
- multiplication is given by

$$[u] \cdot [v] \quad = \quad [u \cdot v]\,,$$

- unit is $[1]$.

Congruence on a monoid

Example

Consider the monoid $M = \{a\}^*$ and the smallest congruence $\approx$ such that $aaa \approx 1$.

Congruence on a monoid

Example

Consider the monoid $M = \{a\}^*$ and the smallest congruence $\approx$ such that $aaa \approx 1$.

The equivalence classes of $M/\approx$ are

$$[1] = \{a^{3n}\} \qquad [a] = \{a^{3n+1}\} \qquad [aa] = \{a^{3n+2}\}$$

and multiplication table is

$\cdot$	$[1]$	$[a]$	$[aa]$
$[1]$	$[1]$	$[a]$	$[aa]$
$[a]$	$[a]$	$[aa]$	$[1]$
$[aa]$	$[aa]$	$[1]$	$[a]$

Congruence on a monoid

Example

Consider the monoid $M = \{a\}^*$ and the smallest congruence $\approx$ such that $aaa \approx 1$.

The equivalence classes of $M/\approx$ are

$$[1] = \{a^{3n}\} \qquad [a] = \{a^{3n+1}\} \qquad [aa] = \{a^{3n+2}\}$$

and multiplication table is

$\cdot$	$[1]$	$[a]$	$[aa]$
$[1]$	$[1]$	$[a]$	$[aa]$
$[a]$	$[a]$	$[aa]$	$[1]$
$[aa]$	$[aa]$	$[1]$	$[a]$

Notice that this monoid is isomorphic to $\mathbb{N}/3\mathbb{N}$.

In order to manipulate a monoid
one would like to come up
with a small description of it.

Presentations of monoids

A **presentation** of a monoid M is a pair

$$\langle G \mid R \rangle$$

where

- G is a set of **generators**
- $R \subseteq G^* \times G^*$ is a set of **relations**

such that

$$M \quad \cong \quad G^* / {\approx^R}$$

where $\approx^R$ is the smallest congruence such that

$$(u, v) \in R \qquad \text{implies} \qquad u \approx^R v \,.$$

Presentations of monoids

Example

- $\mathbb{N}$ (additive) is presented by

$$\langle a \mid \, \rangle$$

Presentations of monoids

Example

- $\mathbb{N}$ (additive) is presented by

$$\langle a \mid \; \rangle$$

- $\mathbb{N}/3\mathbb{N}$ (additive) is presented by

$$\langle a \mid aaa = 1 \rangle$$

Presentations of monoids

Example

- $\mathbb{N}$ (additive) is presented by

$$\langle a \mid \, \rangle$$

- $\mathbb{N}/3\mathbb{N}$ (additive) is presented by

$$\langle a \mid aaa = 1 \rangle$$

- $\mathbb{N} \times \mathbb{N}$ (additive) is presented by

$$\langle a, b \mid ba = ab \rangle$$

Presentations of monoids

Example

- $\mathbb{N}$ (additive) is presented by

$$\langle a \mid \; \rangle$$

- $\mathbb{N}/3\mathbb{N}$ (additive) is presented by

$$\langle a \mid aaa = 1 \rangle$$

- $\mathbb{N} \times \mathbb{N}$ (additive) is presented by

$$\langle a, b \mid ba = ab \rangle$$

- S_3 is presented by

$$\langle a, b \mid bab = aba, aa = 1, bb = 1 \rangle$$

Presentations of monoids

To sum up, when a monoid M admits a presentation

$$\langle G \mid R \rangle$$

this means that

- we have an *interpretation* of elements of G as elements of M

Presentations of monoids

To sum up, when a monoid M admits a presentation

$$\langle G \mid R \rangle$$

this means that

- we have an *interpretation* of elements of G as elements of M
- the elements of G *generate* the monoid M: every element of M can be written as a product of (images of) elements of G

Presentations of monoids

To sum up, when a monoid M admits a presentation

$$\langle G \mid R \rangle$$

this means that

- we have an *interpretation* of elements of G as elements of M
- the elements of G *generate* the monoid M: every element of M can be written as a product of (images of) elements of G
- if two products of elements of G

  $$a_1 \ldots a_m \qquad \text{and} \qquad b_1 \ldots b_n$$

  denote the same element of M then they are related by (the congruence generated by) R

How do we show
that we actually have
a presentation?

Constructing presentations of monoids

For instance,

$$\mathbb{N} \times \mathbb{N} \quad \cong \quad \{a, b\}^* /{\approx}$$

where $\approx$ is the congruence generated by $ba \approx ab$.

Constructing presentations of monoids

For instance,

$$\mathbb{N} \times \mathbb{N} \quad \cong \quad \{a, b\}^* / {\approx}$$

where $\approx$ is the congruence generated by $ba \approx ab$.

In each equivalence class (wrt $\approx$) there is a unique word of the form

$$a^m b^n$$

with $(m, n) \in \mathbb{N} \times \mathbb{N}$, called a **canonical form**, thus the bijection!

For instance,

$$abaa \quad \approx \quad aaba \quad \approx \quad aaab\,.$$

The word problem

Given a presentation $\langle G \mid R \rangle$ the **word problem** is

- *input*: $u, v \in G^*$
- *output*: do we have $u \approx v$?

The word problem

Given a presentation $\langle G \mid R \rangle$ the **word problem** is

- *input*: $u, v \in G^*$
- *output*: do we have $u \approx v$?

In general, this is *undecidable*!

The word problem

Given a presentation $\langle G \mid R \rangle$ the **word problem** is

- *input*: $u, v \in G^*$
- *output*: do we have $u \approx v$?

In general, this is *undecidable*!

For instance (Tseitin):

$$\left\langle a, c, b, d, e \;\middle|\; \begin{array}{l} ac = ca, ad = da, bc = cb, bd = db, \\ eca = ce, edb = de, ccae = cca \end{array} \right\rangle$$

How do we come up
with canonical forms?

Normal forms!

String rewriting systems

A *presentation*

$$\langle G \mid R \rangle$$

is another name for a **string rewriting system** where

- G is the *alphabet*,
- the *rules* are the elements $(v, v') \in R$.

String rewriting systems

A *presentation*

$$\langle G \mid R \rangle$$

is another name for a **string rewriting system** where

- G is the *alphabet*,
- the *rules* are the elements $(v, v') \in R$.

When $(v, v') \in R$ and $u, w \in G^*$, we have a **rewriting step**

$$uvw \quad \Rightarrow \quad uv'w\,.$$

String rewriting systems

A *presentation*

$$\langle G \mid R \rangle$$

is another name for a **string rewriting system** where

- G is the *alphabet*,
- the *rules* are the elements $(v, v') \in R$.

When $(v, v') \in R$ and $u, w \in G^*$, we have a **rewriting step**

$$uvw \quad \Rightarrow \quad uv'w\,.$$

A **rewriting path**

$$u \quad \overset{*}{\Rightarrow} \quad v$$

is a sequence of rewriting steps.

String rewriting systems

A *presentation*

$$\langle G \mid R \rangle$$

is another name for a **string rewriting system** where

- G is the *alphabet*,
- the *rules* are the elements $(v, v') \in R$.

When $(v, v') \in R$ and $u, w \in G^*$, we have a **rewriting step**

$$uvw \quad \Rightarrow \quad uv'w\,.$$

A **rewriting path**

$$u \quad \stackrel{*}{\Rightarrow} \quad v$$

is a sequence of rewriting steps. A **rewriting equivalence**

$$u \quad \stackrel{*}{\Leftrightarrow} \quad v$$

is a sequence of forward ($\Rightarrow$) or backward ($\Leftarrow$) rewriting steps.

String rewriting systems

By definition, we have

$$u \approx^R v \qquad \text{iff} \qquad u \overset{*}{\Leftrightarrow} v\,.$$

String rewriting systems

By definition, we have

$$u \approx^R v \qquad \text{iff} \qquad u \overset{*}{\Leftrightarrow} v\,.$$

Lemma (Church-Rosser)

When the rewriting system is convergent,

$$u \overset{*}{\Leftrightarrow} v \qquad \textit{iff} \qquad \hat{u} = \hat{v}\,.$$

This means that every equivalence class $[u]$ contains exactly one normal form, which is $\hat{u}$.

Constructing presentations

Given $\langle G \mid R \rangle$ and a monoid M, to show

$$G^*/{\approx^R} \quad \cong \quad M$$

one can use the following method:

Constructing presentations

Given $\langle G \mid R \rangle$ and a monoid M, to show

$$G^*/{\approx^R} \quad \cong \quad M$$

one can use the following method:

1. construct a function $f : G \to M$

Constructing presentations

Given $\langle G \mid R\rangle$ and a monoid M, to show

$$G^*/{\approx^R} \quad \cong \quad M$$

one can use the following method:

1. construct a function $f : G \to M$
2. *extend it as a morphism of monoids* $f : G^* \to M$

Constructing presentations

Given $\langle G \mid R \rangle$ and a monoid M, to show

$$G^*/\!\approx^R \quad \cong \quad M$$

one can use the following method:

1. construct a function $f : G \to M$
2. *extend it as a morphism of monoids* $f : G^* \to M$
3. check that for every relation $(u, v) \in R$ we have $f(u) = f(v)$

Constructing presentations

Given $\langle G \mid R \rangle$ and a monoid M, to show

$$G^*/\approx^R \quad \cong \quad M$$

one can use the following method:

1. construct a function $f : G \to M$
2. *extend it as a morphism of monoids* $f : G^* \to M$
3. check that for every relation $(u, v) \in R$ we have $f(u) = f(v)$
4. *deduce that we have a well-defined* $f : G^*/\approx^R \to M$

Constructing presentations

Given $\langle G \mid R \rangle$ and a monoid M, to show

$$G^*/{\approx^R} \quad \cong \quad M$$

one can use the following method:

1. construct a function $f : G \to M$
2. *extend it as a morphism of monoids* $f : G^* \to M$
3. check that for every relation $(u, v) \in R$ we have $f(u) = f(v)$
4. *deduce that we have a well-defined* $f : G^*/{\approx^R} \to M$
5. check that the rewriting system is convergent

Constructing presentations

Given $\langle G \mid R \rangle$ and a monoid M, to show

$$G^*/{\approx^R} \quad \cong \quad M$$

one can use the following method:

1. construct a function $f : G \to M$
2. *extend it as a morphism of monoids* $f : G^* \to M$
3. check that for every relation $(u, v) \in R$ we have $f(u) = f(v)$
4. *deduce that we have a well-defined* $f : G^*/{\approx^R} \to M$
5. check that the rewriting system is convergent
6. *deduce that elements of* $G^*/{\approx^R}$ *are represented by normal forms*

Constructing presentations

Given $\langle G \mid R \rangle$ and a monoid M, to show

$$G^* / {\approx^R} \quad \cong \quad M$$

one can use the following method:

1. construct a function $f : G \to M$
2. *extend it as a morphism of monoids* $f : G^* \to M$
3. check that for every relation $(u, v) \in R$ we have $f(u) = f(v)$
4. *deduce that we have a well-defined* $f : G^* / {\approx^R} \to M$
5. check that the rewriting system is convergent
6. *deduce that elements of* $G^* / {\approx^R}$ *are represented by normal forms*
7. show that f induces a bijection between normal forms and elements of M

Constructing presentations

Given $\langle G \mid R\rangle$ and a monoid M, to show

$$G^*/{\approx^R} \quad \cong \quad M$$

one can use the following method:

1. construct a function $f : G \to M$
2. *extend it as a morphism of monoids* $f : G^* \to M$
3. check that for every relation $(u, v) \in R$ we have $f(u) = f(v)$
4. *deduce that we have a well-defined* $f : G^*/{\approx^R} \to M$
5. check that the rewriting system is convergent
6. *deduce that elements of $G^*/{\approx^R}$ are represented by normal forms*
7. show that f induces a bijection between normal forms and elements of M
8. *deduce that $f : G^*/{\approx^R} \to M$ is an isomorphism.*

Exercises

1. Show that S_3 admits the presentation

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

2. Propose a presentation for S_4.
3. Propose a presentation for S_n.

Correction

We want to show that S_3 is presented by

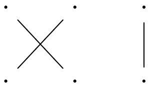
$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

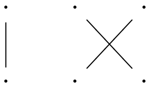
Correction

We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$ by

$f(a) =$ 

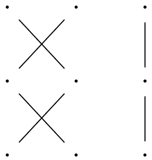
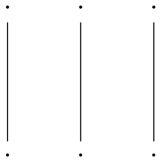
$f(b) =$ 

Correction

We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied

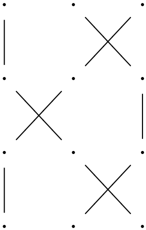
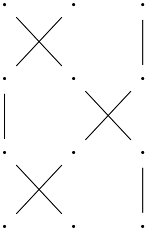
$f(aa) =$  $=$  $= f(1)$

Correction

We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied

$f(bab)$ $=$  $=$  $=$ $f(aba)$

Correction

We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied
5. we check that the rewriting system

$$aa \Rightarrow 1 \qquad bb \Rightarrow 1 \qquad bab \Rightarrow aba$$

   is convergent.

Correction

We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied
5. we check that the rewriting system

$$aa \Rightarrow 1 \qquad bb \Rightarrow 1 \qquad bab \Rightarrow aba$$

   is convergent. *Termination*: the rules decrease the length, or preserve it an decrease the number of b.

Correction

We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied
5. we check that the rewriting system

$$aa \Rightarrow 1 \qquad bb \Rightarrow 1 \qquad bab \Rightarrow aba$$

is convergent. *Confluence*: the critical branchings are

Correction

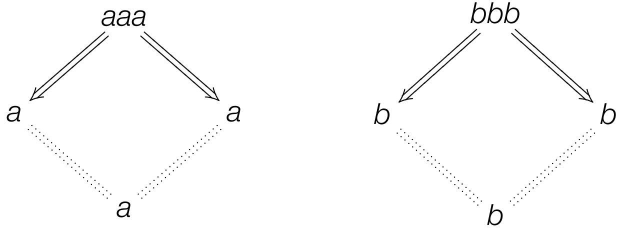
We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied
5. we check that the rewriting system

$$aa \Rightarrow 1 \qquad bb \Rightarrow 1 \qquad bab \Rightarrow aba$$

is convergent. *Confluence*: the critical branchings are



Correction

We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied
5. we check that the rewriting system

$$aa \Rightarrow 1 \qquad bb \Rightarrow 1 \qquad bab \Rightarrow aba$$

is convergent. *Confluence*: the critical branchings are

$$ab \Leftarrow bbab \Rightarrow baba \qquad\qquad abab \Leftarrow babb \Rightarrow ba$$

Correction

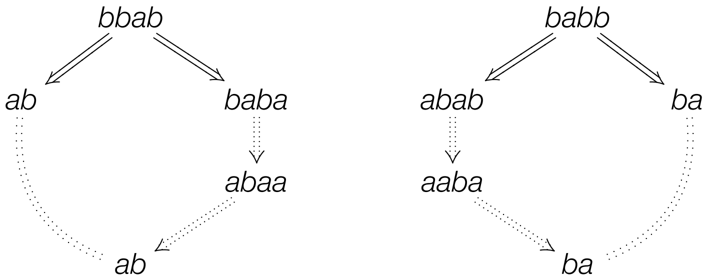
We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied
5. we check that the rewriting system

$$aa \Rightarrow 1 \qquad bb \Rightarrow 1 \qquad bab \Rightarrow aba$$

is convergent. *Confluence*: the critical branchings are



Correction

We want to show that S_3 is presented by

$$\langle a, b \mid aa = 1, bb = 1, bab = aba \rangle$$

1. we define $f : \{a, b\} \to S_3$
3. we check that the relations are satisfied
5. we check that the rewriting system

$$aa \Rightarrow 1 \qquad bb \Rightarrow 1 \qquad bab \Rightarrow aba$$

   is convergent.
7. normal forms are

$1 \qquad a \qquad ab \qquad aba \qquad b \qquad ba$

their images are different and there are $6 = 3!$ of them.

Correction

A presentation for S_4 is

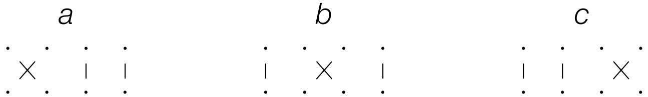
$$\langle a, b, c \mid aa = 1, bb = 1, cc = 1, bab = aba, cbc = bcb, ca = ac \rangle$$

Correction

A presentation for S_4 is

$$\langle a, b, c \mid aa = 1, bb = 1, cc = 1, bab = aba, cbc = bcb, ca = ac \rangle$$

The interpretation of the generators is

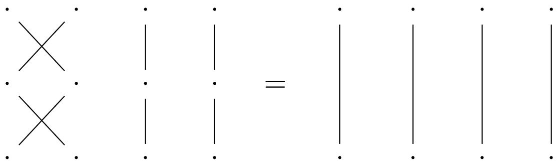


Correction

A presentation for S_4 is

$$\langle a, b, c \mid aa = 1, bb = 1, cc = 1, bab = aba, cbc = bcb, ca = ac \rangle$$

$aa = 1$ corresponds to

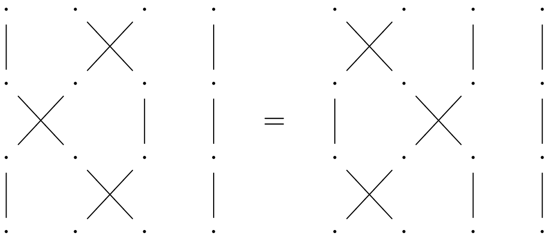


Correction

A presentation for S_4 is

$$\langle a, b, c \mid aa = 1, bb = 1, cc = 1, bab = aba, cbc = bcb, ca = ac \rangle$$

$bab = aba$ corresponds to

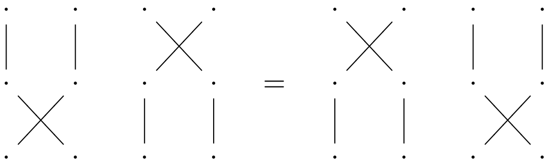


Correction

A presentation for S_4 is

$$\langle a, b, c \mid aa = 1, bb = 1, cc = 1, bab = aba, cbc = bcb, ca = ac\rangle$$

$ca = ac$ corresponds to



Correction

A presentation for S_4 is

$$\langle a, b, c \mid aa = 1, bb = 1, cc = 1, bab = aba, cbc = bcb, ca = ac \rangle$$

A presentation for S_n has

- generators: $a_1, \ldots, a_{n-1}$
- relations: for $1 \le i < i + 1 < n$,

$$a_{i+1} a_i a_{i+1} \quad = \quad a_i a_{i+1} a_i$$

and, for $1 \le i < i + 1 < j < n$,

$$a_j a_i \quad = \quad a_i a_j$$

PRESENTATIONS OF MONOIDAL CATEGORIES

Higher-dimensional rewriting

The idea of **higher-dimensional rewriting** is that we have the following hierarchy of rewriting systems:

0. words

$$u$$

Higher-dimensional rewriting

The idea of **higher-dimensional rewriting** is that we have the following hierarchy of rewriting systems:

0. words

$$u$$

1. string rewriting systems

$$u \Longrightarrow v$$

Higher-dimensional rewriting

The idea of **higher-dimensional rewriting** is that we have the following hierarchy of rewriting systems:

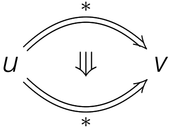
0. words

$$u$$

1. string rewriting systems

$$u \Longrightarrow v$$

2. 2-dimensional rewriting systems



Higher-dimensional rewriting

The idea of **higher-dimensional rewriting** is that we have the following hierarchy of rewriting systems:

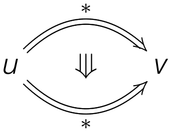
0. words

$$u$$

1. string rewriting systems

$$u \Longrightarrow v$$

2. 2-dimensional rewriting systems



n. …

Plan

We focus here on 2-dimensional rewriting systems.

1. What do they present?
   Monoidal categories

2. How do we extend classical rewriting techniques?
   Termination, confluence, …

3. Some examples of presented monoidal categories.

Rewriting systems

Up to now a rewriting system was $\langle G \mid R \rangle$ with $R \subseteq G^* \times G^*$.

We slightly modify the definition and notations.

Rewriting systems

Up to now a rewriting system was $\langle G \mid R \rangle$ with $R \subseteq G^* \times G^*$.

We slightly modify the definition and notations.

A **1-dimensional rewriting system** consists of

- *letters*: a set G_1
- *rules*: a set R together with two functions

$$s, t \quad : \quad R \quad \to \quad G_1^*$$

Rewriting systems

Up to now a rewriting system was $\langle G \mid R \rangle$ with $R \subseteq G^* \times G^*$.

We slightly modify the definition and notations.

A **1-dimensional rewriting system** consists of

- *object generators*: a set G_1
- *morphism generators*: a set G_2 together with two functions

$$s, t \quad : \quad G_2 \quad \to \quad G_1^*$$

Rewriting systems

Up to now a rewriting system was $\langle G \mid R \rangle$ with $R \subseteq G^* \times G^*$.

We slightly modify the definition and notations.

A **1-dimensional rewriting system** consists of

- *object generators*: a set G_1
- *morphism generators*: a set G_2 together with two functions

$$s, t \quad : \quad G_2 \quad \to \quad G_1^*$$

what we write

$$\langle G_1 \mid G_2 \rangle$$

For instance

$$\langle a, b \mid \gamma : ba \Rightarrow ab \rangle$$

Rewriting paths

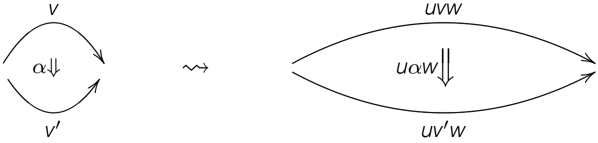
We can now give names to rewriting steps: given a rule in G_2

$$\alpha \quad : \quad v \quad \Rightarrow \quad v'$$

and $u, w \in G_1^*$, we have a **rewriting step**

$$u\alpha w \quad : \quad uvw \quad \Rightarrow \quad uv'w$$

which is the rule α "in the context (u, w)".



Rewriting paths

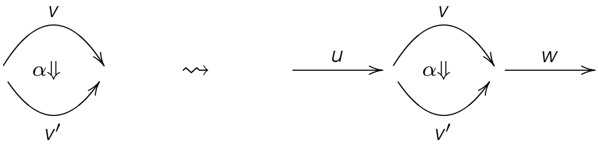
We can now give names to rewriting steps: given a rule in G_2

$$\alpha \quad : \quad v \quad \Rightarrow \quad v'$$

and $u, w \in G_1^*$, we have a **rewriting step**

$$u\alpha w \quad : \quad uvw \quad \Rightarrow \quad uv'w$$

which is the rule α "in the context (u, w)".



Rewriting paths

We can now give names to rewriting steps: given a rule in G_2

$$\alpha \quad : \quad v \quad \Rightarrow \quad v'$$

and $u, w \in G_1^*$, we have a **rewriting step**

$$u\alpha w \quad : \quad uvw \quad \Rightarrow \quad uv'w$$

which is the rule α "in the context (u, w)".

A **rewriting path** is thus of the form

$$u_1\alpha_1 w_1 \ \cdot \ u_2\alpha_2 w_2 \ \cdot \ \ldots \ \cdot \ u_n\alpha_n w_n$$

where "$\cdot$" denotes concatenation.

Rewriting paths

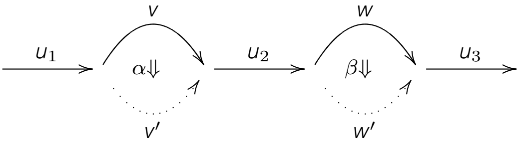
Suppose given a word of the form

$$u_1 v u_2 w u_3$$

and two rules

$$\alpha : v \Rightarrow v' \qquad\qquad \beta : w \Rightarrow w'$$

We can use α and β independently, and we will not distinguish between the order in which they are applied.

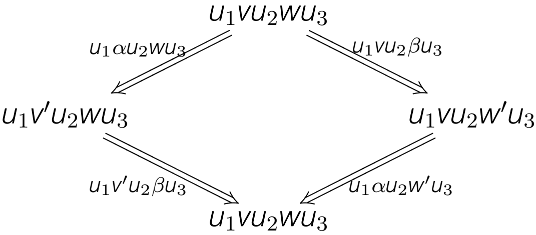


Rewriting paths

In the following, we will quotient it and identify paths of the following form:

$$u_1 \alpha u_2 w u_3 \cdot u_1 v' u_2 \beta u_3 \quad = \quad u_1 v u_2 \beta u_3 \cdot u_1 \alpha u_2 w' u_3$$

Graphically,



Order does not matter when rewriting at independent positions.

The category of rewriting paths

Given a rewriting system G of the form

$$\langle G_1 \mid G_2 \rangle$$

we can form a category G^* where

- an object is a word in G_1^*
- a morphism is a rewriting path

$$\phi \quad : \quad u \quad \overset{*}{\Rightarrow} \quad v$$

- composition is given by concatenation

$$u \quad \overset{\phi}{\Rightarrow} \quad v \quad \overset{\psi}{\Rightarrow} \quad w$$

- identities are empty paths

Categories

A **category** C consists of

such that

Categories

A **category** C consists of

- a set $\mathrm{Ob}(C)$ of objects,

such that

Categories

A **category** C consists of

- a set $\mathrm{Ob}(C)$ of objects,
- for every objects $x, y \in C$, a set $C(x, y)$ of morphisms, we write $f : x \Rightarrow y$ for $f \in C(x, y)$,

such that

Categories

A **category** C consists of

- a set $\mathrm{Ob}(C)$ of objects,
- for every objects $x, y \in C$, a set $C(x, y)$ of morphisms, we write $f : x \Rightarrow y$ for $f \in C(x, y)$,
- a composition operation: given

$$f : x \Rightarrow y \qquad \text{and} \qquad g : y \Rightarrow z$$

  we have

$$f \cdot g \quad : \quad x \quad \Rightarrow \quad z$$

such that

Categories

A **category** C consists of

- a set $\mathrm{Ob}(C)$ of objects,
- for every objects $x, y \in C$, a set $C(x, y)$ of morphisms,
  we write $f : x \Rightarrow y$ for $f \in C(x, y)$,
- a composition operation: given

  $$f : x \Rightarrow y \qquad \text{and} \qquad g : y \Rightarrow z$$

  we have

  $$f \cdot g \quad : \quad x \quad \Rightarrow \quad z$$

- an identity morphism for every object $x \in C$

  $$1_x \quad : \quad x \quad \Rightarrow \quad x$$

such that

Categories

A **category** C consists of

- a set $\mathrm{Ob}(C)$ of objects,
- for every objects $x, y \in C$, a set $C(x, y)$ of morphisms, we write $f : x \Rightarrow y$ for $f \in C(x, y)$,
- a composition operation: given

  $$f : x \Rightarrow y \qquad \text{and} \qquad g : y \Rightarrow z$$

  we have

  $$f \cdot g \quad : \quad x \quad \Rightarrow \quad z$$

- an identity morphism for every object $x \in C$

  $$1_x \quad : \quad x \quad \Rightarrow \quad x$$

such that

- composition is associative: for $f : x \Rightarrow y$, $g : y \Rightarrow z$, $h : z \Rightarrow w$,

  $$(f \cdot g) \cdot h \quad = \quad f \cdot (g \cdot h)$$

A **category** C consists of

- a set $\mathrm{Ob}(C)$ of objects,
- for every objects $x, y \in C$, a set $C(x, y)$ of morphisms, we write $f : x \Rightarrow y$ for $f \in C(x, y)$,
- a composition operation: given

$$f : x \Rightarrow y \qquad \text{and} \qquad g : y \Rightarrow z$$

  we have

$$f \cdot g \quad : \quad x \quad \Rightarrow \quad z$$

- an identity morphism for every object $x \in C$

$$1_x \quad : \quad x \quad \Rightarrow \quad x$$

such that

- composition is associative: for $f : x \Rightarrow y$, $g : y \Rightarrow z$, $h : z \Rightarrow w$,

$$(f \cdot g) \cdot h \quad = \quad f \cdot (g \cdot h)$$

- identities are neutral elements: for $f : x \Rightarrow y$,

$$1_x \cdot f \quad = \quad f \quad = \quad f \cdot 1_y \, .$$

Categories

In a category, we have typed morphisms

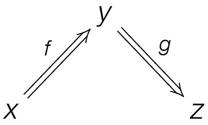
$$x \xRightarrow{f} y$$

Categories

In a category, we have typed morphisms

$$x \xRightarrow{f} y$$

that we can compose

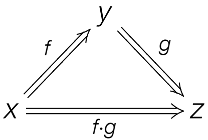


Categories

In a category, we have typed morphisms

$$x \overset{f}{\Longrightarrow} y$$

that we can compose

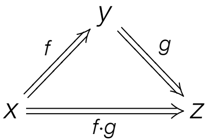


Categories

In a category, we have typed morphisms

$$x \xRightarrow{f} y$$

that we can compose



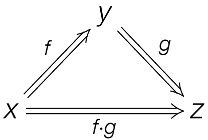
in an associative way

Categories

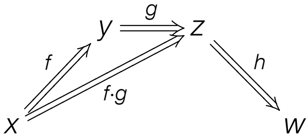
In a category, we have typed morphisms

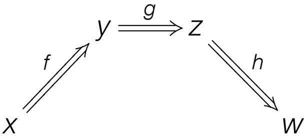
$$x \xRightarrow{f} y$$

that we can compose



in an associative way

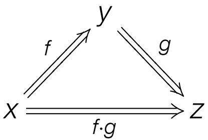




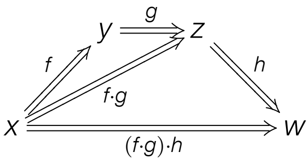
In a category, we have typed morphisms

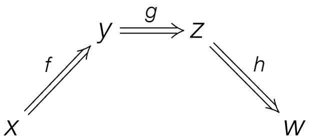
$$x \xRightarrow{f} y$$

that we can compose



in an associative way



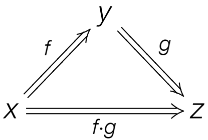


Categories

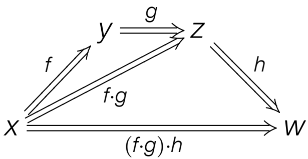
In a category, we have typed morphisms

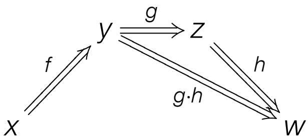
$$x \xRightarrow{f} y$$

that we can compose



in an associative way



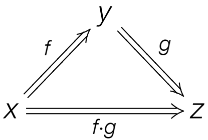


Categories

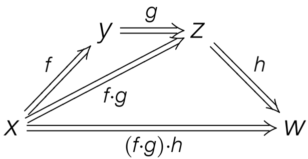
In a category, we have typed morphisms

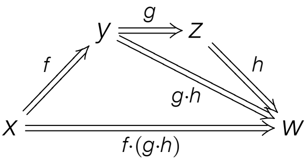
$$x \xRightarrow{f} y$$

that we can compose



in an associative way



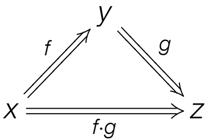


Categories

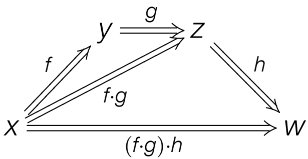
In a category, we have typed morphisms

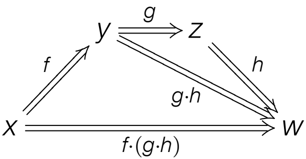
$$x \xRightarrow{f} y$$

that we can compose



in an associative way





and we have identities $x \xRightarrow{1_x} x$.

Categories

Examples

- **Set**: sets as objects / functions as morphisms
- **Top**: topological spaces / continuous functions
- **Mon**: monoids / morphisms of monoids
- **Gph**: graphs / morphisms of graphs
- **Cat**: categories / functors
- etc.

The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given two objects u, v, we can concatenate them

$$u \otimes v \quad = \quad uv$$

graphically,

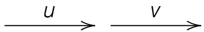
The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given two objects u, v, we can concatenate them

$$u \otimes v \quad = \quad uv$$

graphically,



The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given two objects u, v, we can concatenate them

$$u \otimes v \quad = \quad uv$$

graphically,

$$\xrightarrow{u\otimes v} \quad = \quad \xrightarrow{u}\ \xrightarrow{v}$$

The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given two objects u, v, we can concatenate them

$$u \otimes v \quad = \quad uv$$

  graphically,

$$\xrightarrow{\quad u\otimes v \quad} \quad = \quad \xrightarrow{\quad u \quad}\xrightarrow{\quad v \quad}$$

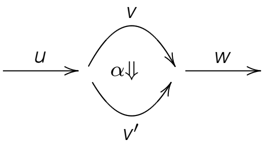
- there is an empty word 1,

The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given a rewriting step $u\alpha w$ and objects u', w', we can put the step "in context":

$$u' \otimes (u\alpha w) \otimes w' \quad = \quad (u'u)\alpha(ww')$$

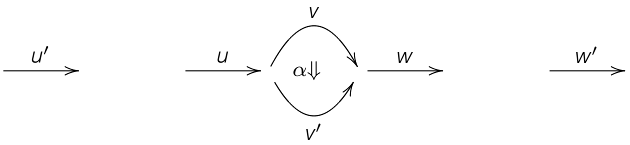


The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given a rewriting step $u\alpha w$ and objects u', w', we can put the step "in context":

$$u' \otimes (u\alpha w) \otimes w' \quad = \quad (u'u)\alpha(ww')$$

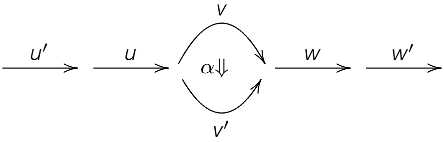


The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given a rewriting step $u\alpha w$ and objects u', w', we can put the step "in context":

$$u' \otimes (u\alpha w) \otimes w' \quad = \quad (u'u)\alpha(ww')$$



The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given a rewriting step $u\alpha w$ and objects u', w', we can put the step "in context":

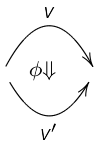
$$u' \otimes (u\alpha w) \otimes w' \quad = \quad (u'u)\alpha(ww')$$

which extends to rewriting paths

$$\phi = u_1\alpha_1 w_1 \;\cdot\; \ldots \;\cdot\; u_n\alpha_n w_n$$

by

$$u'\phi w' \quad = \quad (u'u_1)\alpha_1(w_1 w') \;\cdot\; \ldots \;\cdot\; (u'u_n)\alpha_n(w_n w')$$



The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given a rewriting step $u\alpha w$ and objects u', w', we can put the step "in context":

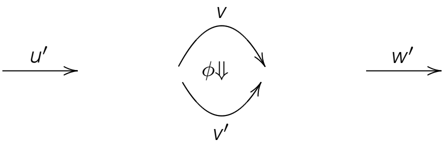
$$u' \otimes (u\alpha w) \otimes w' \quad = \quad (u'u)\alpha(ww')$$

which extends to rewriting paths

$$\phi = u_1\alpha_1 w_1 \;\cdot\; \ldots \;\cdot\; u_n\alpha_n w_n$$

by

$$u'\phi w' \quad = \quad (u'u_1)\alpha_1(w_1w') \;\cdot\; \ldots \;\cdot\; (u'u_n)\alpha_n(w_nw')$$



The category of rewriting paths

The category G^* of rewriting paths has more structure:

- given a rewriting step $u\alpha w$ and objects u', w', we can put the step "in context":

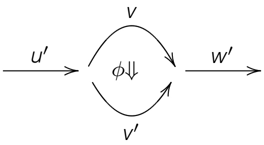
$$u' \otimes (u\alpha w) \otimes w' \quad = \quad (u'u)\alpha(ww')$$

  which extends to rewriting paths

$$\phi = u_1\alpha_1 w_1 \ \cdot \ \ldots \ \cdot \ u_n\alpha_n w_n$$

  by

$$u'\phi w' \quad = \quad (u'u_1)\alpha_1(w_1 w') \ \cdot \ \ldots \ \cdot \ (u'u_n)\alpha_n(w_n w')$$



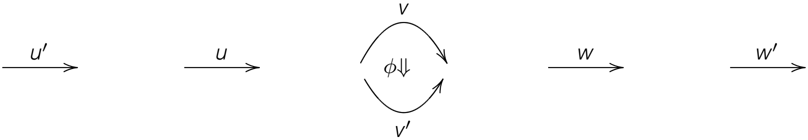
The category of rewriting paths

The category G^* of rewriting paths has more structure:

- the operation $\otimes$ is "associative":

$$u' \otimes (u \otimes \phi \otimes w) \otimes w' \quad = \quad (u' \otimes u) \otimes \phi \otimes (w \otimes w')$$

graphically,



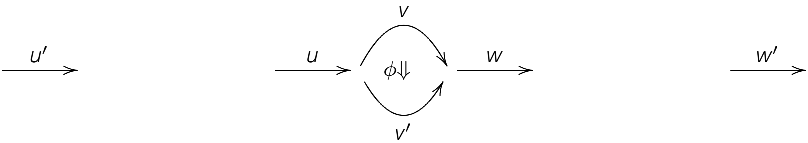
The category of rewriting paths

The category G^* of rewriting paths has more structure:

- the operation $\otimes$ is “associative”:

$$u' \otimes (u \otimes \phi \otimes w) \otimes w' \quad = \quad (u' \otimes u) \otimes \phi \otimes (w \otimes w')$$

graphically,



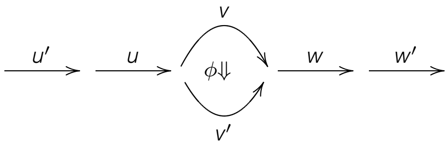
The category of rewriting paths

The category G^* of rewriting paths has more structure:

- the operation $\otimes$ is "associative":

$$u' \otimes (u \otimes \phi \otimes w) \otimes w' \quad = \quad (u' \otimes u) \otimes \phi \otimes (w \otimes w')$$

  graphically,



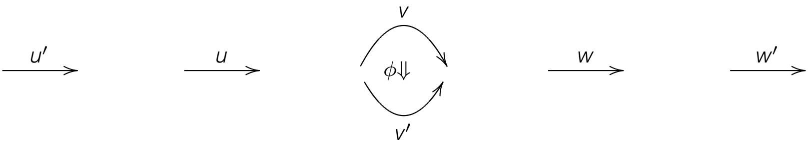
The category of rewriting paths

The category G^* of rewriting paths has more structure:

- the operation $\otimes$ is "associative":

$$u' \otimes (u \otimes \phi \otimes w) \otimes w' \quad = \quad (u' \otimes u) \otimes \phi \otimes (w \otimes w')$$

graphically,



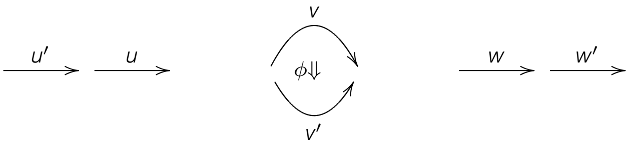
The category of rewriting paths

The category G^* of rewriting paths has more structure:

- the operation $\otimes$ is “associative”:

$$u' \otimes (u \otimes \phi \otimes w) \otimes w' \quad = \quad (u' \otimes u) \otimes \phi \otimes (w \otimes w')$$

graphically,



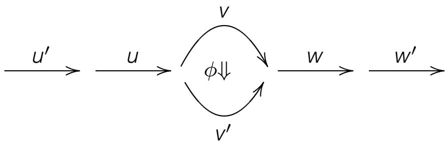
The category of rewriting paths

The category G^* of rewriting paths has more structure:

- the operation $\otimes$ is "associative":

$$u' \otimes (u \otimes \phi \otimes w) \otimes w' \quad = \quad (u' \otimes u) \otimes \phi \otimes (w \otimes w')$$

graphically,



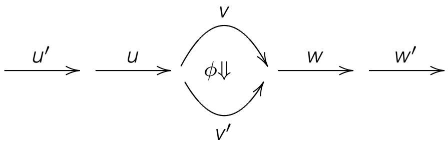
The category of rewriting paths

The category G^* of rewriting paths has more structure:

- the operation $\otimes$ is "associative":

$$u' \otimes (u \otimes \phi \otimes w) \otimes w' \quad = \quad (u' \otimes u) \otimes \phi \otimes (w \otimes w')$$

  graphically,



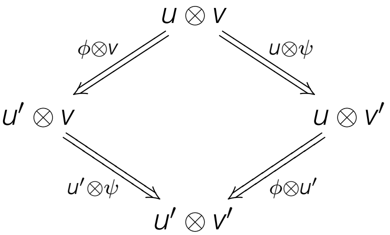
- and the empty word is a neutral element.

The category of rewriting paths

This operation $\otimes$ satisfies the **exchange law**:

$$(\phi \otimes v) \cdot (u' \otimes \psi) \quad = \quad (u \otimes \psi) \cdot (\phi \otimes v')$$

Graphically,

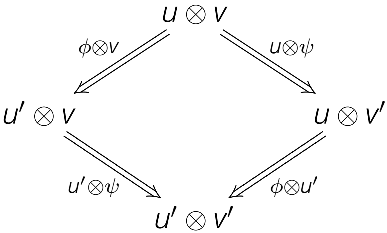


The category of rewriting paths

This operation $\otimes$ satisfies the **exchange law**:

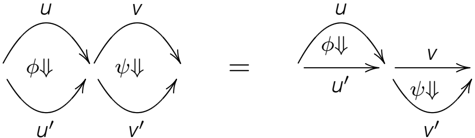
$$(\phi \otimes v) \ \cdot \ (u' \otimes \psi) \quad = \quad (u \otimes \psi) \ \cdot \ (\phi \otimes v')$$

Graphically,



We can thus define "rewriting by ϕ and ψ in parallel":

$$\phi \otimes \psi \quad = \quad (\phi \otimes v) \ \cdot \ (u' \otimes \psi)$$

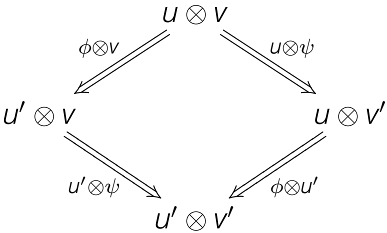


The category of rewriting paths

This operation $\otimes$ satisfies the **exchange law**:

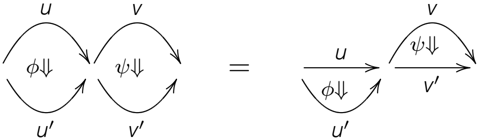
$$(\phi \otimes v) \ \cdot \ (u' \otimes \psi) \quad = \quad (u \otimes \psi) \ \cdot \ (\phi \otimes v')$$

Graphically,



We can thus define "rewriting by ϕ and ψ in parallel":

$$\phi \otimes \psi \quad = \quad (\phi \otimes v) \ \cdot \ (u' \otimes \psi)$$

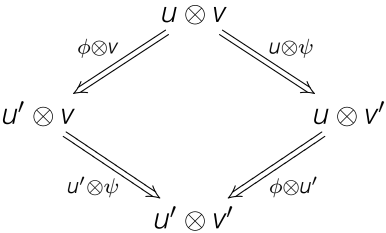


The category of rewriting paths

This operation $\otimes$ satisfies the **exchange law**:

$$(\phi \otimes v) \;\cdot\; (u' \otimes \psi) \quad = \quad (u \otimes \psi) \;\cdot\; (\phi \otimes v')$$

Graphically,



We can thus define “rewriting by ϕ and ψ in parallel”:

$$\phi \otimes \psi \quad = \quad (\phi \otimes v) \;\cdot\; (u' \otimes \psi)$$

and we can recover “context extension” from this operation:

$$u \otimes \phi \otimes v \quad = \quad \mathrm{id}_u \otimes \phi \otimes \mathrm{id}_v$$

The category of rewriting paths

To sum up, G^* is a **monoidal category**.

Monoidal categories

A (strict) **monoidal category** $(C, \otimes, 1)$ is

- a category C
- $(C, \otimes, 1)$ is a monoid
- given morphisms

$$f : x \to x' \qquad\qquad g : y \to y'$$

we have a morphism

$$f \otimes g \quad : \quad x \otimes x' \quad \to \quad y \otimes y'$$

and this operation is associative and admits id_1 as unit:

$$(f \otimes g) \otimes h = f \otimes (g \otimes h) \qquad\qquad \mathrm{id}_1 \otimes f = f = f \otimes \mathrm{id}_1$$

this operation is compatible with composition

$$(f \cdot f') \otimes (g \cdot g') \quad = \quad (f \otimes g) \cdot (f' \otimes g')$$

and units.

The simplicial category

The **simplicial category** $\triangle$ whose

- objects are natural numbers $n \in \mathbb{N}$,
- a morphism

$$f \quad : \quad m \quad \to \quad n$$

  is an increasing function

$$f \quad : \quad \{0, \ldots, m-1\} \quad \to \quad \{0, \ldots, n-1\}$$

- composition and identities are the usual ones.

The simplicial category

The **simplicial category** $\triangle$ whose

- objects are natural numbers $n \in \mathbb{N}$,
- a morphism

$$f \quad : \quad m \quad \to \quad n$$

  is an increasing function

$$f \quad : \quad \{0, \ldots, m-1\} \quad \to \quad \{0, \ldots, n-1\}$$

- composition and identities are the usual ones.

Exercise

Show that this category is monoidal with $\otimes$ defined on objects by

$$m \otimes n \quad = \quad m + n$$

The simplicial category

Correction
Given

$$f : m \to m' \qquad\qquad g : n \to n'$$

we define the function

$$\begin{aligned} f \otimes g : \quad \{0, \ldots, m{+}n{-}1\} \quad &\to \quad \{0, \ldots, m'{+}n'{-}1\} \\ i \quad &\mapsto \quad \begin{cases} f(i) & \text{if } 0 \leq i < m \\ m' + (g(i-m)) & \text{if } m \leq i < m+n \end{cases} \end{aligned}$$

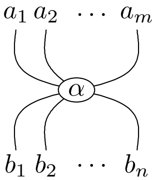
String diagrams

The morphisms of G^* admit a representation as **string diagrams**.

The idea is that a morphism generator

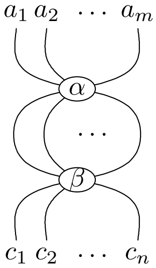
$$\alpha \quad : \quad a_1 \ldots a_m \quad \Rightarrow \quad b_1 \ldots b_n$$

can be pictured as a "gate"



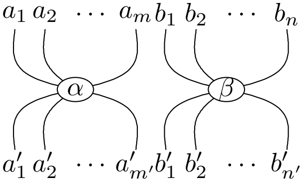
Composition is vertical juxtaposition and linking:

$\alpha \cdot \beta \quad =$



Tensor product is horizontal juxtaposition:

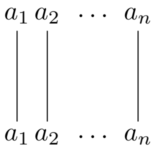
$\alpha \otimes \beta \quad =$



String diagrams

Identities are wires:

$\mathrm{id}_{a_1 \otimes a_2 \otimes \ldots \otimes a_n} \quad =$



String diagrams

Theorem (Joyal-Street'91)

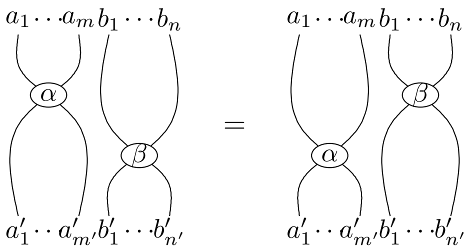
Diagrams up to deformations correspond precisely to morphisms.

String diagrams

Theorem (Joyal-Street'91)

Diagrams up to deformations correspond precisely to morphisms.

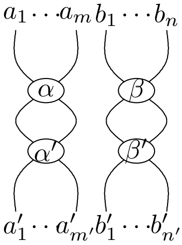
A deformation is for instance



Theorem (Joyal-Street'91)

Diagrams up to deformations correspond precisely to morphisms.

The interpretation of diagrams is unambiguous:



$$(\alpha \cdot \alpha') \otimes (\beta \cdot \beta') \quad = \quad (\alpha \otimes \beta) \cdot (\alpha' \otimes \beta')$$

Proposition

The monoidal category G^* is the **free monoidal category** containing

- the elements of G_1 as objects,
- the elements of G_2 as morphisms.

Presentations of monoidal categories

A **presentation** P of a monoidal category is

$$\langle G \mid R \rangle$$

where

- *generators*: $G = \langle G_1 \mid G_2 \rangle$ is a presentation of a monoid,
- *relations*: $R \subseteq G^* \times G^*$ consists of pairs of morphisms with same source and same target.

Presentations of monoidal categories

A **presentation** P of a monoidal category is

$$\langle G \mid R \rangle$$

where

- *generators*: $G = \langle G_1 \mid G_2 \rangle$ is a presentation of a monoid,
- *relations*: $R \subseteq G^* \times G^*$ consists of pairs of morphisms with same source and same target.

The monoidal category **presented** by P is

$$G^* / {\approx^R}$$

where $\approx^R$ is the congruence generated by R.

Presentations of monoidal categories

A **presentation** P of a monoidal category is

$$\langle G \mid R \rangle$$

where

- *generators*: $G = \langle G_1 \mid G_2 \rangle$ is a presentation of a monoid,
- *relations*: $R \subseteq G^* \times G^*$ consists of pairs of morphisms with same source and same target.

The monoidal category **presented** by P is

$$G^* / {\approx^R}$$

where $\approx^R$ is the congruence generated by R.

A monoidal category C is presented by P when

$$C \quad \cong \quad G^* / {\approx^R} .$$

A presentation for △

Consider the presentation $\langle G \mid R \rangle$ where

- $G_1 = \{a\}$

A presentation for $\triangle$

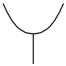
Consider the presentation $\langle G \mid R \rangle$ where

- $G_1 = \{a\}$
- $G_2 = \{\mu : aa \Rightarrow a, \eta : 1 \Rightarrow a\}$

A presentation for $\triangle$

Consider the presentation $\langle G \mid R \rangle$ where

- $G_1 = \{a\}$
- $G_2 = \{\mu : aa \Rightarrow a, \eta : 1 \Rightarrow a\}$



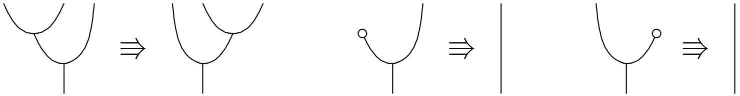
A presentation for △

Consider the presentation $\langle G \mid R \rangle$ where

- $G_1 = \{a\}$
- $G_2 = \{\mu : aa \Rightarrow a, \eta : 1 \Rightarrow a\}$

- relations are

$$(\mu \otimes a) \cdot \mu \Rrightarrow (a \otimes \mu) \cdot \mu \quad (\eta \otimes a) \cdot \mu \Rrightarrow \mathrm{id}_a \quad (a \otimes \eta) \cdot \mu \Rrightarrow \mathrm{id}_a$$



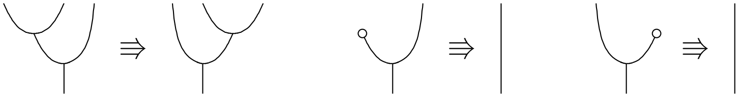
A presentation for △

Consider the presentation $\langle G \mid R \rangle$ where

- $G_1 = \{a\}$
- $G_2 = \{\mu : aa \Rightarrow a, \eta : 1 \Rightarrow a\}$

- relations are

$$(\mu \otimes a) \cdot \mu \Rrightarrow (a \otimes \mu) \cdot \mu \quad (\eta \otimes a) \cdot \mu \Rrightarrow \mathrm{id}_a \quad (a \otimes \eta) \cdot \mu \Rrightarrow \mathrm{id}_a$$



Claim: this is a presentation of △.

A presentation for $\triangle$

The idea to show that this is a presentation for $\triangle$ is a before:

1. show that this presentation is confluent:
   terminating + confluent critical branchings

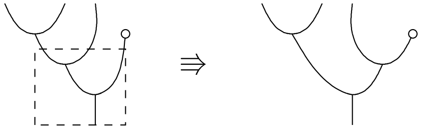
2. show that normal forms are in bijection with morphisms of $\triangle$.

Let’s study critical branchings

(graphically, from now on)

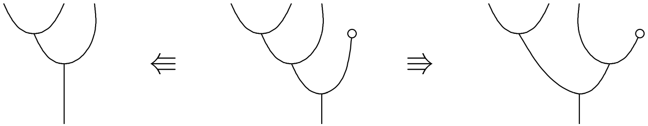
Rewriting steps

A **rewriting step** is a rewriting rule “in context”:



Branchings

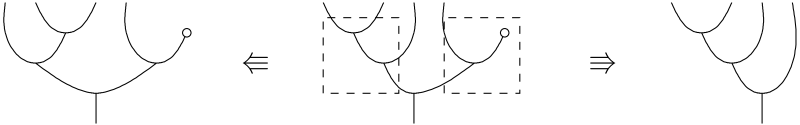
A **branching** is a pair of rewriting steps from the same diagram:



Critical branchings

A branching is **non-critical** when

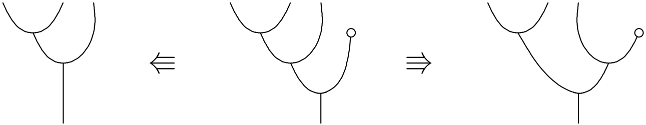
- it consists in two *independent* applications of rules (rules do not share 1-generators)



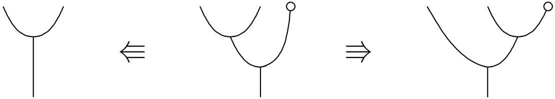
Critical branchings

A branching is **non-critical** when

- is it not *minimal*
  (can be obtained by putting another branching in context)



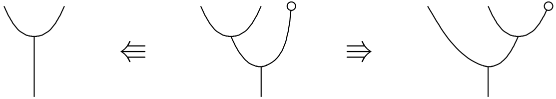
can be obtained from



Critical branchings

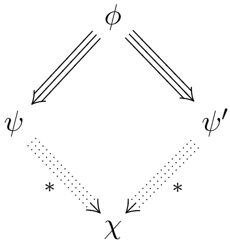
A branching is **critical** when it is not non-critical:

- branches are not independent: left members of rules overlap
- it is minimal: all the 1-generators are used



Lemma

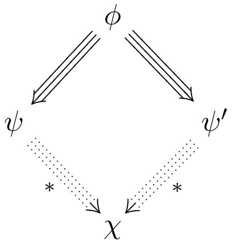
A 2-dimensional rewriting system is locally confluent iff all critical branchings are confluent.



Critical pairs lemma

Lemma

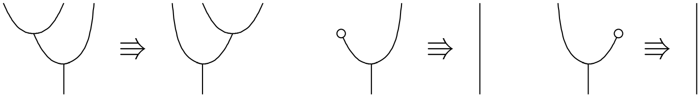
A 2-dimensional rewriting system is locally confluent iff all critical branchings are confluent.



In particular, a terminating 2-dimensional rewriting system with confluent critical branchings is confluent.

Exercise

Consider the previous rewriting system



We assume that it is terminating.

1. Show that it is confluent.
2. What do the normal forms look like?
3. Define an interpretation of generators in $\triangle$.
4. Show that normal forms

$$\phi \quad : \quad a^m \quad \to \quad a^n$$

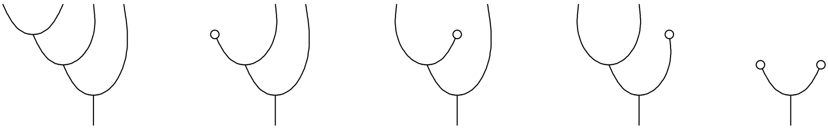
   are in bijection with functions

$$f \quad : \quad \{0, \ldots, m-1\} \quad \to \quad \{0, \ldots, n-1\}$$

5. Deduce that we have a presentation of $\triangle$.

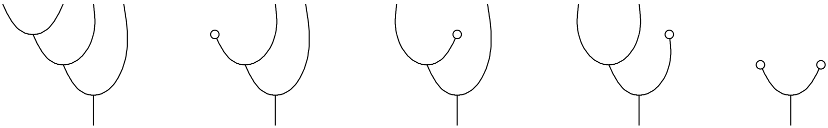
Correction

1. The critical pairs are confluent:



Correction

1. The critical pairs are confluent:



2. The *right comb* $\kappa_n : a^n \to a$ is

Normal forms are tensor products of right combs.

Correction

3. The interpretation of generators into $\triangle$ is given as follows.
   - We interpret

$$a \qquad \text{as} \qquad 1$$

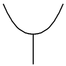
     thus a^n is interpreted as n.

Correction

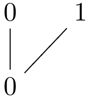
3. The interpretation of generators into $\triangle$ is given as follows.
   - We interpret

     $$a \qquad \text{as} \qquad 1$$

     thus a^n is interpreted as n.
   - We interpret



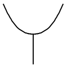
as



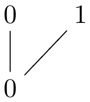
3. The interpretation of generators into $\triangle$ is given as follows.
   - We interpret

     $$a \quad \text{as} \quad 1$$

     thus a^n is interpreted as n.
   - We interpret

     as

   - We interpret

     as

     $$0$$

Correction

4. The interpretation of the normal form

$$\kappa_{n_1} \otimes \kappa_{n_2} \otimes \ldots \otimes \kappa_{n_k}$$

is a function

$$f \quad : \quad n_1 + n_2 + \ldots + n_k \quad \to \quad k$$

such that for $0 \leq i < k$,

$$|f^{-1}(i)| \quad = \quad n_i$$

Correction

4. The interpretation of the normal form

$$\kappa_{n_1} \otimes \kappa_{n_2} \otimes \ldots \otimes \kappa_{n_k}$$

is a function

$$f \quad : \quad n_1 + n_2 + \ldots + n_k \quad \to \quad k$$

such that for $0 \leq i < k$,

$$|f^{-1}(i)| \quad = \quad n_i$$

Every increasing function can be obtained in this way, and the sequence $(n_i)_{1 \leq i \leq k}$ determines uniquely the function.

Correction

4. The interpretation of the normal form

$$\kappa_{n_1} \otimes \kappa_{n_2} \otimes \ldots \otimes \kappa_{n_k}$$

is a function

$$f \quad : \quad n_1 + n_2 + \ldots + n_k \quad \to \quad k$$

such that for $0 \leq i < k$,

$$|f^{-1}(i)| \quad = \quad n_i$$

Every increasing function can be obtained in this way, and the sequence $(n_i)_{1 \leq i \leq k}$ determines uniquely the function.

5. We thus have a presentation of $\triangle$.

The category $\mathbf{B}$

The category $\mathbf{B}$ has

- objects: $\mathbb{N}$
- a morphism

$$f \quad : \quad m \quad \to \quad n$$

  is a bijection

$$f \quad : \quad \{0, \ldots, m-1\} \quad \to \quad \{0, \ldots, n-1\}$$

- compositions and identities are as usual,
- tensor product is as in the case of $\triangle$.

Exercise

1. Propose some generators for this category.
2. Propose some relations for this category.
3. What are the critical pairs?
4. Show local confluence.
5. Assuming termination, show that this is a presentation of **B**.

Question

Does a finite rewriting system necessarily has a finite number of critical pairs?

An example of termination.

Showing termination

A poset is **well-founded** if every decreasing sequence is eventually stationary (e.g. $\mathbb{N}$).

Showing termination

A poset is **well-founded** if every decreasing sequence is eventually stationary (e.g. $\mathbb{N}$).

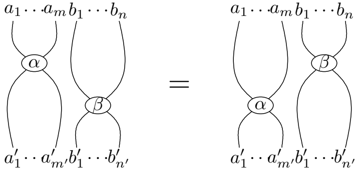
In order to show that a rewriting system is terminating, we can interpret all the diagrams in a well-founded poset, in such a way that all rules are strictly decreasing.

Showing termination

A poset is **well-founded** if every decreasing sequence is eventually stationary (e.g. $\mathbb{N}$).

In order to show that a rewriting system is terminating, we can interpret all the diagrams in a well-founded poset, in such a way that all rules are strictly decreasing.

Note that this interpretation should be compatible with the axioms of monoidal categories:



Counting generators

For instance, we consider $(\mathbb{N}, \leq)$ and associate to each diagram the number of generators occurring in it.

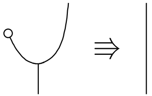
The rules

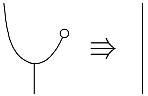
are strictly decreasing.

Counting generators

For instance, we consider $(\mathbb{N}, \leq)$ and associate to each diagram the number of generators occurring in it.

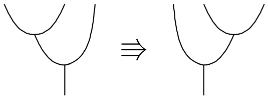
The rules





are strictly decreasing.

But not the rule



Multiple well-founded posets

Rewriting preserves typing:

$$(f : m \to n) \quad \Rrightarrow \quad (g : m \to n)$$

We can therefore have a different well-founded poset for each pair of objects!

Multiple well-founded posets

Rewriting preserves typing:

$$(f : m \to n) \quad \Rrightarrow \quad (g : m \to n)$$

We can therefore have a different well-founded poset for each pair of objects!

Lafont had the idea of interpreting morphisms

$$f : m \to n$$

as functions in

$$\mathbb{N}_*^m \to \mathbb{N}_*^n$$

equipped with a particular well-founded order.

Multiple well-founded posets

Given $n \in \mathbb{N}$, we consider

$$\mathbb{N}_*^n$$

(where $N_* = \mathbb{N} \setminus \{0\}$) equipped with the product order:

$$(x_1, \ldots, x_n) \quad \leq \quad (x'_1, \ldots, x'_n)$$

iff for every $1 \leq i \leq n$

$$x_i \quad \leq \quad x'_i \,.$$

Multiple well-founded posets

Given $n \in \mathbb{N}$, we consider

$$\mathbb{N}_*^n$$

(where $N_* = \mathbb{N} \setminus \{0\}$) equipped with the product order:

$$(x_1, \ldots, x_n) \quad \leq \quad (x'_1, \ldots, x'_n)$$

iff for every $1 \leq i \leq n$

$$x_i \quad \leq \quad x'_i \, .$$

Lemma

This is a well-founded poset.

Multiple well-founded posets

Given objects m, n we consider strictly increasing functions

$$\mathbb{N}_*^m \to \mathbb{N}_*^n$$

ordered by

$$f \quad < \quad f'$$

whenever for every $(x_1, \ldots, x_m)$

$$f(x_1, \ldots, x_n) \quad < \quad f'(x_1, \ldots, x_n)\,.$$

Multiple well-founded posets

Given objects m, n we consider strictly increasing functions

$$\mathbb{N}_*^m \to \mathbb{N}_*^n$$

ordered by

$$f \quad < \quad f'$$

whenever for every $(x_1, \ldots, x_m)$

$$f(x_1, \ldots, x_n) \quad < \quad f'(x_1, \ldots, x_n)\,.$$

Lemma

This is a well-founded poset.

Multiple well-founded posets

We have a monoidal category where

- an objects is an integer
- a morphism

$$f \quad : \quad m \quad \to \quad n$$

  is a strictly increasing function

$$f \quad : \quad \mathbb{N}_*^m \quad \to \quad \mathbb{N}_*^n$$

and moreover the relations $<$ are compatible with composition and tensor.

Multiple well-founded posets

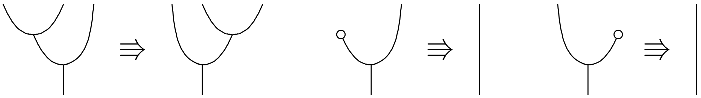
In order to provide an interpretation of every diagram

$$m \to n$$

it is sufficient to interpret generators (and extend it in a way compatible with composition and tensor).

Exercise

Show that the rewriting system



is terminating.

Exercise

Show that the presentation of **B** is terminating.