Computing Critical Pairs in 2-Dimensional Rewriting Systems

Samuel Mimram

Rewriting Theory and Applications 2010

July 12th 2010

The big plan

Rewriting theory has to be generalized
in higher dimensions!

The big plan

Rewriting theory has to be generalized
in higher dimensions!

also: Burroni, Lafont, Métayer, Guiraud, Malbos, ...

The big plan

Rewriting theory has to be generalized
in higher dimensions!

also: Burroni, Lafont, Métayer, Guiraud, Malbos, ...

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

The big plan

Rewriting theory has to be generalized
in higher dimensions!

also: Burroni, Lafont, Métayer, Guiraud, Malbos, ...

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

Critical pairs

A **rewriting system** consists of

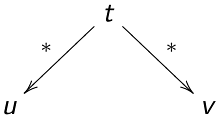
- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

Critical pairs

A **rewriting system** consists of

- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

The rewriting system is confluent when

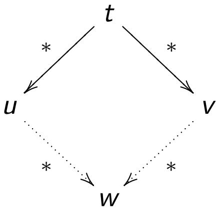


Critical pairs

A **rewriting system** consists of

- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

The rewriting system is confluent when

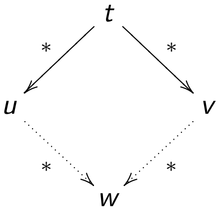


Critical pairs

A **rewriting system** consists of

- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

The rewriting system is confluent when



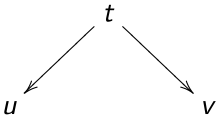
$$\boxed{\text{Confluence} + \text{termination} \quad \Rightarrow \quad \text{normal forms!}}$$

Critical pairs

A **rewriting system** consists of

- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

The rewriting system is locally confluent when

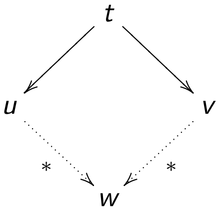


Critical pairs

A **rewriting system** consists of

- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

The rewriting system is locally confluent when

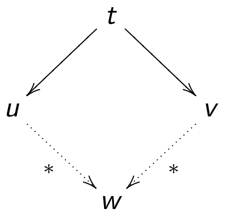


Critical pairs

A **rewriting system** consists of

- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

The rewriting system is locally confluent when



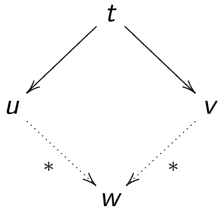
$$\boxed{\text{termination} + \text{local confluence} \quad \Rightarrow \quad \text{confluence}}$$

Critical pairs

A **rewriting system** consists of

- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

The rewriting system is locally confluent when



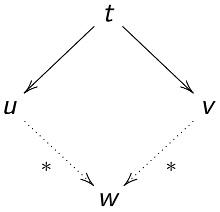
We only have to check *minimal* and *non-trivial* such t:
critical pairs

Critical pairs

A **rewriting system** consists of

- *terms*: t, u, ...
- *rewriting rules*: $r_i : t_i \to u_i$

The rewriting system is locally confluent when



We only have to check *minimal* and *non-trivial* such t:
critical pairs
which can be computed using a **unification algorithm**

So, we want to compute **critical pairs**
in higher dimensions

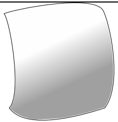
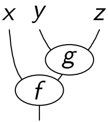
What is dimension?

	Geometry	Rewriting systems
0	$\cdot$	$\cdot x$

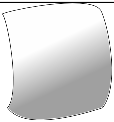
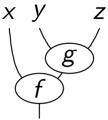
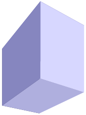
What is dimension?

	Geometry	Rewriting systems
0	$\cdot$	$\cdot x$
1		$\xrightarrow{a}\ \xrightarrow{b}\ \xrightarrow{c}$

What is dimension?

	Geometry	Rewriting systems
0	$\cdot$	$\cdot x$
1		$\xrightarrow{a}\xrightarrow{b}\xrightarrow{c}$
2		

What is dimension?

	Geometry	Rewriting systems
0	$\cdot$	$\cdot x$
1		$\xrightarrow{a} \xrightarrow{b} \xrightarrow{c}$
2		
3		???
...	...	???

Why rewriting systems?

- *Presentations*: giving finite descriptions of monoids, etc.
- *Coherence*: e.g. MacLane's coherence theorem for monoidal categories
- ...

This has to be generalized to n-categories

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

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

Example

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

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 locally 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 locally confluent.
4. Show that the normal forms are in bijection with M.

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

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 locally confluent.
4. Show that the normal forms are in bijection with M.

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

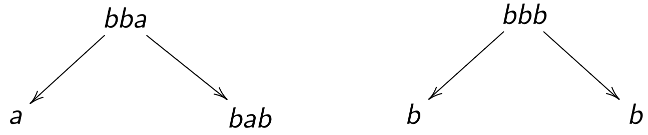
Presentations of monoids

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 locally confluent.
4. Show that the normal forms are in bijection with M.

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

Critical pairs are:



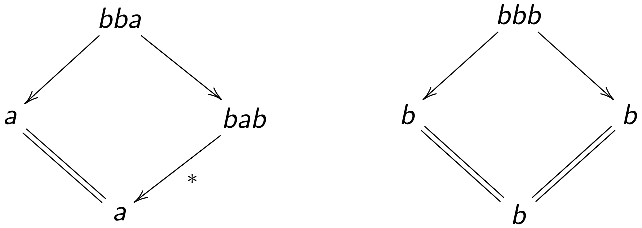
Presentations of monoids

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 locally confluent.
4. Show that the normal forms are in bijection with M.

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

Critical pairs are joinable:



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 locally confluent.
4. Show that the normal forms are in bijection with M.

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

Normal forms are:

a^n and $a^n b$

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

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 (Σ_2, Σ_3) induce LT whose morphisms $m \to n$ are n-uples of terms with variables $x_1, \ldots, x_m$, considered modulo relations.

Presentation: $\mathcal{C} \quad \cong \quad \Sigma_2^* / \equiv_{\Sigma_3}$

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 (Σ_2, Σ_3) induce LT whose morphisms $m \to n$ are n-uples of terms with variables $x_1, \ldots, x_m$, considered modulo relations.

Presentation: $\mathcal{C} \quad \cong \quad \Sigma_2^* / \equiv_{\Sigma_3}$

Example

Consider the TRS of commutative monoids: $\Sigma_2 = \{m : 2,\ e : 0\}$

$$\Sigma_3 = \left\{ \begin{array}{ll} \alpha : & m(m(x,y),z) \to m(x,m(y,z)) \\ \lambda : & m(e,x) \to x \\ \rho : & m(x,e) \to x \\ \gamma : & m(x,y) \to m(y,x) \end{array} \right\}$$

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 (Σ_2, Σ_3) induce LT whose morphisms $m \to n$ are n-uples of terms with variables $x_1, \ldots, x_m$, considered modulo relations.

Presentation: $\mathcal{C} \quad \cong \quad \Sigma_2^* / \equiv_{\Sigma_3}$

Example

Consider the TRS of commutative monoids: $\Sigma_2 = \{m : 2, \ e : 0\}$

It presents the Lawvere theory whose morphisms $M : m \to n$ 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 (Σ_2, Σ_3) induce LT whose morphisms $m \to n$ are n-uples of terms with variables $x_1, \ldots, x_m$, considered modulo relations.

Presentation: $\mathcal{C} \quad \cong \quad \Sigma_2^* / \equiv_{\Sigma_3}$

Example

Consider the TRS of commutative monoids: $\Sigma_2 = \{m : 2,\ e : 0\}$

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

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 (Σ_2, Σ_3) induce LT whose morphisms $m \to n$ are n-uples of terms with variables $x_1, \ldots, x_m$, considered modulo relations.

Presentation: $\mathcal{C} \quad \cong \quad \Sigma_2^* / \equiv_{\Sigma_3}$

Example

Consider the TRS of commutative monoids: $\Sigma_2 = \{m : 2,\ e : 0\}$

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

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 (Σ_2, Σ_3) induce LT whose morphisms $m \to n$ are n-uples of terms with variables $x_1, \ldots, x_m$, considered modulo relations.

Presentation: $\mathcal{C} \quad \cong \quad \Sigma_2^* / \equiv_{\Sigma_3}$

Example

Consider the TRS of commutative monoids: $\Sigma_2 = \{m : 2,\ e : 0\}$

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

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 (Σ_2, Σ_3) induce LT whose morphisms $m \to n$ are n-uples of terms with variables $x_1, \ldots, x_m$, considered modulo relations.

Presentation: $\mathcal{C} \quad \cong \quad \Sigma_2^* / \equiv_{\Sigma_3}$

Example

Consider the TRS of commutative monoids: $\Sigma_2 = \{m : 2,\ e : 0\}$

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

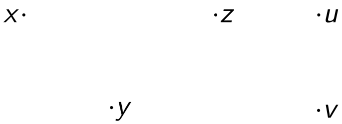
Use rewriting theory!

n-categories

We want to generalize rewriting systems
to present *n*-**categories**

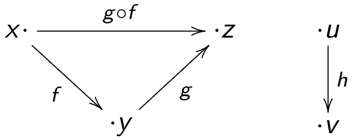
We want to generalize rewriting systems
to present n-**categories**

0-category:



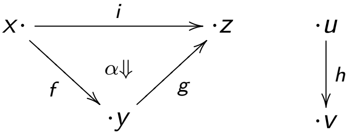
We want to generalize rewriting systems
to present *n*-**categories**

1-category:



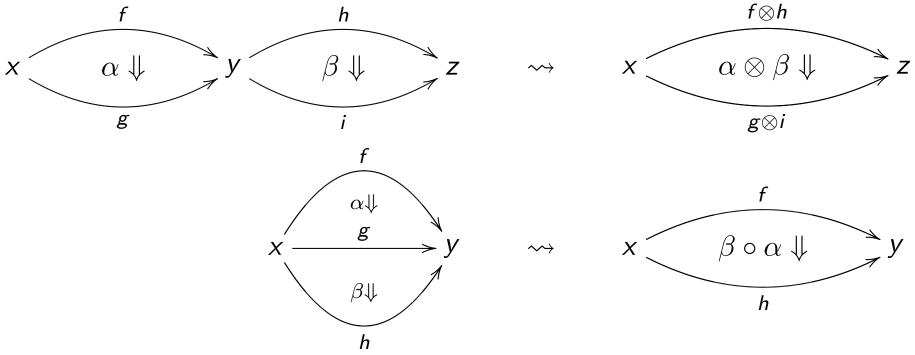
We want to generalize rewriting systems
to present *n*-**categories**

2-category:



We want to generalize rewriting systems
to present *n*-**categories**

2 compositions in a 2-category:



Presenting *n*-categories

We want to generalize rewriting systems

dimension	rewr. syst.	presents
1	string	monoid

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	0-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	monoid
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: $\xrightarrow{a} \xrightarrow{b} \quad \rightsquigarrow \quad x \xrightarrow{a} y \xrightarrow{b} y$

Presenting n-categories

We want to generalize rewriting systems

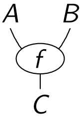
dimension	rewr. syst.	presents
0	element	0-category
1	string	1-category
2	term	Lawvere th.

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

Generalization:  $\rightsquigarrow$ 

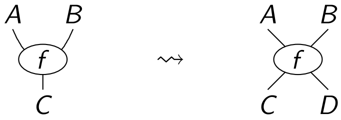
Presenting *n*-categories

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

Generalization:

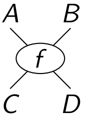


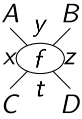
Presenting n-categories

We want to generalize rewriting systems

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

monoidal category $\quad = \quad$ 2-category with only one object

Generalization: 

$\rightsquigarrow$ 

Polygraphs

A 0-signature

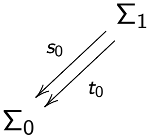
Σ_0

Example

signature

x y

A 0-rewriting system

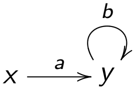


Example

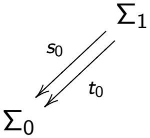
signature

x y

rules

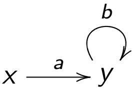


A 1-signature = a 0-rewriting system

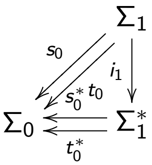


Example

signature

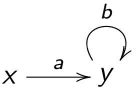


A 1-signature generates a *category*



Example

signature

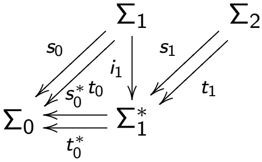


terms

$$x \xrightarrow{a} y \xrightarrow{b} y \xrightarrow{b} y$$

Polygraphs

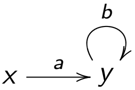
A 1-rewriting system



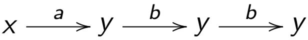
such that $s_0^* \circ s_1 = s_0^* \circ t_1$ and $t_0^* \circ s_1 = t_0^* \circ t_1$

Example

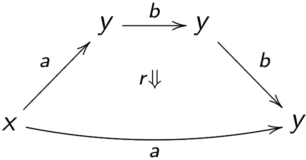
signature



terms

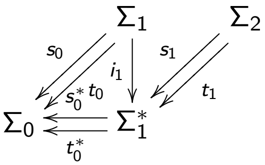


rules



Polygraphs

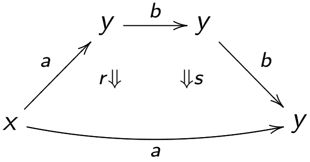
A 2-signature = a 1-rewriting system



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

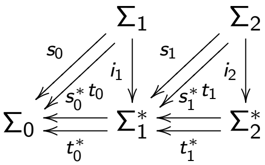
Example

signature



Polygraphs

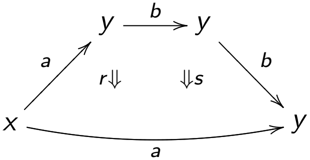
A 2-signature generates a 2-category



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

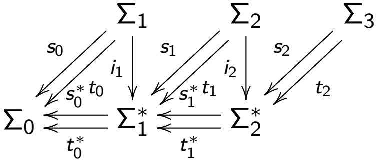
Example

signature



Polygraphs

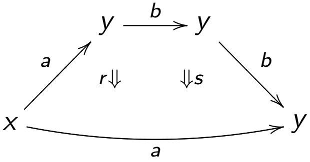
A 2-rewriting system



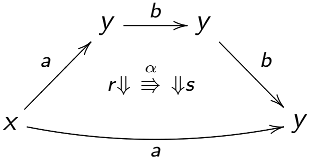
such that $s_1^* \circ s_2 = s_1^* \circ t_2$ and $t_1^* \circ s_2 = t_1^* \circ t_2$

Example

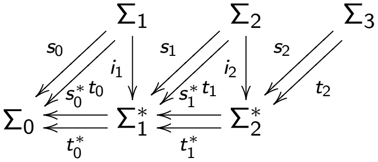
signature



rules



A 2-rewriting system



Right notion of *n-rewriting system*:

n-**polygraph**

[Street76,Power90,Burroni93]

A presentation of **Bij**

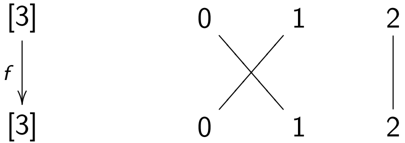
The category **Bij** has

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

A presentation of **Bij**

The category **Bij** has

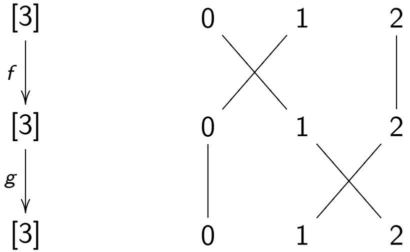
- objects are integers $[n] = \{0, \dots, n-1\}$
- morphisms $f : [m] \to [n]$ are bijections



A presentation of **Bij**

The category **Bij** has

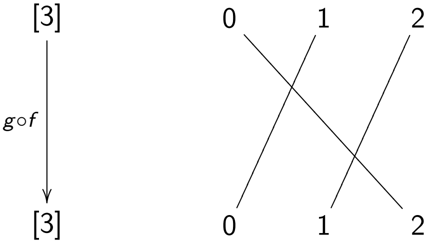
- objects are integers $[n] = \{0, \ldots, n-1\}$
- morphisms $f : [m] \to [n]$ are bijections



A presentation of **Bij**

The category **Bij** has

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

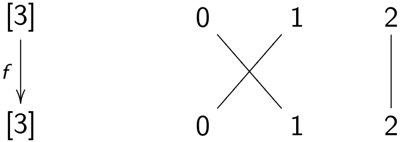


Vertical composition $\circ$

A presentation of **Bij**

The category **Bij** has

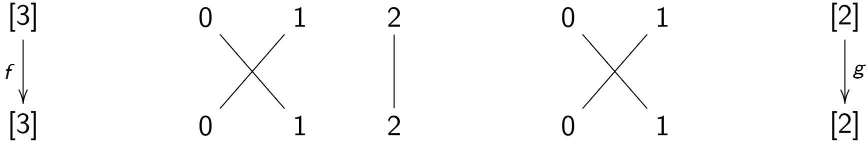
- objects are integers $[n] = \{0, \ldots, n-1\}$
- morphisms $f : [m] \to [n]$ are bijections



A presentation of **Bij**

The category **Bij** has

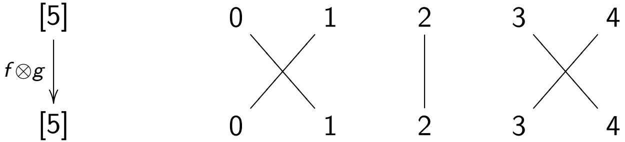
- objects are integers $[n] = \{0, \ldots, n-1\}$
- morphisms $f : [m] \to [n]$ are bijections



A presentation of **Bij**

The category **Bij** has

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



Horizontal composition $\otimes$

We want to give a presentation of **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 **Bij**

Bij is presented by the 3-polygraph such that [Lafont03]

- $\Sigma_0 = \{*\}$

A presentation of **Bij**

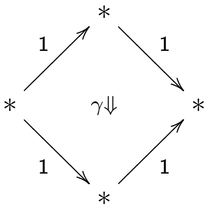
Bij is presented by the 3-polygraph such that [Lafont03]

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

A presentation of **Bij**

Bij is presented by the 3-polygraph such that [Lafont03]

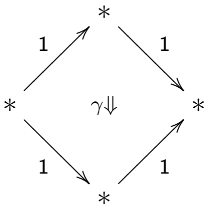
- $\Sigma_0 = \{*\}$
- $\Sigma_1 = \{1 : * \to *\}$ (so $\Sigma_1^* \approx \mathbb{N}$)
- $\Sigma_2 = \{\gamma : 1 \otimes 1 \to 1 \otimes 1\}$

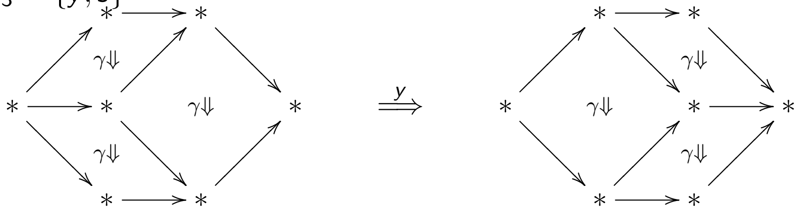


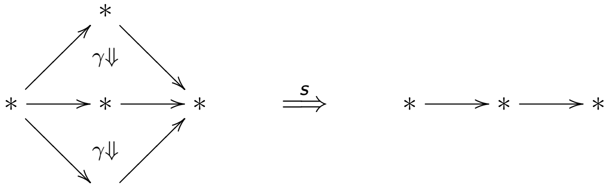
A presentation of **Bij**

Bij is presented by the 3-polygraph such that [Lafont03]

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

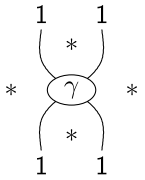




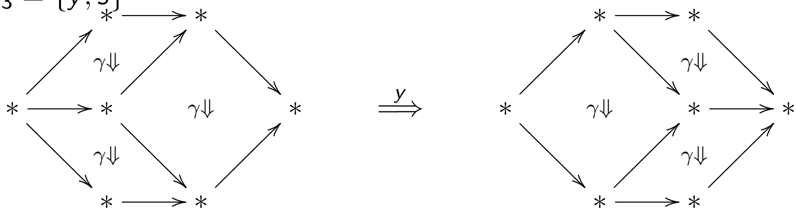


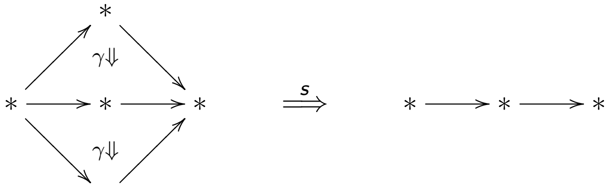
A presentation of **Bij**

Bij is presented by the 3-polygraph such that [Lafont03]

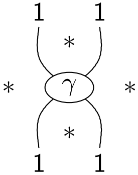


- $\Sigma_0 = \{*\}$
- $\Sigma_1 = \{1 : * \to *\}$ (so $\Sigma_1^* \approx \mathbb{N}$)
- $\Sigma_2 = \{\gamma : 1 \otimes 1 \to 1 \otimes 1\}$
- $\Sigma_3 = \{y, s\}$

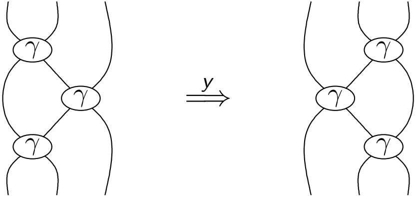


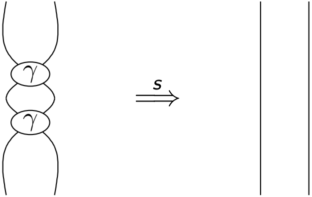


Bij is presented by the 3-polygraph such that [Lafont03]



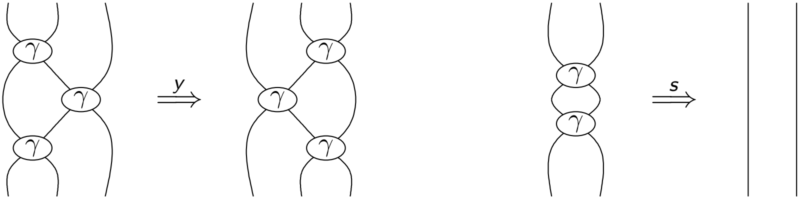
- $\Sigma_0 = \{*\}$
- $\Sigma_1 = \{1 : * \to *\}$ (so $\Sigma_1^* \approx \mathbb{N}$)
- $\Sigma_2 = \{\gamma : 1 \otimes 1 \to 1 \otimes 1\}$
- $\Sigma_3 = \{y, s\}$



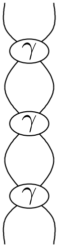


A presentation of **Bij**

The rules

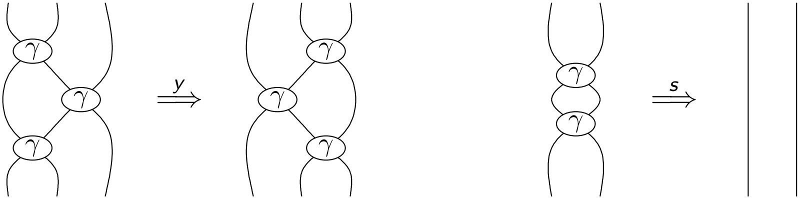


of the rewriting system induce critical pairs

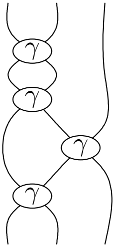


A presentation of **Bij**

The rules

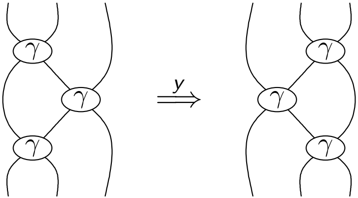


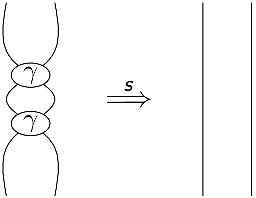
of the rewriting system induce critical pairs



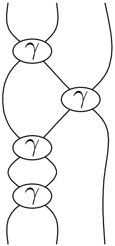
A presentation of **Bij**

The rules



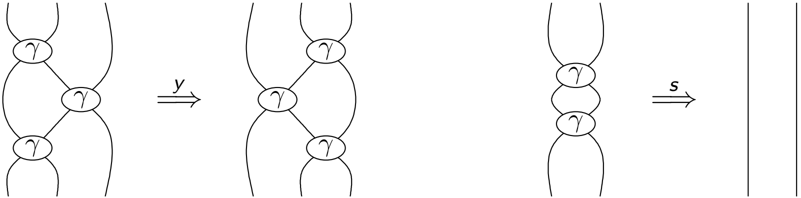


of the rewriting system induce critical pairs

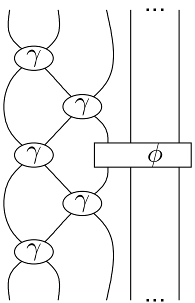


A presentation of **Bij**

The rules



of the rewriting system induce critical pairs

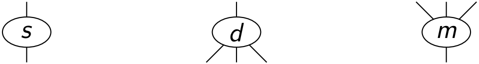


Two problems with critical pairs

Consider the 2-rewriting system Σ with

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

the generators for 2-cells are drawn respectively as

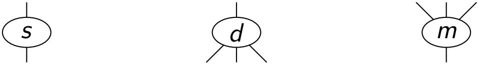


Two problems with critical pairs

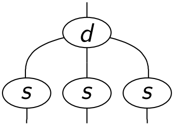
Consider the 2-rewriting system Σ with

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

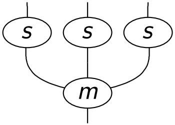
the generators for 2-cells are drawn respectively as



with rules



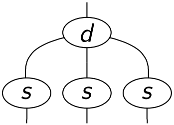
$\Rightarrow \ldots$ and



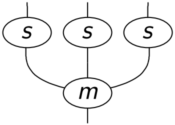
$\Rightarrow \ldots$

Two problems with critical pairs

The two rules



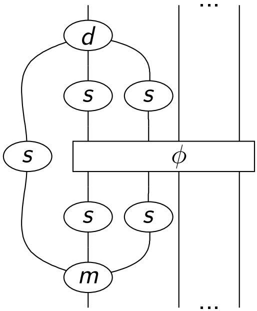
$\Rightarrow \ldots$ and



$\Rightarrow \ldots$

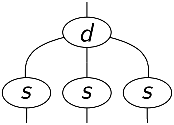
induce an infinite number of critical pairs:

variables on the border:

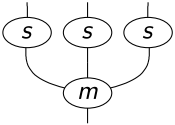


Two problems with critical pairs

The two rules



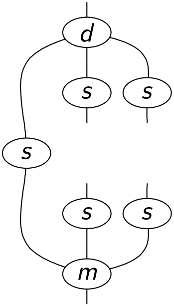
$\Rightarrow \ldots$ and



$\Rightarrow \ldots$

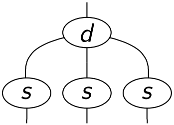
induce an infinite number of critical pairs:

variables on the border:
use compact morphisms!

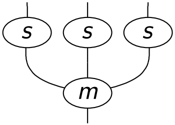


Two problems with critical pairs

The two rules



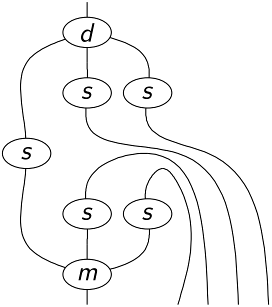
$\Rightarrow \ldots$ and



$\Rightarrow \ldots$

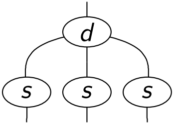
induce an infinite number of critical pairs:

variables on the border:
use compact morphisms!

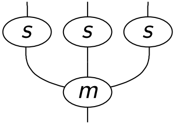


Two problems with critical pairs

The two rules



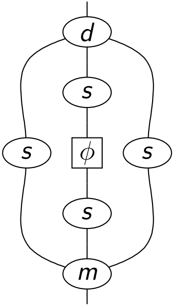
$\Rightarrow \dots$ and



$\Rightarrow \dots$

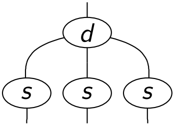
induce an infinite number of critical pairs:

variables inside:

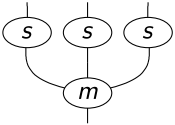


Two problems with critical pairs

The two rules



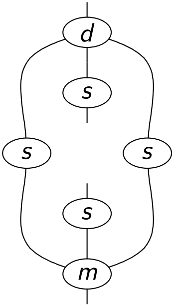
$\Rightarrow \dots$ and



$\Rightarrow \dots$

induce an infinite number of critical pairs:

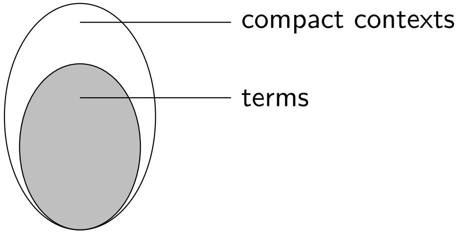
variables inside:
use contexts!
(= terms with metavariables)



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****.*



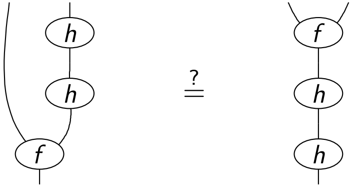
In other words, there is a finite number of *generating families* of critical pairs in those rewriting systems.

Unification in TRS

Suppose that we have a TRS

$f : 2 \qquad h : 1 \qquad\qquad f(x, h(h(y))) \Rightarrow \ldots \qquad h(h(f(x, y))) \Rightarrow \ldots$

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

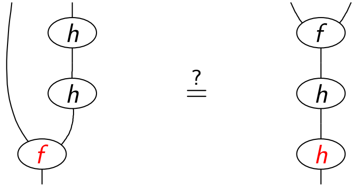


Unification in TRS

Suppose that we have a TRS

$f : 2 \qquad h : 1 \qquad\qquad f(x, h(h(y))) \Rightarrow \ldots \qquad h(h(f(x, y))) \Rightarrow \ldots$

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

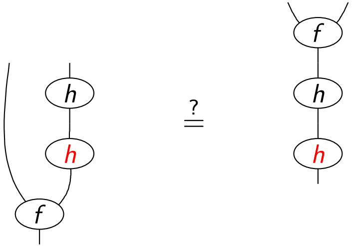


Unification in TRS

Suppose that we have a TRS

$f : 2 \qquad h : 1 \qquad f(x, h(h(y))) \Rightarrow \ldots \qquad h(h(f(x, y))) \Rightarrow \ldots$

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

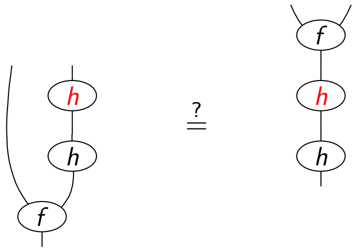


Unification in TRS

Suppose that we have a TRS

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

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

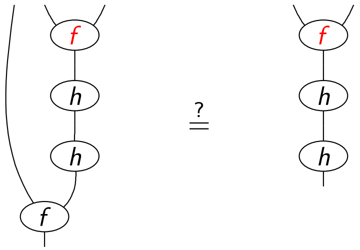


Unification in TRS

Suppose that we have a TRS

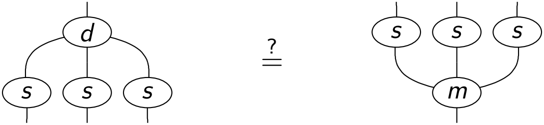
$f : 2 \qquad h : 1 \qquad\qquad f(x, h(h(y))) \Rightarrow \ldots \qquad h(h(f(x, y))) \Rightarrow \ldots$

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



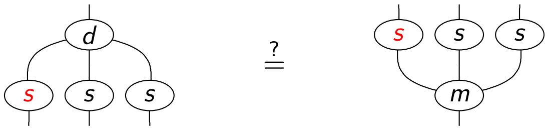
Unification in 2-rewriting systems

We want to compute the critical pairs generated by



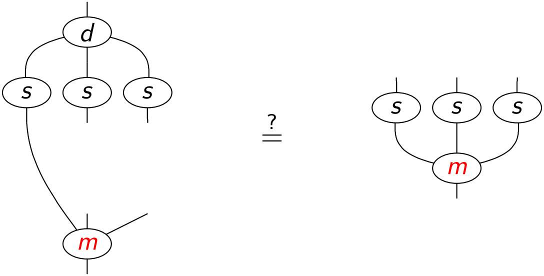
Unification in 2-rewriting systems

We want to compute the critical pairs generated by



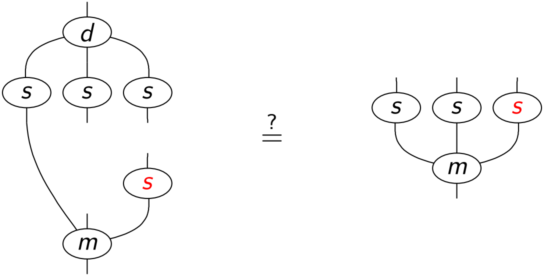
Unification in 2-rewriting systems

We want to compute the critical pairs generated by



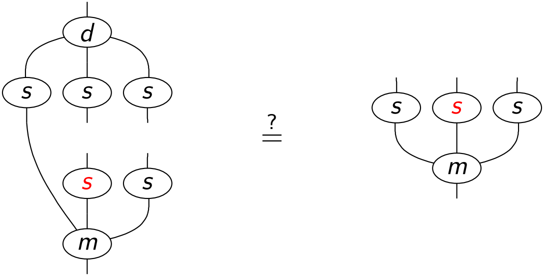
Unification in 2-rewriting systems

We want to compute the critical pairs generated by



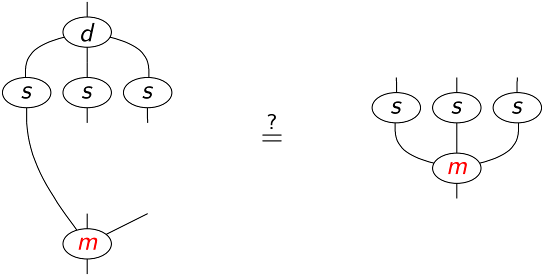
Unification in 2-rewriting systems

We want to compute the critical pairs generated by



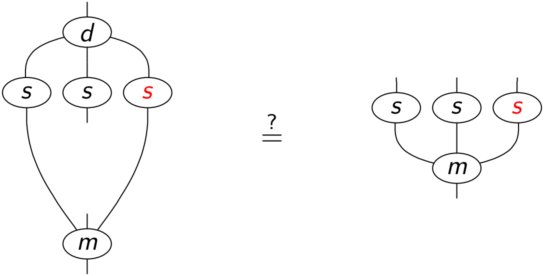
Unification in 2-rewriting systems

We want to compute the critical pairs generated by



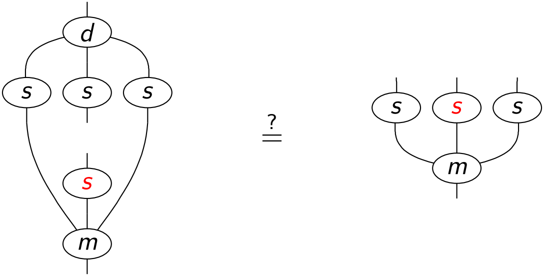
Unification in 2-rewriting systems

We want to compute the critical pairs generated by



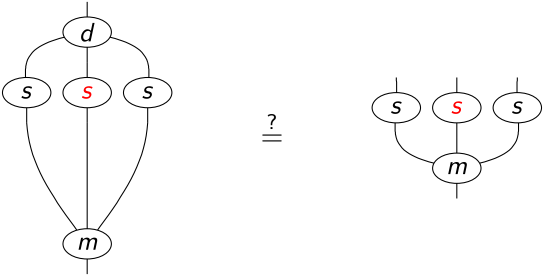
Unification in 2-rewriting systems

We want to compute the critical pairs generated by



Unification in 2-rewriting systems

We want to compute the critical pairs generated by



Under the carpet

- Precise definition of compact contexts
- Formal algebraic definition of morphisms in free categories
- Precise statement of the algorithm

TODO

- Generalize termination techniques to this setting (cf. [Guiraud])
- Generalize to higher dimensions
- Automated tools for studying categories
- ...