

# **CRITICAL PAIRS IN 2-DIMENSIONAL REWRITING SYSTEMS**

SAMUEL MIMRAM

**OPERADS AND REWRITING**

3 NOVEMBER 2011

# TOWARDS HIGHER-DIMENSIONAL REWRITING THEORY

**Rewriting theory** has proven to be very useful to study

- ▶ monoids (and groups)
- ▶ term algebras

# TOWARDS HIGHER-DIMENSIONAL REWRITING THEORY

**Rewriting theory** has proven to be very useful to study

- ▶ monoids (and groups)
- ▶ term algebras
- ▶  ***$n$ -categories?***

Can we generalize it  
to higher dimensions?

# TOWARDS HIGHER-DIMENSIONAL REWRITING THEORY

**Rewriting theory** has proven to be very useful to study

- ▶ monoids (and groups)
- ▶ term algebras
- ▶ *n*-categories?

Can we generalize it  
to higher dimensions?

In this talk, I will be interested in extending the procedures of  
**unification** in *higher dimensions*.

# TOWARDS HIGHER-DIMENSIONAL REWRITING THEORY

**Rewriting theory** has proven to be very useful to study

- ▶ monoids (and groups)
- ▶ term algebras
- ▶ ***n*-categories?**

Can we generalize it  
to higher dimensions?

In this talk, I will be interested in extending the procedures of  
**unification** in **dimension 2**.

# REWRITING SYSTEMS

# REWRITING SYSTEMS

A **rewriting system** consists of

- ▶ a set of *terms* generated by a free construction:
  - ▶ free monoid: *string rewriting systems*
  - ▶ free term algebra: *term rewriting systems*
- ▶ a set of *rewriting rules*:  $r : t \rightarrow u$

## Example

$$\Sigma = \{a, b\}$$

$$\text{terms} = \Sigma^*$$

$$\text{rules} = \{ba \rightarrow ab\}$$

# REWRITING SYSTEMS

A **rewriting system** consists of

- ▶ a set of *terms* generated by a free construction:
  - ▶ free monoid: *string rewriting systems*
  - ▶ free term algebra: *term rewriting systems*
- ▶ a set of *rewriting rules*:  $r : t \rightarrow u$

A term  $t$  **rewrites** to a term  $t'$  when there exists

- ▶ a rule  $r : u \rightarrow u'$
- ▶ a context  $C$  such that  $t = C[u]$  and  $t' = C[u']$

## Example

$$\Sigma = \{a, b\}$$

$$\text{terms} = \Sigma^*$$

$$\text{rules} = \{ba \rightarrow ab\}$$

$$a**ba**ab \xrightarrow{a**a**ab} aa**ab**ab$$

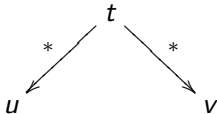
# CONVERGENT REWRITING SYSTEMS

- ▶ A rewriting system can be **terminating** when there is no infinite reduction path



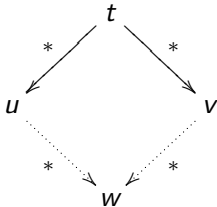
# CONVERGENT REWRITING SYSTEMS

- ▶ A rewriting system can be **terminating**
- ▶ A rewriting can be **confluent** when



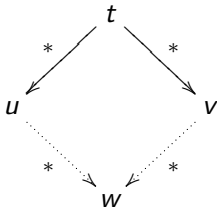
# CONVERGENT REWRITING SYSTEMS

- ▶ A rewriting system can be **terminating**
- ▶ A rewriting can be **confluent** when



# CONVERGENT REWRITING SYSTEMS

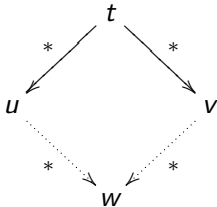
- ▶ A rewriting system can be **terminating**
- ▶ A rewriting can be **confluent** when



- ▶ A rewriting system is **convergent** when both terminating and confluent

# CONVERGENT REWRITING SYSTEMS

- ▶ A rewriting system can be **terminating**
- ▶ A rewriting can be **confluent** when



- ▶ A rewriting system is **convergent** when both terminating and confluent

In a convergent rewriting system, every term has a **normal form**:  
**canonical representative** of terms modulo rewriting.

Why  
are those properties  
interesting?

# PRESENTATIONS OF MONOIDS

A **presentation**

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

of a monoid  $M$  consists of

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

such that

$$M \cong G^* / \equiv_R$$

**Example**

- ▶  $\mathbb{N} \cong \langle a \mid \rangle$
- ▶  $\mathbb{N}/2\mathbb{N} \cong \langle a \mid aa = 1 \rangle$
- ▶  $\mathbb{N} \times \mathbb{N} \cong \langle a, b \mid ba = ab \rangle$
- ▶  $\mathfrak{S}_n \cong \langle \sigma_1, \dots, \sigma_n \mid \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}, \sigma_i^2 = 1, \sigma_i \sigma_j = \sigma_j \sigma_i \rangle$
- ▶ ...

# PRESENTATIONS OF MONOIDS

How do we show that  $M \cong \langle G \mid R \rangle$  i.e.  $M \cong G^* / \equiv_R$  ?

# PRESENTATIONS OF MONOIDS

How do we show that  $M \cong \langle G \mid R \rangle$  i.e.  $M \cong G^* / \equiv_R$  ?

1. Orient  $R$  to get a string rewriting system.
2. Show that the rewriting system is terminating.
3. Show that the rewriting system is confluent.
4. Show that the normal forms are in bijection with  $M$ .

# PRESENTATIONS OF MONOIDS

How do we show that  $M \cong \langle G \mid R \rangle$  i.e.  $M \cong G^* / \equiv_R$  ?

1. Orient  $R$  to get a string rewriting system.
2. Show that the rewriting system is terminating.
3. Show that the rewriting system is confluent.
4. Show that the normal forms are in bijection with  $M$ .

**Example**  $\mathbb{N} \times (\mathbb{N}/2\mathbb{N}) \stackrel{?}{\cong} \langle a, b \mid ba = ab, bb = 1 \rangle$

# PRESENTATIONS OF MONOIDS

How do we show that  $M \cong \langle G \mid R \rangle$  i.e.  $M \cong G^* / \equiv_R$  ?

1. Orient  $R$  to get a string rewriting system.
2. Show that the rewriting system is terminating.
3. Show that the rewriting system is confluent.
4. Show that the normal forms are in bijection with  $M$ .

**Example**  $\mathbb{N} \times (\mathbb{N}/2\mathbb{N}) \stackrel{?}{\cong} \langle a, b \mid ba \rightarrow ab, bb \rightarrow 1 \rangle$

# PRESENTATIONS OF MONOIDS

How do we show that  $M \cong \langle G \mid R \rangle$  i.e.  $M \cong G^* / \equiv_R$  ?

1. Orient  $R$  to get a string rewriting system.
2. Show that the rewriting system is terminating.
3. Show that the rewriting system is confluent.
4. Show that the normal forms are in bijection with  $M$ .

**Example**  $\mathbb{N} \times (\mathbb{N}/2\mathbb{N}) \stackrel{?}{\cong} \langle a, b \mid ba \rightarrow ab, bb \rightarrow 1 \rangle$

Normal forms are:

$a^n$       and       $a^n b$

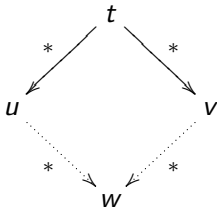
They are in bijection with  $\mathbb{N} \times (\mathbb{N}/2\mathbb{N})!$

How do we show  
that a rewriting system  
is confluent?

# CRITICAL PAIRS

Given a terminating rewriting system the following are equivalent:

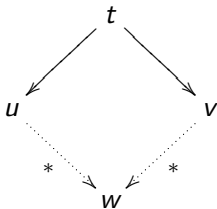
1. the rewriting system is **confluent**



# CRITICAL PAIRS

Given a *terminating* rewriting system the following are equivalent:

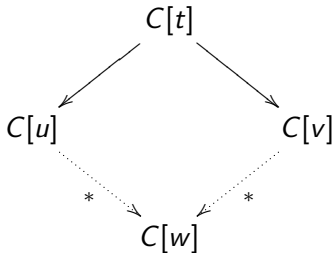
1. the rewriting system is **confluent**
2. the rewriting system is **locally confluent**



# CRITICAL PAIRS

Given a terminating rewriting system the following are equivalent:

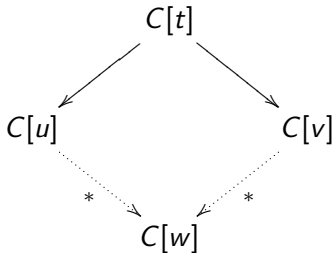
1. the rewriting system is **confluent**
2. the rewriting system is **locally confluent**



# CRITICAL PAIRS

Given a terminating rewriting system the following are equivalent:

1. the rewriting system is **confluent**
2. the rewriting system is **locally confluent**



3. all the **critical pairs are joinable**  
i.e. the property above is satisfied for all minimal  $t$

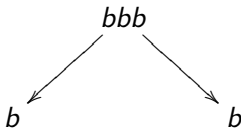
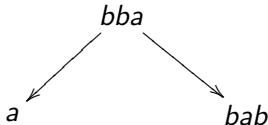
# CRITICAL PAIRS

Given a terminating rewriting system the following are equivalent:

1. the rewriting system is **confluent**
2. the rewriting system is **locally confluent**
3. all the **critical pairs are joinable**  
i.e. the property above is satisfied for all minimal  $t$

**Example**  $\mathbb{N} \times (\mathbb{N}/2\mathbb{N}) \stackrel{?}{\cong} \langle a, b \mid ba \rightarrow ab, bb \rightarrow 1 \rangle$

Critical pairs are:



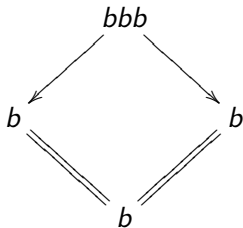
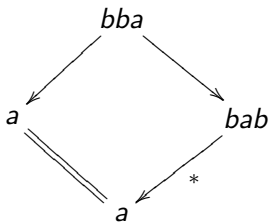
# CRITICAL PAIRS

Given a terminating rewriting system the following are equivalent:

1. the rewriting system is **confluent**
2. the rewriting system is **locally confluent**
3. all the **critical pairs are joinable**  
i.e. the property above is satisfied for all minimal  $t$

**Example**  $\mathbb{N} \times (\mathbb{N}/2\mathbb{N}) \stackrel{?}{\cong} \langle a, b \mid ba \rightarrow ab, bb \rightarrow 1 \rangle$

Critical pairs are joinable:



string rs = presentation of a monoid  
term rs = presentation of ?

# TERM REWRITING SYSTEMS

- ▶ A **signature**  $(\Sigma, \alpha)$  consists of
  - ▶ a set  $\Sigma$  of *generators*
  - ▶ an *arity* function  $\alpha : \Sigma \rightarrow \mathbb{N}$
- ▶ Terms are elements of the free algebra  $\Sigma^*$  over this signature

## Example

The TRS of commutative monoids:  $\Sigma = \{m : 2, e : 0\}$

$$R = \left\{ \begin{array}{l} \alpha : m(m(x, y), z) \rightarrow m(x, m(y, z)) \\ \lambda : m(e, x) \rightarrow x \\ \rho : m(x, e) \rightarrow x \\ \gamma : m(x, y) \rightarrow m(y, x) \end{array} \right\}$$

# PRESENTATIONS OF LAWVERE THEORIES

String rewriting systems correspond to presentations of monoids.

# PRESENTATIONS OF LAWVERE THEORIES

Term rewriting systems correspond to presentations of **Lawvere theories**.

# PRESENTATIONS OF LAWVERE THEORIES

A **Lawvere theory** is a category  $\mathcal{C}$

- ▶ whose objects are integers
- ▶ which is cartesian
- ▶ whose cartesian product is given on objects by addition

# PRESENTATIONS OF LAWVERE THEORIES

A **Lawvere theory** is a category  $\mathcal{C}$

- ▶ whose objects are integers
- ▶ which is cartesian
- ▶ whose cartesian product is given on objects by addition

TRS  $((\Sigma, \alpha), R)$  induce a LT whose morphisms  $m \rightarrow n$  are  $n$ -uples of terms with variables  $x_1, \dots, x_m$ , considered modulo relations.

Presentation:  $\mathcal{C} \cong \Sigma^* / \equiv_R$

# PRESENTATIONS OF LAWVERE THEORIES

A **Lawvere theory** is a category  $\mathcal{C}$

- ▶ whose objects are integers
- ▶ which is cartesian
- ▶ whose cartesian product is given on objects by addition

TRS  $((\Sigma, \alpha), R)$  induce a LT whose morphisms  $m \rightarrow n$  are  $n$ -uples of terms with variables  $x_1, \dots, x_m$ , considered modulo relations.

Presentation:  $\mathcal{C} \cong \Sigma^* / \equiv_R$

## Example

Consider the TRS of commutative monoids:  $\Sigma = \{m : 2, e : 0\}$

$$R = \left\{ \begin{array}{l} \alpha : m(m(x, y), z) \rightarrow m(x, m(y, z)) \\ \lambda : m(e, x) \rightarrow x \\ \rho : m(x, e) \rightarrow x \\ \gamma : m(x, y) \rightarrow m(y, x) \end{array} \right\}$$

# PRESENTATIONS OF LAWVERE THEORIES

A **Lawvere theory** is a category  $\mathcal{C}$

- ▶ whose objects are integers
- ▶ which is cartesian
- ▶ whose cartesian product is given on objects by addition

TRS  $((\Sigma, \alpha), R)$  induce a LT whose morphisms  $m \rightarrow n$  are  $n$ -uples of terms with variables  $x_1, \dots, x_m$ , considered modulo relations.

Presentation:  $\mathcal{C} \cong \Sigma^* / \equiv_R$

## Example

Consider the TRS of commutative monoids:  $\Sigma = \{m : 2, e : 0\}$

It presents the Lawvere theory whose morphisms  $M : m \rightarrow n$  are  $(m \times n)$ -matrices with coefficients in  $\mathbb{N}$ .

# PRESENTATIONS OF LAWVERE THEORIES

A **Lawvere theory** is a category  $\mathcal{C}$

- ▶ whose objects are integers
- ▶ which is cartesian
- ▶ whose cartesian product is given on objects by addition

TRS  $((\Sigma, \alpha), R)$  induce a LT whose morphisms  $m \rightarrow n$  are  $n$ -uples of terms with variables  $x_1, \dots, x_m$ , considered modulo relations.

Presentation:  $\mathcal{C} \cong \Sigma^* / \equiv_R$

## Example

Consider the TRS of commutative monoids:  $\Sigma = \{m : 2, e : 0\}$

$$[m(m(x_1, x_1), x_2) ; e ; x_2] : 2 \rightarrow 3$$

# PRESENTATIONS OF LAWVERE THEORIES

A **Lawvere theory** is a category  $\mathcal{C}$

- ▶ whose objects are integers
- ▶ which is cartesian
- ▶ whose cartesian product is given on objects by addition

TRS  $((\Sigma, \alpha), R)$  induce a LT whose morphisms  $m \rightarrow n$  are  $n$ -uples of terms with variables  $x_1, \dots, x_m$ , considered modulo relations.

Presentation:  $\mathcal{C} \cong \Sigma^* / \equiv_R$

## Example

Consider the TRS of commutative monoids:  $\Sigma = \{m : 2, e : 0\}$

$$[m(m(x_1, x_1), x_2) ; e ; x_2] : 2 \rightarrow 3 \quad \rightsquigarrow \quad \begin{pmatrix} 2 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

# PRESENTATIONS OF LAWVERE THEORIES

A **Lawvere theory** is a category  $\mathcal{C}$

- ▶ whose objects are integers
- ▶ which is cartesian
- ▶ whose cartesian product is given on objects by addition

TRS  $((\Sigma, \alpha), R)$  induce a LT whose morphisms  $m \rightarrow n$  are  $n$ -uples of terms with variables  $x_1, \dots, x_m$ , considered modulo relations.

Presentation:  $\mathcal{C} \cong \Sigma^* / \equiv_R$

## Example

Consider the TRS of commutative monoids:  $\Sigma = \{m : 2, e : 0\}$

$$\begin{array}{l} [m(m(x_1, x_1), x_2) ; e ; x_2] \\ [m(x_1, m(x_1, x_2)) ; e ; m(e, x_2)] \end{array} : 2 \rightarrow 3 \quad \rightsquigarrow \quad \begin{pmatrix} 2 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

# PRESENTATIONS OF LAWVERE THEORIES

A **Lawvere theory** is a category  $\mathcal{C}$

- ▶ whose objects are integers
- ▶ which is cartesian
- ▶ whose cartesian product is given on objects by addition

TRS  $((\Sigma, \alpha), R)$  induce a LT whose morphisms  $m \rightarrow n$  are  $n$ -uples of terms with variables  $x_1, \dots, x_m$ , considered modulo relations.

Presentation:  $\mathcal{C} \cong \Sigma^* / \equiv_R$

## Example

Consider the TRS of commutative monoids:  $\Sigma = \{m : 2, e : 0\}$

$$\begin{array}{l} [m(m(x_1, x_1), x_2) ; e ; x_2] \\ [m(x_1, m(x_1, x_2)) ; e ; m(e, x_2)] \end{array} : 2 \rightarrow 3 \quad \rightsquigarrow \quad \begin{pmatrix} 2 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Use rewriting theory!

Can we use the same techniques  
in order to build  
presentations of  $n$ -categories?

# POLYGRAPHS

[Street76,Burroni93,Power90]

# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

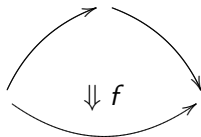
dimension	rewr. syst.	presents
1	string	monoid

$$\xrightarrow{a} \xrightarrow{b} \xrightarrow{c}$$

# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
1	string	monoid
2	term	Lawvere th.



# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	set
1	string	monoid
2	term	Lawvere th.

# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	set
1	string	monoid
2	term	Lawvere th.

# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	<b>0</b> -category
1	string	monoid
2	term	Lawvere th.

set = 0-category

# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	0-category
1	string	<b>monoid</b>
2	term	Lawvere th.

# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	0-category
1	string	1-category
2	term	Lawvere th.

monoid = 1-category with only one object

Generalization:  $\xrightarrow{a} \xrightarrow{b}$   $\rightsquigarrow$   $x \xrightarrow{a} y \xrightarrow{b} y$

# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	0-category
1	string	1-category
2	term	Lawvere th.

# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	0-category
1	string	1-category
2	term	cartesian category

Lawvere th. = cartesian category with  $\mathbb{N}$  as objects

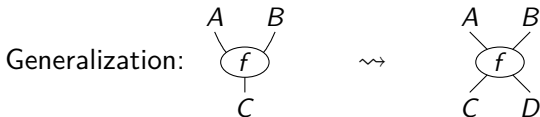


# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	0-category
1	string	1-category
2	term	monoidal category

cartesian category = monoidal category in which every object is a comonoid

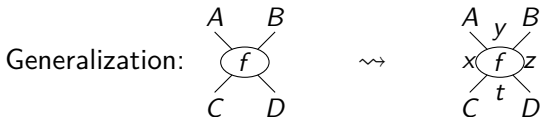


# PRESENTING $n$ -CATEGORIES

We want to generalize rewriting systems

dimension	rewr. syst.	presents
0	element	0-category
1	string	1-category
2	term	2-category

monoidal category = 2-category with only one object



# POLYGRAPHS

A 0-signature

$\Sigma_0$

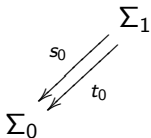
Example

signature

$x \quad y$

# POLYGRAPHS

A 0-rewriting system

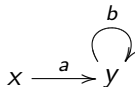


Example

signature

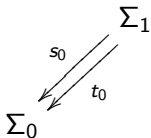
$x$        $y$

rules

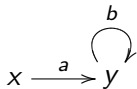


# POLYGRAPHS

A 1-signature = a 0-rewriting system

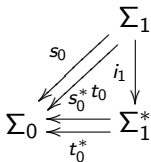


Example  
signature



# POLYGRAPHS

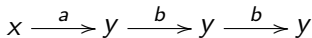
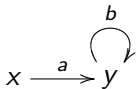
A 1-signature generates a *category*



## Example

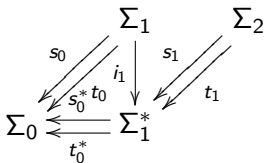
signature

terms



# POLYGRAPHS

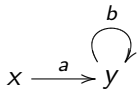
A 1-rewriting system



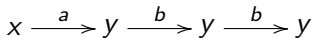
such that  $s_0^* \circ s_1 = s_0^* \circ t_1$  and  $t_0^* \circ s_1 = t_0^* \circ t_1$

Example

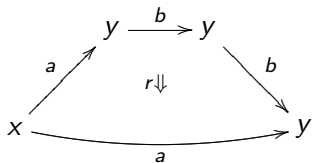
signature



terms

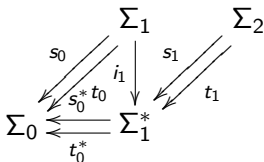


rules



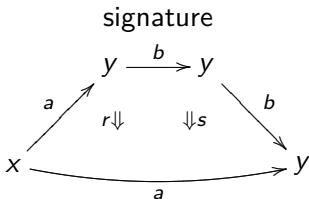
# POLYGRAPHS

A 2-signature = a 1-rewriting system



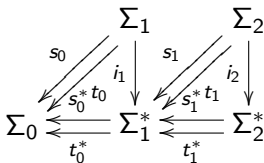
such that  $s_0^* \circ s_1 = s_0^* \circ t_1$  and  $t_0^* \circ s_1 = t_0^* \circ t_1$

## Example



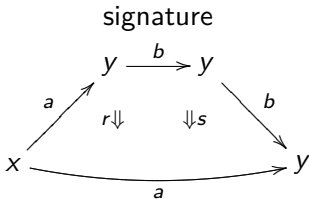
# POLYGRAPHS

A 2-signature generates a 2-category



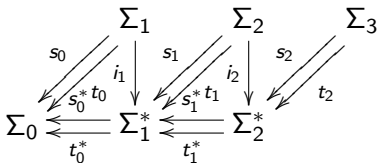
such that  $s_0^* \circ s_1 = s_0^* \circ t_1$  and  $t_0^* \circ s_1 = t_0^* \circ t_1$

## Example



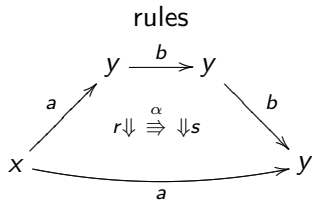
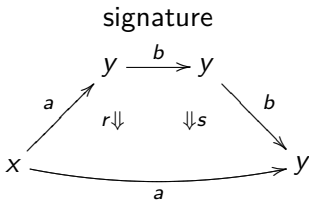
# POLYGRAPHS

A 2-rewriting system



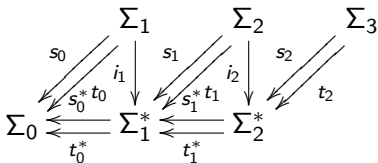
such that  $s_1^* \circ s_2 = s_1^* \circ t_2$  and  $t_1^* \circ s_2 = t_1^* \circ t_2$

Example



# POLYGRAPHS

A 2-rewriting system



Right notion of  $n$ -rewriting system:  $n$ -polygraphs.

An example: a presentation of **Bij**

[Lafont03]

# A PRESENTATION OF $\mathbf{Bij}$

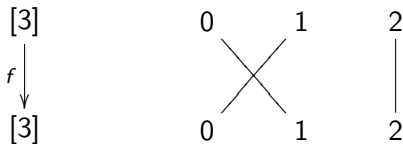
The category  $\mathbf{Bij}$  has

- ▶ objects are integers  $[n] = \{0, \dots, n - 1\}$
- ▶ morphisms  $f : [m] \rightarrow [n]$  are bijections

# A PRESENTATION OF $\mathbf{Bij}$

The category  $\mathbf{Bij}$  has

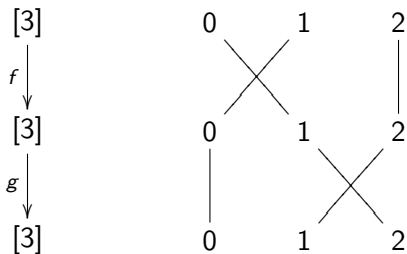
- ▶ objects are integers  $[n] = \{0, \dots, n-1\}$
- ▶ morphisms  $f : [m] \rightarrow [n]$  are bijections



# A PRESENTATION OF $\mathbf{Bij}$

The category  $\mathbf{Bij}$  has

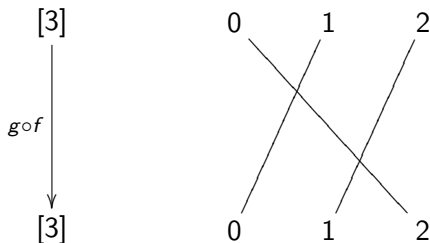
- ▶ objects are integers  $[n] = \{0, \dots, n-1\}$
- ▶ morphisms  $f : [m] \rightarrow [n]$  are bijections



# A PRESENTATION OF $\mathbf{Bij}$

The category  $\mathbf{Bij}$  has

- ▶ objects are integers  $[n] = \{0, \dots, n-1\}$
- ▶ morphisms  $f : [m] \rightarrow [n]$  are bijections

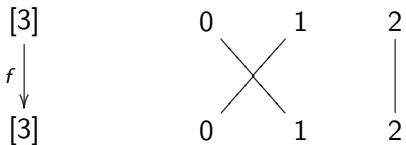


Vertical composition  $\circ$

# A PRESENTATION OF $\mathbf{Bij}$

The category  $\mathbf{Bij}$  has

- ▶ objects are integers  $[n] = \{0, \dots, n-1\}$
- ▶ morphisms  $f : [m] \rightarrow [n]$  are bijections

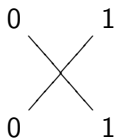
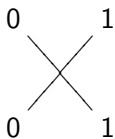


# A PRESENTATION OF $\mathbf{Bij}$

The category  $\mathbf{Bij}$  has

- ▶ objects are integers  $[n] = \{0, \dots, n-1\}$
- ▶ morphisms  $f : [m] \rightarrow [n]$  are bijections

$$\begin{array}{c} [3] \\ f \downarrow \\ [3] \end{array}$$

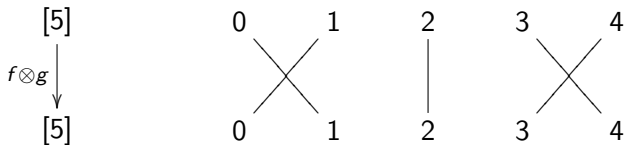


$$\begin{array}{c} [2] \\ g \downarrow \\ [2] \end{array}$$

# A PRESENTATION OF $\mathbf{Bij}$

The category  $\mathbf{Bij}$  has

- ▶ objects are integers  $[n] = \{0, \dots, n-1\}$
- ▶ morphisms  $f : [m] \rightarrow [n]$  are bijections



Horizontal composition  $\otimes$

# A PRESENTATION OF $\mathbf{Bij}$

We want to give a presentation of  $\mathbf{Bij}$ , i.e. describe it as

- ▶ a free category on sets of typed generators for 0-, 1- and 2-cells
- ▶ quotiented by relations between 2-cells in the generated 2-category

# A PRESENTATION OF $\mathbf{Bij}$

$\mathbf{Bij}$  is presented by the 3-polygraph such that [Lafont03]

- ▶  $\Sigma_0 = \{*\}$

# A PRESENTATION OF $\mathbf{Bij}$

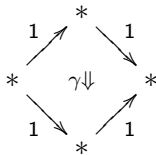
$\mathbf{Bij}$  is presented by the 3-polygraph such that [Lafont03]

- ▶  $\Sigma_0 = \{*\}$
- ▶  $\Sigma_1 = \{1 : * \rightarrow *\}$  (so  $\Sigma_1^* \approx \mathbb{N}$ )

# A PRESENTATION OF $\mathbf{Bij}$

$\mathbf{Bij}$  is presented by the 3-polygraph such that [Lafont03]

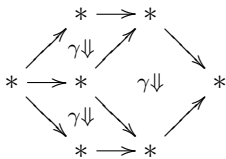
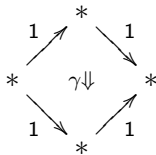
- ▶  $\Sigma_0 = \{*\}$
- ▶  $\Sigma_1 = \{1 : * \rightarrow *\}$  (so  $\Sigma_1^* \approx \mathbb{N}$ )
- ▶  $\Sigma_2 = \{\gamma : 1 \otimes 1 \rightarrow 1 \otimes 1\}$



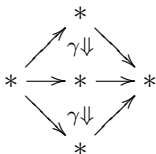
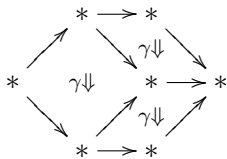
# A PRESENTATION OF $\text{Bij}$

$\text{Bij}$  is presented by the 3-polygraph such that [Lafont03]

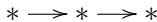
- ▶  $\Sigma_0 = \{*\}$
- ▶  $\Sigma_1 = \{1 : * \rightarrow *\}$  (so  $\Sigma_1^* \approx \mathbb{N}$ )
- ▶  $\Sigma_2 = \{\gamma : 1 \otimes 1 \rightarrow 1 \otimes 1\}$
- ▶  $\Sigma_3 = \{y, s\}$



$\xRightarrow{y}$



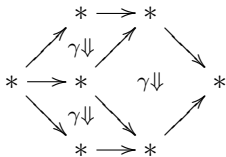
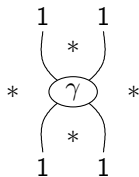
$\xRightarrow{s}$



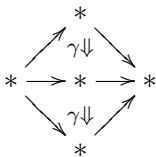
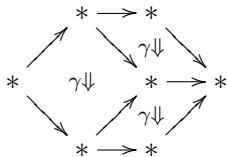
# A PRESENTATION OF $\text{Bij}$

$\text{Bij}$  is presented by the 3-polygraph such that [Lafont03]

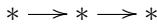
- ▶  $\Sigma_0 = \{*\}$
- ▶  $\Sigma_1 = \{1 : * \rightarrow *\}$  (so  $\Sigma_1^* \approx \mathbb{N}$ )
- ▶  $\Sigma_2 = \{\gamma : 1 \otimes 1 \rightarrow 1 \otimes 1\}$
- ▶  $\Sigma_3 = \{y, s\}$



$\xRightarrow{y}$



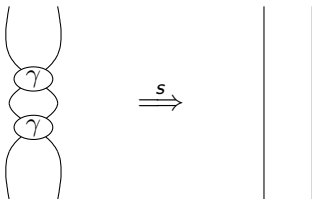
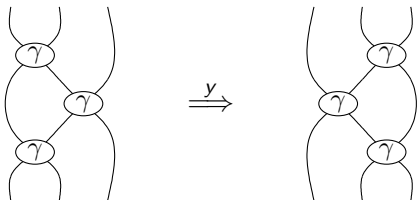
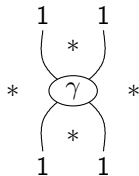
$\xRightarrow{s}$



# A PRESENTATION OF $\text{Bij}$

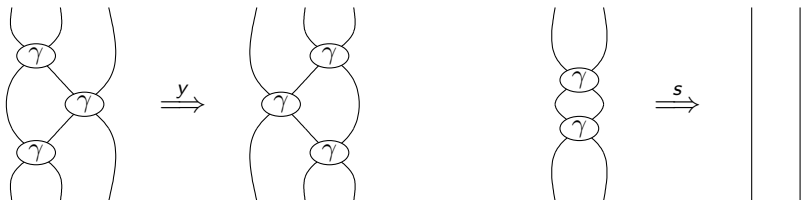
$\text{Bij}$  is presented by the 3-polygraph such that [Lafont03]

- ▶  $\Sigma_0 = \{*\}$
- ▶  $\Sigma_1 = \{1 : * \rightarrow *\}$  (so  $\Sigma_1^* \approx \mathbb{N}$ )
- ▶  $\Sigma_2 = \{\gamma : 1 \otimes 1 \rightarrow 1 \otimes 1\}$
- ▶  $\Sigma_3 = \{y, s\}$



# A PRESENTATION OF $Bij$

The rules

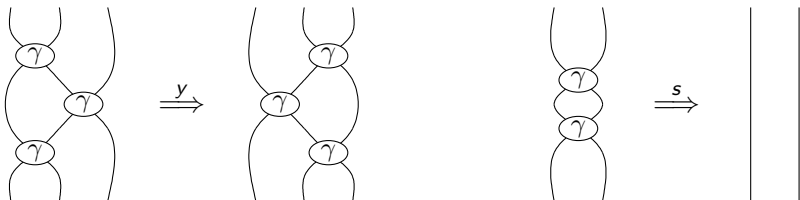


of the rewriting system induce critical pairs

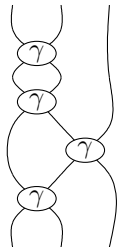


# A PRESENTATION OF $Bij$

The rules

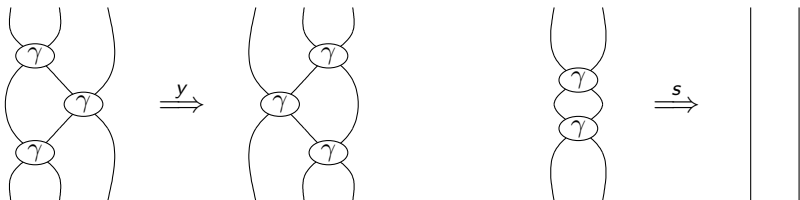


of the rewriting system induce critical pairs

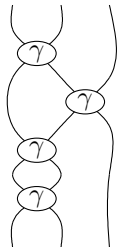


# A PRESENTATION OF $Bij$

The rules

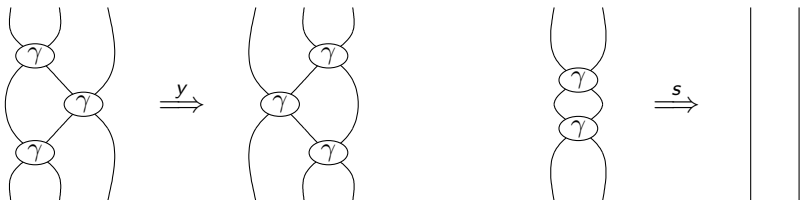


of the rewriting system induce critical pairs

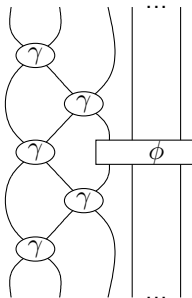


# A PRESENTATION OF $Bij$

The rules



of the rewriting system induce critical pairs



Critical pairs are computed using a  
unification procedure.

We want to extend it to  
2-dimensional rewriting systems

Contrarily to term rewriting systems  
we can have an **infinite** number  
of critical pairs...

IDEA:  
change the definition  
of critical pairs

# **CRITICAL PAIRS**

**IN THE MULTICATEGORY OF  
COMPACT CONTEXTS**

# TWO PROBLEMS WITH CRITICAL PAIRS

Consider the 2-rewriting system  $\Sigma$  with

$$\Sigma_0 = \{*\} \quad \Sigma_1 = \{1\} \quad \Sigma_2 = \{s : 1 \rightarrow 1, d : 1 \rightarrow 3, m : 3 \rightarrow 1\}$$

the generators for 2-cells are drawn respectively as



# TWO PROBLEMS WITH CRITICAL PAIRS

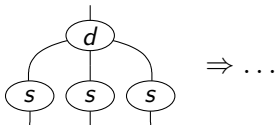
Consider the 2-rewriting system  $\Sigma$  with

$$\Sigma_0 = \{*\} \quad \Sigma_1 = \{1\} \quad \Sigma_2 = \{s : 1 \rightarrow 1, d : 1 \rightarrow 3, m : 3 \rightarrow 1\}$$

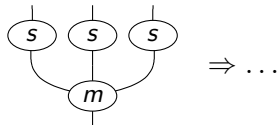
the generators for 2-cells are drawn respectively as



with rules

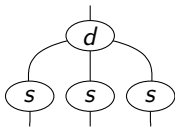


and



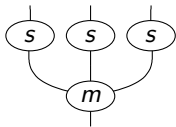
# TWO PROBLEMS WITH CRITICAL PAIRS

The two rules



$\Rightarrow \dots$

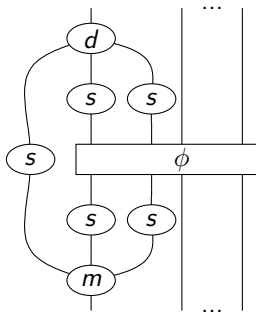
and



$\Rightarrow \dots$

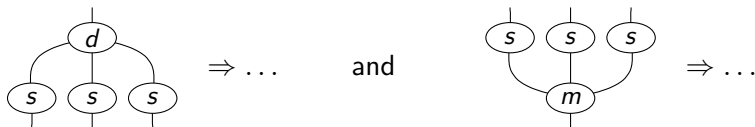
induce an **infinite** number of critical pairs:

*variables on the border:*



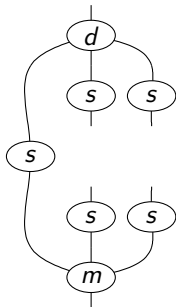
# TWO PROBLEMS WITH CRITICAL PAIRS

The two rules



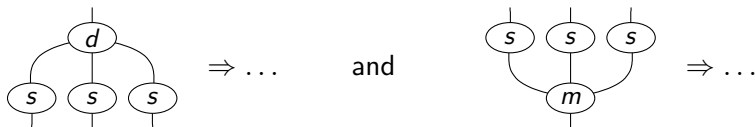
induce an **infinite** number of critical pairs:

*variables on the border:*  
use **compact** morphisms!



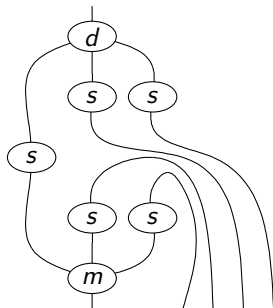
# TWO PROBLEMS WITH CRITICAL PAIRS

The two rules



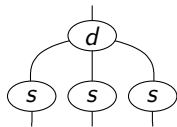
induce an **infinite** number of critical pairs:

*variables on the border:*  
use **compact** morphisms!



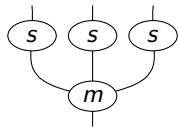
# TWO PROBLEMS WITH CRITICAL PAIRS

The two rules



$\Rightarrow \dots$

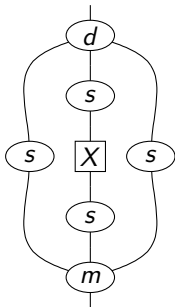
and



$\Rightarrow \dots$

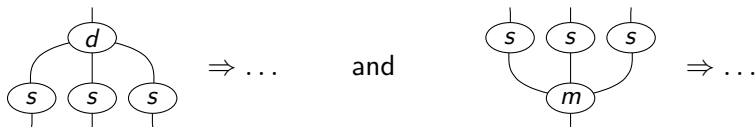
induce an **infinite** number of critical pairs:

*variables inside:*



# TWO PROBLEMS WITH CRITICAL PAIRS

The two rules

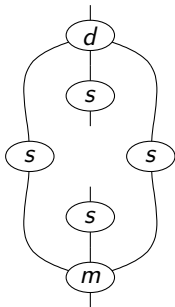


induce an **infinite** number of critical pairs:

*variables inside:*

use **contexts!**

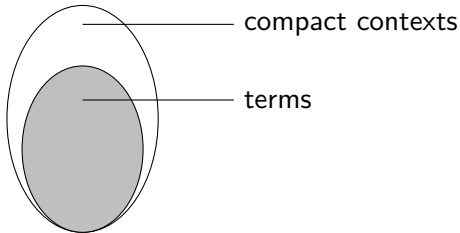
(= multicategory of terms with metavariables)



# BACK TO A FINITE NUMBER OF CRITICAL PAIRS

## Theorem

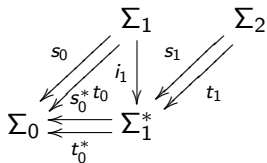
The 2-category of “terms” generated by a signature can be embedded into the ***multicategory of compact contexts***.



In other words, there is a finite number of *generating families* of critical pairs in those rewriting systems.

# THE MULTICATEGORY OF CONTEXTS

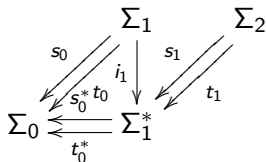
Consider a 2-polygraph  $\Sigma$



and a family  $\mathcal{X} = \{X_1 : f_1 \Rightarrow g_1, \dots, X_n : f_n \Rightarrow g_n\}$  of 2-globes, with  $f_i, g_i \in \Sigma_1^*$  parallel 1-cells considered as *formal variables*

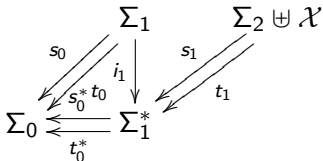
# THE MULTICATEGORY OF CONTEXTS

Consider a 2-polygraph  $\Sigma$



and a family  $\mathcal{X} = \{X_1 : f_1 \Rightarrow g_1, \dots, X_n : f_n \Rightarrow g_n\}$  of 2-globes, with  $f_i, g_i \in \Sigma_1^*$  parallel 1-cells considered as *formal variables*

We define the 2-polygraph  $\Sigma[\mathcal{X}]$  as



# THE MULTICATEGORY OF CONTEXTS

Consider a 2-polygraph  $\Sigma$

$$\begin{array}{ccccc}
 & & \Sigma_1 & & \Sigma_2 \\
 & \swarrow s_0 & \downarrow i_1 & \swarrow s_1 & \downarrow i_2 \\
 & \Sigma_0 & \Sigma_1^* & \Sigma_2^* & \\
 & \leftarrow s_0^* t_0 & \leftarrow s_1^* t_1 & & \\
 & \leftarrow t_0^* & \leftarrow t_1^* & & 
 \end{array}$$

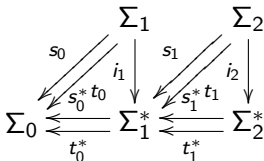
and a family  $\mathcal{X} = \{X_1 : f_1 \Rightarrow g_1, \dots, X_n : f_n \Rightarrow g_n\}$  of 2-globes, with  $f_i, g_i \in \Sigma_1^*$  parallel 1-cells considered as *formal variables*

We define the 2-polygraph  $\Sigma[\mathcal{X}]$  as

$$\begin{array}{ccccc}
 & & \Sigma_1 & & \Sigma_2 \uplus \mathcal{X} \\
 & \swarrow s_0 & \downarrow i_1 & \swarrow s_1 & \downarrow i_2 \\
 & \Sigma_0 & \Sigma_1^* & (\Sigma_2 \uplus \mathcal{X})^* & \\
 & \leftarrow s_0^* t_0 & \leftarrow s_1^* t_1 & & \\
 & \leftarrow t_0^* & \leftarrow t_1^* & & 
 \end{array}$$

# THE MULTICATEGORY OF CONTEXTS

Consider a 2-polygraph  $\Sigma$



## Substitution

Given

- ▶ a 2-cell  $\alpha : f \Rightarrow f'$  in  $\Sigma_2^*$
- ▶ and a 2-cell  $\beta : g \Rightarrow g'$  in  $\Sigma_2[X : f \Rightarrow f']^*$

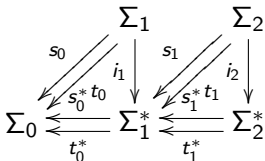
we can define

- ▶ a 2-cell  $\alpha[\beta/X] : f \Rightarrow f'$  in  $\Sigma_2^*$

which corresponds to the 2-cell  $\alpha$  where all occurrences of  $X$  have been replaced by  $\beta$ .

# THE MULTICATEGORY OF CONTEXTS

Consider a 2-polygraph  $\Sigma$



## Definition

We can thus define the **multicategory of contexts**  $\mathcal{K}_\Sigma$  whose

- ▶ objects are globes  $f \Rightarrow g$ , i.e. parallel 1-cells in  $\Sigma_1^*$
- ▶ operations in  $\mathcal{K}_\Sigma(f_1 \Rightarrow g_1, \dots, f_n \Rightarrow g_n; f \Rightarrow g)$  are 2-cells  $\kappa: f \Rightarrow g$  in  $\Sigma_2[X_1: f_1 \Rightarrow g_1, \dots, X_n: f_n \Rightarrow g_n]^*$
- ▶ composition is given by generalizing the previous substitution

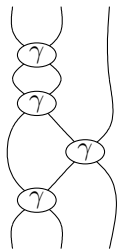
# CRITICAL PAIRS

## Definition

Suppose that we are given two 2-cells

$$\alpha_1 : f_1 \Rightarrow g_1 \quad \text{and} \quad \beta_1 : f_1 \Rightarrow g_1$$

in a 2-polygraph  $\Sigma$ . A **most general unifier** is



# CRITICAL PAIRS

## Definition

Suppose that we are given two 2-cells

$$\alpha_1 : f_1 \Rightarrow g_1 \quad \text{and} \quad \beta_1 : f_1 \Rightarrow g_1$$

in a 2-polygraph  $\Sigma$ . A **most general unifier** is a pair

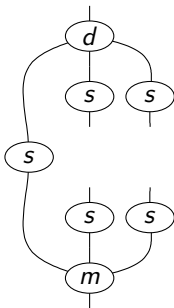
$$\kappa_1 \in \mathcal{K}_\Sigma(f_1 \Rightarrow g_1; f \Rightarrow g) \quad \text{and} \quad \kappa_2 \in \mathcal{K}_\Sigma(f_2 \Rightarrow g_2; f \Rightarrow g)$$

of *linear* contexts such that

1. *unifier*:  $\kappa_1(\alpha_1) = \kappa_2(\alpha_2)$
2. *minimal*: if  $\kappa_1 = \kappa_1'' \circ \kappa_1'$  and  $\kappa_2 = \kappa_2'' \circ \kappa_2'$  where  $(\kappa_1', \kappa_2')$  is a unifier then  $\kappa_1'' = \text{id}$  and  $\kappa_2'' = \text{id}$
3. *overlapping*: there is no binary context  $\kappa$  such that  $\kappa_1 = (\text{id}, \alpha_2)$  and  $\kappa_2 = (\alpha_1, \text{id})$

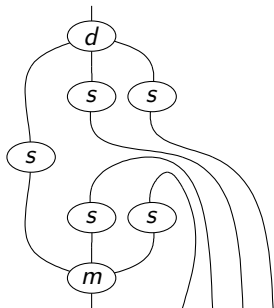
# COMPACT 2-CATEGORIES

Now, we want to represent morphisms with “holes” in the border



# COMPACT 2-CATEGORIES

Now, we want to represent morphisms with “holes” in the border



# COMPACT 2-CATEGORIES

## Definition

Given a 2-category  $\mathcal{C}$ , a 1-cell  $f : A \rightarrow B$  is **left adjoint** to a 1-cell  $g : B \rightarrow A$ , which we write

$$A \begin{array}{c} \xrightarrow{f} \\ \perp \\ \xleftarrow{g} \end{array} B$$

when there exists two 2-cells  $\eta : A \rightarrow f \otimes g$  and  $\varepsilon : g \otimes f \rightarrow B$

$$\begin{array}{c} A \\ \circlearrowleft \eta \\ \begin{array}{cc} f & g \end{array} \end{array} \qquad \begin{array}{c} \begin{array}{cc} g & f \end{array} \\ \circlearrowright \varepsilon \\ B \end{array}$$

such that

$$(f \otimes \varepsilon) \circ (\eta \otimes f) = f \quad \text{and} \quad (\varepsilon \otimes g) \circ (g \otimes \eta) = g$$

# COMPACT 2-CATEGORIES

## Definition

A 2-category  $\mathcal{C}$  is **compact** when every 1-cell admits both a left and a right adjoint.

# COMPACT 2-CATEGORIES

## Definition

A 2-category  $\mathcal{C}$  is **compact** when every 1-cell admits both a left and a right adjoint.

## Lemma

A 2-category  $\mathcal{C}$  generates a free compact 2-category  $\mathcal{A}_{\mathcal{C}}$ , which is

- ▶ 0-cells: same as  $\mathcal{C}$ ,
- ▶ 1-cells:  $f^n$  with  $f$  a 1-cell of  $\mathcal{C}$  and  $n \in \mathbb{Z}$ ,
- ▶ 2-cells:
  - ▶  $\alpha : f^0 \Rightarrow g^0$  for  $\alpha : f \Rightarrow g$  a 2-cell of  $\mathcal{C}$ ,
  - ▶  $\eta_{f^n} : \text{id} \Rightarrow f^{n-1} \otimes f^n$
  - ▶  $\varepsilon_{f^n} : f^n \otimes f^{n-1} \Rightarrow \text{id}$
- ▶ + equations

# COMPACT 2-CATEGORIES

## Definition

A 2-category  $\mathcal{C}$  is **compact** when every 1-cell admits both a left and a right adjoint.

## Lemma

A 2-category  $\mathcal{C}$  generates a free compact 2-category  $\mathcal{A}_{\mathcal{C}}$ , which is

- ▶ 0-cells: same as  $\mathcal{C}$ ,
- ▶ 1-cells:  $f^n$  with  $f$  a 1-cell of  $\mathcal{C}$  and  $n \in \mathbb{Z}$ ,
- ▶ 2-cells:
  - ▶  $\alpha : f^0 \Rightarrow g^0$  for  $\alpha : f \Rightarrow g$  a 2-cell of  $\mathcal{C}$ ,
  - ▶  $\eta_{f^n} : \text{id} \Rightarrow f^{n-1} \otimes f^n$
  - ▶  $\varepsilon_{f^n} : f^n \otimes f^{n-1} \Rightarrow \text{id}$
- ▶ + equations

## Theorem

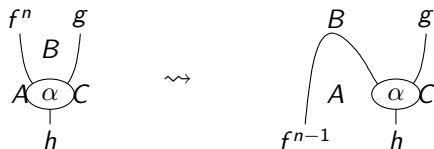
A 2-category  $\mathcal{C}$  embeds fully and faithfully in the free compact category it generates.

# ROTATIONS

In the free compact 2-category  $\mathcal{A}_C$ , the following Hom-sets

$$\mathcal{A}_C(f^n \otimes g, h) \cong \mathcal{A}_C(g, f^{n-1} \otimes h)$$

are isomorphic:



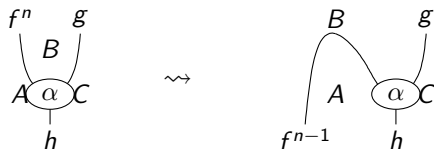
We call these isomorphisms **rotations**.

# ROTATIONS

In the free compact 2-category  $\mathcal{A}_C$ , the following Hom-sets

$$\mathcal{A}_C(f^n \otimes g, h) \cong \mathcal{A}_C(g, f^{n-1} \otimes h)$$

are isomorphic:

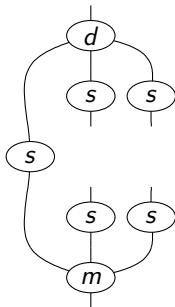


We call these isomorphisms **rotations**.

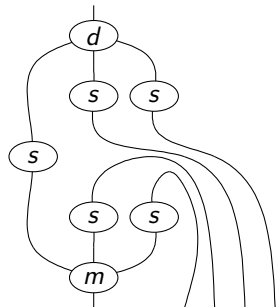
Rotations are unary operations in  $\mathcal{K}_{\mathcal{A}_C}$ .

# ROTATIONS

The diagram



is a representation of



*up to rotation!*

# THE MULTICATEGORY OF COMPACT CONTEXTS

A 2-polygraph  $\Sigma$  generates a 2-category  $\mathcal{C}$ ,

- ▶ which can be embedded in the free compact 2-category  $\mathcal{A}_{\mathcal{C}}$
- ▶ and the 2-cells  $\alpha : f \Rightarrow g$  of  $\mathcal{A}_{\mathcal{C}}$  can be seen as nullary contexts in  $\mathcal{K}_{\mathcal{A}_{\mathcal{C}}} (; f \Rightarrow g)$

# UNIFICATION IN THE MULTICATEGORY OF COMPACT CONTEXTS

## Theorem

Given a 3-polygraph  $R$  with underlying 2-polygraph  $\Sigma$  generating a 2-category  $\mathcal{C}$ , there exists a **finite** number of contexts

$$\kappa^i \in \mathcal{K}_{\mathcal{A}_{\mathcal{C}}}(f_1^i \Rightarrow g_1^i, \dots, f_{k_i}^i \Rightarrow g_{k_i}^i; f^i \Rightarrow g^i)$$

such that

- ▶ for any nullary contexts  $\kappa_1, \dots, \kappa_{k_i}$  and unary context  $\kappa$  such that the composite  $\kappa \circ \kappa^i \circ (\kappa_1, \dots, \kappa_{k_i}) \in \mathcal{K}_{\mathcal{A}_{\mathcal{C}}} (; f' \Rightarrow g')$  is of the form  $\kappa_{\alpha}$ , for some 2-cell  $\alpha : f' \Rightarrow g'$  of  $\mathcal{C}$ , the 2-cell  $\alpha$  is an unifier of left members of two rewriting rules of  $R$ ,
- ▶ and moreover any such unifier can be obtained in this way (in particular the critical pairs).

# UNIFICATION IN THE MULTICATEGORY OF COMPACT CONTEXTS

## Theorem

Given a 3-polygraph  $R$  with underlying 2-polygraph  $\Sigma$  generating a 2-category  $\mathcal{C}$ , there exists a **finite** number of contexts

$$\kappa^i \in \mathcal{K}_{\mathcal{A}_{\mathcal{C}}}(f_1^i \Rightarrow g_1^i, \dots, f_{k_i}^i \Rightarrow g_{k_i}^i; f^i \Rightarrow g^i)$$

such that

- ▶ for any nullary contexts  $\kappa_1, \dots, \kappa_{k_i}$  and unary context  $\kappa$  such that the composite  $\kappa \circ \kappa^i \circ (\kappa_1, \dots, \kappa_{k_i}) \in \mathcal{K}_{\mathcal{A}_{\mathcal{C}}} (; f' \Rightarrow g')$  is of the form  $\kappa_{\alpha}$ , for some 2-cell  $\alpha : f' \Rightarrow g'$  of  $\mathcal{C}$ , the 2-cell  $\alpha$  is an unifier of left members of two rewriting rules of  $R$ ,
- ▶ and moreover any such unifier can be obtained in this way (in particular the critical pairs).

## Remark

There is unicity up to rotations.

# **UNIFICATION**

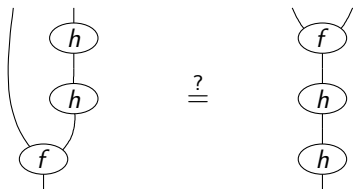
**IN THE MULTICATEGORY OF  
COMPACT CONTEXTS**

# UNIFICATION IN TRS

Suppose that we have a TRS

$$f : 2 \quad h : 1 \quad f(x, h(h(y))) \Rightarrow \dots \quad h(h(f(x, y))) \Rightarrow \dots$$

In order to generate critical pairs,  
we unify a subterm of  $f(x, h(h(y)))$  with  $h(h(f(x, y)))$

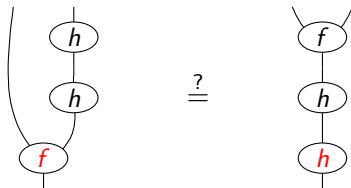


# UNIFICATION IN TRS

Suppose that we have a TRS

$$f : 2 \quad h : 1 \quad f(x, h(h(y))) \Rightarrow \dots \quad h(h(f(x, y))) \Rightarrow \dots$$

In order to generate critical pairs,  
we unify a subterm of  $f(x, h(h(y)))$  with  $h(h(f(x, y)))$

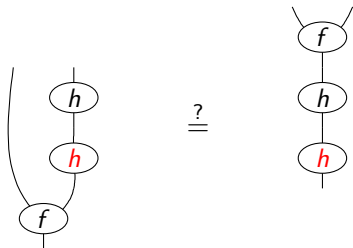


# UNIFICATION IN TRS

Suppose that we have a TRS

$$f : 2 \quad h : 1 \quad f(x, h(h(y))) \Rightarrow \dots \quad h(h(f(x, y))) \Rightarrow \dots$$

In order to generate critical pairs,  
we unify a subterm of  $f(x, h(h(y)))$  with  $h(h(f(x, y)))$

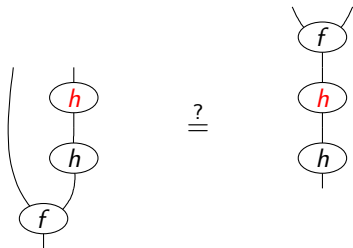


# UNIFICATION IN TRS

Suppose that we have a TRS

$$f : 2 \quad h : 1 \quad f(x, h(h(y))) \Rightarrow \dots \quad h(h(f(x, y))) \Rightarrow \dots$$

In order to generate critical pairs,  
we unify a subterm of  $f(x, h(h(y)))$  with  $h(h(f(x, y)))$

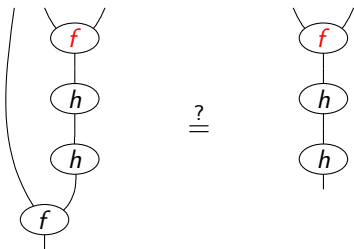


# UNIFICATION IN TRS

Suppose that we have a TRS

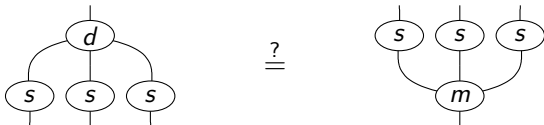
$$f : 2 \quad h : 1 \quad f(x, h(h(y))) \Rightarrow \dots \quad h(h(f(x, y))) \Rightarrow \dots$$

In order to generate critical pairs,  
we unify a subterm of  $f(x, h(h(y)))$  with  $h(h(f(x, y)))$



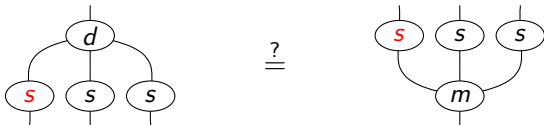
# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



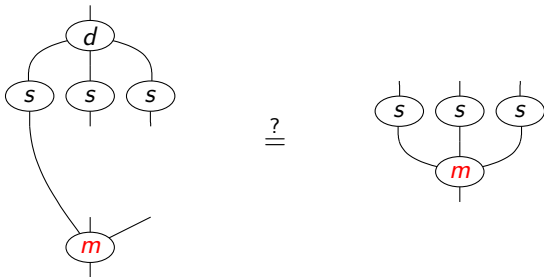
# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



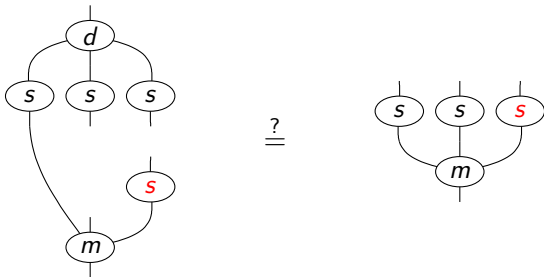
# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



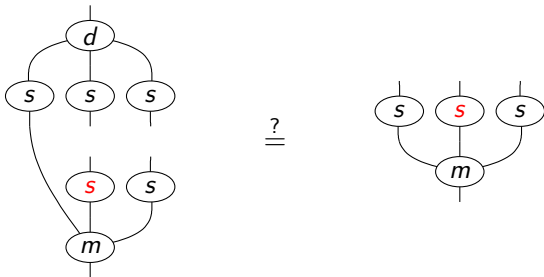
# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



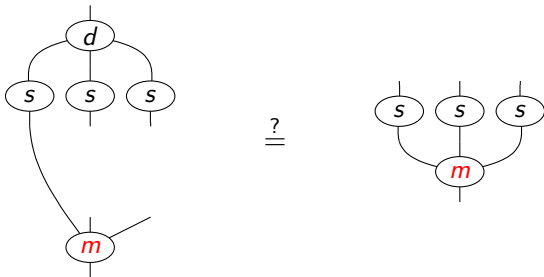
# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



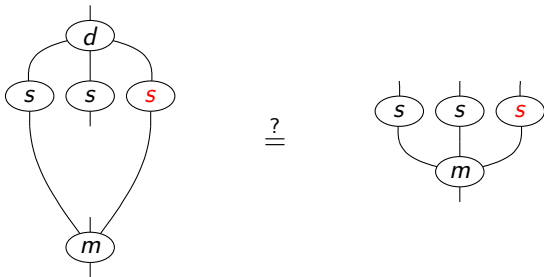
# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



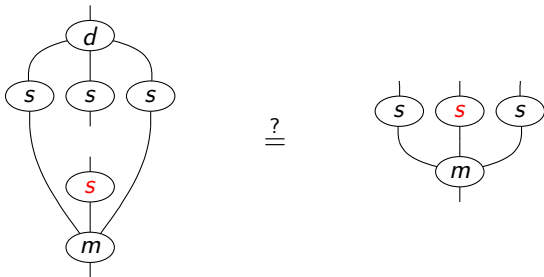
# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



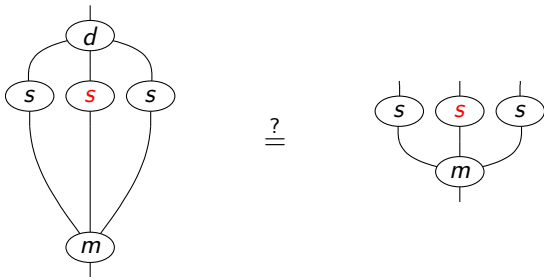
# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



# UNIFICATION IN 2-REWRITING SYSTEMS

We want to compute the critical pairs generated by



# METACONFLUENCE

## Remark

*When a 2-rewriting system is confluent, its critical pairs (in our generalized sense) are not necessarily confluent. But at least, we get a finite description of the critical pairs!*

# NOT SHOWN HERE

- ▶ The operations in the multicategory of compact contexts can be represented in an effective way.

# NOT SHOWN HERE

- ▶ The operations in the multicategory of compact contexts can be represented in an effective way.
- ▶ Precise formulation of the algorithm.

# NOT SHOWN HERE

- ▶ The operations in the multicategory of compact contexts can be represented in an effective way.
- ▶ Precise formulation of the algorithm.
- ▶ An implementation was realized.

# FUTURE WORKS

- ▶ Generalize techniques developed by Guiraud and Malbos to this setting
- ▶ Generalize to higher dimensions
- ▶ Towards automated tools for studying higher categories?
- ▶ ...

**THANKS!**

Any questions?