GEOMETRIC INVARIANTS OF ALGEBRAIC STRUCTURES

Samuel Mimram

École Polytechnique

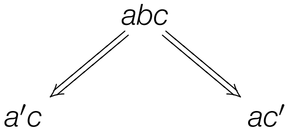
ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Sémin'ouvert

April 20th, 2017

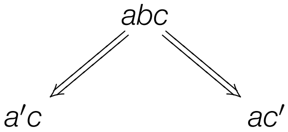
Geometric invariants of concurrent computations

- We consider a very simple “concurrent programming language”: **string rewriting systems**



Geometric invariants of concurrent computations

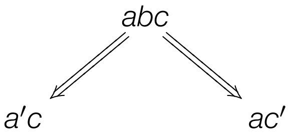
- We consider a very simple "concurrent programming language": **string rewriting systems**



- We are interested in the **geometry** of the space of possible computations (and not in computing geometric invariants)

Geometric invariants of concurrent computations

- We consider a very simple "concurrent programming language": **string rewriting systems**

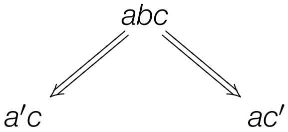


- We are interested in the **geometry** of the space of possible computations (and not in computing geometric invariants)

- We will explain **Squier's theorem**:
  an impossibility result based on geometric invariants

Geometric invariants of concurrent computations

- We consider a very simple "concurrent programming language": **string rewriting systems**



- We are interested in the **geometry** of the space of possible computations (and not in computing geometric invariants)

- We will explain **Squier's theorem**:
  an impossibility result based on geometric invariants

- This generalizes to **term rewriting systems**

Squier's result in a nutshell

When a rewriting system satisfies good properties (*confluence*) the computation will always give rise to the same result in the end.

Squier's result in a nutshell

When a rewriting system satisfies good properties (*confluence*) the computation will always give rise to the same result in the end.

Can we always transform a finite rewriting system into an "equivalent" one which is confluent?

Squier's result in a nutshell

When a rewriting system satisfies good properties (*confluence*) the computation will always give rise to the same result in the end.

Can we always transform a finite rewriting system into an "equivalent" one which is confluent?

Squier: NO

Let's go.

Monoids

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

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

such that

- multiplication is associative

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

- unit is a neutral element

$$1 \cdot a \quad = \quad a \quad = \quad a \cdot 1$$

Example

- $(\mathbb{N}, +, 0)$

Monoids

Example

- $(\mathbb{N}, +, 0)$
- $(\mathbb{N}, \times, 1)$

Monoids

Example

- $(\mathbb{N}, +, 0)$
- $(\mathbb{N}, \times, 1)$
- given a set G, we have a free monoid $(G^*, \cdot, 1)$ of words ($\cdot$ is concatenation and 1 the empty word)

Example

- $(\mathbb{N}, +, 0)$
- $(\mathbb{N}, \times, 1)$
- given a set G, we have a free monoid $(G^*, \cdot, 1)$ of words ($\cdot$ is concatenation and 1 the empty word)
- every group is a monoid:

Example

- $(\mathbb{N}, +, 0)$
- $(\mathbb{N}, \times, 1)$
- given a set G, we have a free monoid $(G^*, \cdot, 1)$ of words ($\cdot$ is concatenation and 1 the empty word)
- every group is a monoid:
  - $\mathbb{Z}$, $\mathbb{Z}/n\mathbb{Z}$

Monoids

Example

- $(\mathbb{N}, +, 0)$
- $(\mathbb{N}, \times, 1)$
- given a set G, we have a free monoid $(G^*, \cdot, 1)$ of words ($\cdot$ is concatenation and 1 the empty word)
- every group is a monoid:
  - $\mathbb{Z}$, $\mathbb{Z}/n\mathbb{Z}$
  - S_n: group of permutations of n elements

Monoids

Example

- $(\mathbb{N}, +, 0)$
- $(\mathbb{N}, \times, 1)$
- given a set G, we have a free monoid $(G^*, \cdot, 1)$ of words ($\cdot$ is concatenation and 1 the empty word)
- every group is a monoid:
  - $\mathbb{Z}$, $\mathbb{Z}/n\mathbb{Z}$
  - S_n: group of permutations of n elements
  - B_n: group of braids with n strands

Example

- $(\mathbb{N}, +, 0)$
- $(\mathbb{N}, \times, 1)$
- given a set G, we have a free monoid $(G^*, \cdot, 1)$ of words ($\cdot$ is concatenation and 1 the empty word)
- every group is a monoid:
  - $\mathbb{Z}$, $\mathbb{Z}/n\mathbb{Z}$
  - S_n: group of permutations of n elements
  - B_n: group of braids with n strands
- etc.

Congruence on a monoid

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

$$b \approx b' \qquad \text{implies} \qquad a \cdot b \cdot c \approx a \cdot b' \cdot c$$

Congruence on a monoid

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

$$b \approx b' \qquad \text{implies} \qquad a \cdot b \cdot c \approx a \cdot b' \cdot c$$

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

$$M/\!\approx$$

as expected.

We can come up
with small descriptions
of monoids.

Presentations of monoids

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

Presentations of monoids

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

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

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

where

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

such that

$$M \quad \cong \quad G^*/{\approx_R}$$

where $\approx_R$ is the smallest congruence such that

$$(u, v) \in R \qquad \text{implies} \qquad u \approx_R v$$

Presentations of monoids

Example

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

$$\langle a \mid \rangle$$

Presentations of monoids

Example

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

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

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

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

Presentations of monoids

Example

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

$$\langle a \mid \rangle$$

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

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

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

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

Presentations of monoids

Example

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

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

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

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

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

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

- S_3 is presented by

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

Presentations of monoids

A monoid M admits a presentation $\langle G \mid R \rangle$ means that

- the elements of G generate the monoid:
  any element of M can be obtained as a product of those

Presentations of monoids

A monoid M admits a presentation $\langle G \mid R \rangle$ means that

- the elements of G generate the monoid:
  any element of M can be obtained as a product of those
- R generate equality: given $u, v \in G^*$ whose evaluation in M is the same, we have $u \approx v$

Presentations of monoids

A monoid M admits a presentation $\langle G \mid R \rangle$ means that

- the elements of G generate the monoid:
  any element of M can be obtained as a product of those
- R generate equality: given $u, v \in G^*$ whose evaluation in M is the same, we have $u \approx v$

For $\mathbb{N} \times \mathbb{N}$ presented by $\langle a, b \mid ba = ab \rangle$, we have

Presentations of monoids

A monoid M admits a presentation $\langle G \mid R \rangle$ means that

- the elements of G generate the monoid:
  any element of M can be obtained as a product of those
- R generate equality: given $u, v \in G^*$ whose evaluation in M is the same, we have $u \approx v$

For $\mathbb{N} \times \mathbb{N}$ presented by $\langle a, b \mid ba = ab \rangle$, we have

- any element can be obtained as a sum of

$$a = (1,0) \qquad \text{and} \qquad b = (0,1)$$

Presentations of monoids

A monoid M admits a presentation $\langle G \mid R \rangle$ means that

- the elements of G generate the monoid:
  any element of M can be obtained as a product of those
- R generate equality: given $u, v \in G^*$ whose evaluation in M is the same, we have $u \approx v$

For $\mathbb{N} \times \mathbb{N}$ presented by $\langle a, b \mid ba = ab \rangle$, we have

- any element can be obtained as a sum of

$$a = (1, 0) \qquad \text{and} \qquad b = (0, 1)$$

- equality is generated by ab:

$$baa = (0,1)+(1,0)+(1,0) = (2,1) = (1,0)+(1,0)+(0,1) = aab$$

and

$$baa \quad \approx \quad aba \quad \approx \quad aab$$

Presentations of monoids

Note that every monoid M admits a presentation:

- *generators*: take $G = M$
- *relations*: all pairs $(u, v) \in G^* \times G^*$ such that $u = v$ in M, i.e.

$$u_1 \times \ldots \times u_m \quad = \quad v_1 \times \ldots \times v_n$$

We are mostly interested in small (at least finite) ones.

How do we show
that we actually have
a presentation?

Constructing presentations of monoids

For instance,

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

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

Constructing presentations of monoids

For instance,

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

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

In each equivalence class (w.r.t. $\approx$) there is a unique word of the form

$$a^m b^n$$

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

For instance,

$$abaa \quad \approx \quad aaba \quad \approx \quad aaab$$

Inventing *canonical forms*
can be difficult
let's see a generic method.

String rewriting systems

A **string rewriting systems** $\langle G \mid R \rangle$ consists of

- an *alphabet* G
- a set of *rules* $R \subseteq G^* \times G^*$

String rewriting systems

A **string rewriting systems** $\langle G \mid R \rangle$ consists of

- an *alphabet* G
- a set of *rules* $R \subseteq G^* \times G^*$

A rule (v, v') is interpreted as v' being "more canonical" than v.

String rewriting systems

A **string rewriting systems** $\langle G \mid R \rangle$ consists of

- an *alphabet* G
- a set of *rules* $R \subseteq G^* \times G^*$

A rule (v, v') is interpreted as v' being "more canonical" than v.

A **rewriting step** is a pair of the form

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

from some rule $(v, v') \in R$ and words $u, w \in G^*$.

String rewriting systems

A **string rewriting systems** $\langle G \mid R \rangle$ consists of

- an *alphabet* G
- a set of *rules* $R \subseteq G^* \times G^*$

A rule (v, v') is interpreted as v' being "more canonical" than v.

A **rewriting step** is a pair of the form

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

from some rule $(v, v') \in R$ and words $u, w \in G^*$.

A **rewriting path** $u \overset{*}{\Rightarrow} v$ is a sequence of rewriting steps, and we say that u **rewrites** to v.

String rewriting systems

A **string rewriting systems** $\langle G \mid R \rangle$ consists of

- an *alphabet* G
- a set of *rules* $R \subseteq G^* \times G^*$

A rule (v, v') is interpreted as v' being "more canonical" than v.

A **rewriting step** is a pair of the form

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

from some rule $(v, v') \in R$ and words $u, w \in G^*$.

A **rewriting path** $u \overset{*}{\Rightarrow} v$ is a sequence of rewriting steps, and we say that u **rewrites** to v.

Lemma

$u \overset{*}{\Rightarrow} v$ implies $u \approx v$.
$\approx_R$ is the symmetric and transitive closure of $\overset{*}{\Rightarrow}$.

String rewriting systems

Example

In the rewriting system

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

we have the rewriting path

$$abaa \quad \Rightarrow \quad aaba \quad \Rightarrow \quad aaab$$

Normal forms

A **normal form** u is a word which rewrites only to itself: there is no v such that

$$u \quad \Rightarrow \quad v$$

These are "maximally canonical" words.

Normal forms

A **normal form** u is a word which rewrites only to itself:
there is no v such that

$$u \quad \Rightarrow \quad v$$

These are "maximally canonical" words.

Can we ensure that every equivalence class contains exactly one normal form?

Termination

A rewriting system is **terminating** if there is no infinite sequence

$$u \quad \Rightarrow \quad u_1 \quad \Rightarrow \quad u_2 \quad \Rightarrow \quad \ldots$$

of rewriting steps.

Termination

A rewriting system is **terminating** if there is no infinite sequence

$$u \quad \Rightarrow \quad u_1 \quad \Rightarrow \quad u_2 \quad \Rightarrow \quad \ldots$$

of rewriting steps.

Lemma

In this case, every equivalence class contains at least one normal form.

Proof.

Given an element u of an equivalence class, rewrite it as much as possible. □

Example

The rewriting system

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

is terminating (because rules put *b*s on the right).

Termination

Example

The rewriting system

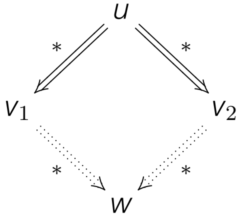
$$\langle a, b \mid ba \Rightarrow ab \rangle$$

is terminating (because rules put *b*s on the right).

A normal form for *abaa* is *aaab*:

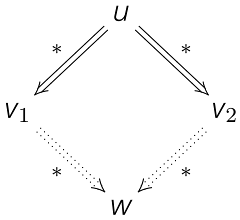
$$abaa \quad \Rightarrow \quad aaba \quad \Rightarrow \quad aaab$$

A rewriting system is **confluent** if



Confluence

A rewriting system is **confluent** if



Lemma (Church-Rosser'36)

In a confluent rewriting system any equivalence class contains at most one normal form.

Convergent rewriting systems

A rewriting system is **convergent** when it is

- terminating
- confluent

Lemma

In such a system, every equivalence class of a word u admits exactly one representative in normal form $\hat{u}$.

The word problem

In a convergent rewriting system is easy to decide the **word problem** for a presentation:

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

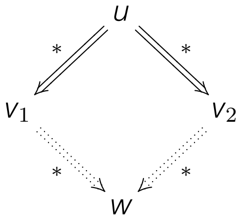
Namely:

1. rewrite u to its normal form $\hat{u}$
2. rewrite v to its normal form $\hat{v}$
3. return $\hat{u} = \hat{v}$

How do we show confluence in practice?

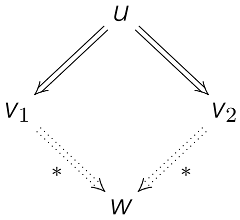
Local confluence

A rewriting system is **confluent** if



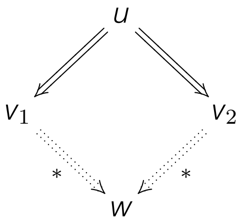
Local confluence

A rewriting system is **locally confluent** if



Local confluence

A rewriting system is **locally confluent** if



Lemma (Newman'42)

For terminating rewriting systems, confluence is equivalent to local confluence.

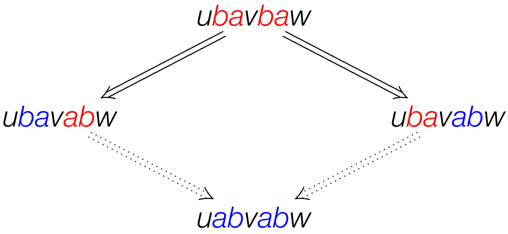
Critical branchings

We can further reduce the number of local branchings to check.

Critical branchings

We can further reduce the number of local branchings to check.

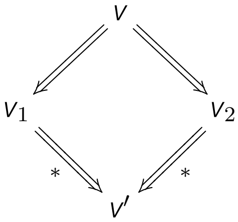
Independent branchings.
Consider the rule $ba \Rightarrow ab$, then we have



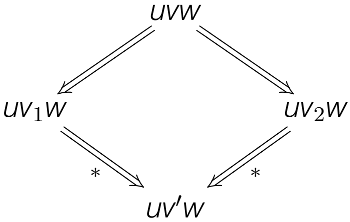
Critical branchings

We can further reduce the number of local branchings to check.

Non-minimal branchings.



implies



Critical branchings

For this reason, we can restrict to **critical branchings**, which are those being

- overlapping (= not independent)
- minimal (wrt to context)

Critical branchings

For this reason, we can restrict to **critical branchings**, which are those being

- overlapping (= not independent)
- minimal (wrt to context)

Lemma

A terminating rewriting system with confluent critical branchings is convergent.

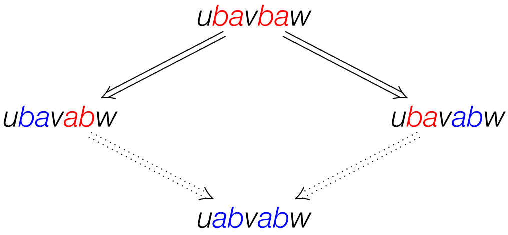
Critical branchings

Example

In the rewriting system

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

all branchings are of the form



i.e. there is no critical branching.

It is thus convergent and normal forms are words $a^m b^n$.

Critical branchings

Example

Consider the rewriting system

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

Critical branchings

Example

Consider the rewriting system

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

The critical pairs are

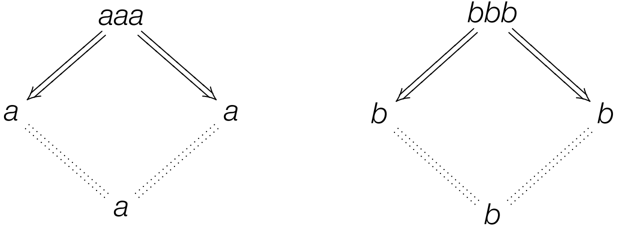
Critical branchings

Example

Consider the rewriting system

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

The critical pairs are



Example

Consider the rewriting system

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

The critical pairs are

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

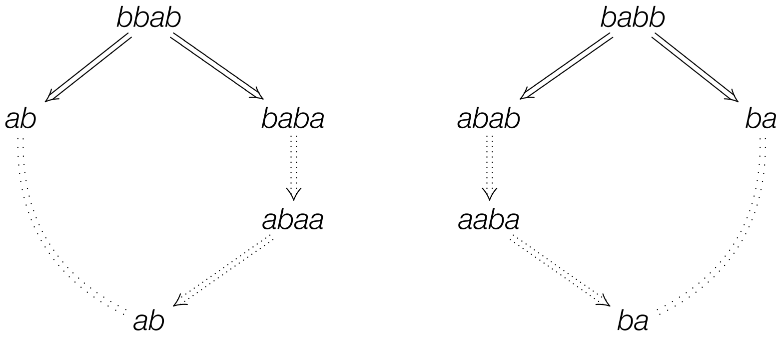
Critical branchings

Example

Consider the rewriting system

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

The critical pairs are



Example

Consider the rewriting system

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

The rewriting system is terminating and thus convergent.

Normal forms are

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

from which we can deduce that this is a presentation of S_3 (you can already check that there are $6 = 3!$ elements).

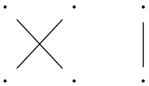
Critical branchings

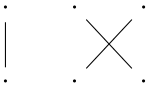
Example

Consider the rewriting system

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

The generators *a* and *b* respectively correspond to

$a =$ 

$b =$ 

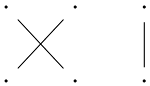
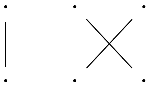
Critical branchings

Example

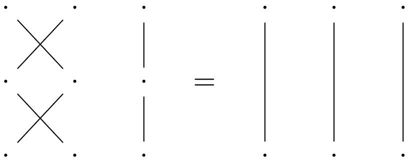
Consider the rewriting system

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

The generators *a* and *b* respectively correspond to

$a =$  $b =$ 

The relation $aa = 1$ is



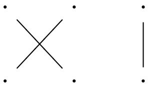
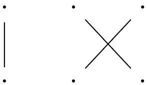
Critical branchings

Example

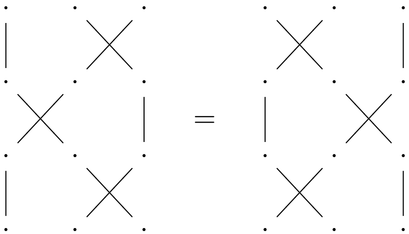
Consider the rewriting system

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

The generators *a* and *b* respectively correspond to

$a =$  $b =$ 

The relation $bab = aba$ is



Lemma

Given a finite rewriting system $\langle G \mid R \rangle$ (both G and R finite), there is a finite number of critical branchings.

Proof.

We have an algorithm for computing critical pairs:

- for every pair of rules $u_1 \Rightarrow v_1$ and $u_2 \Rightarrow v_2$
- compute all the ways u_1 and u_2 can overlap □

Does this solve
all the problems
in the world?

Universality of convergent rewriting

The **word problem**: do we have $u \approx v$?

Universality of convergent rewriting

The **word problem**: do we have $u \approx v$?

For convergent presentations, this is easy: $\hat{u} = \hat{v}$?

Universality of convergent rewriting

The **word problem**: do we have $u \approx v$?

For convergent presentations, this is easy: $\hat{u} = \hat{v}$?

Universality of convergent rewriting: does every finitely presented monoid with decidable word problem admit a finite convergent presentation?

When do two presentations present the same monoid?

Tietze transformations

The **Tietze transformations** preserve the presented monoid:

1. add a definable generator:

$$\langle G \mid R \rangle \qquad \rightsquigarrow \qquad \langle G, a \mid R, u = a \rangle$$

   with $u \in G^*$,

Tietze transformations

The **Tietze transformations** preserve the presented monoid:

1. add a definable generator:

$$\langle G \mid R \rangle \quad \rightsquigarrow \quad \langle G, a \mid R, u = a \rangle$$

   with $u \in G^*$,

2. remove a definable generator:

$$\langle G, a \mid R, u = a \rangle \quad \rightsquigarrow \quad \langle G \mid R \rangle$$

   where a does not occur in R,

Tietze transformations

The **Tietze transformations** preserve the presented monoid:

1. add a definable generator:

$$\langle G \mid R \rangle \qquad \rightsquigarrow \qquad \langle G, a \mid R, u = a \rangle$$

   with $u \in G^*$,

2. remove a definable generator:

$$\langle G, a \mid R, u = a \rangle \qquad \rightsquigarrow \qquad \langle G \mid R \rangle$$

   where a does not occur in R,

3. add a derivable relation:

$$\langle G \mid R \rangle \qquad \rightsquigarrow \qquad \langle G \mid R, u = v \rangle$$

   when $u \approx_R v$,

Tietze transformations

The **Tietze transformations** preserve the presented monoid:

1. add a definable generator:

$$\langle G \mid R \rangle \qquad \rightsquigarrow \qquad \langle G, a \mid R, u = a \rangle$$

   with $u \in G^*$,

2. remove a definable generator:

$$\langle G, a \mid R, u = a \rangle \qquad \rightsquigarrow \qquad \langle G \mid R \rangle$$

   where a does not occur in R,

3. add a derivable relation:

$$\langle G \mid R \rangle \qquad \rightsquigarrow \qquad \langle G \mid R, u = v \rangle$$

   when $u \approx_R v$,

4. remove a derivable relation:

$$\langle G \mid R, u = v \rangle \qquad \rightsquigarrow \qquad \langle G \mid R \rangle$$

   when $u \approx_R v$.

Tietze transformations

Theorem

Two presentations present the same monoid if and only if they are related by a series of Tietze transformations.

For instance, consider the presentation

$$\langle a, b \mid bab = aba \rangle$$

we can apply the following series of transformations:

- $\langle a, b \mid bab = aba \rangle$

For instance, consider the presentation

$$\langle a, b \mid bab = aba \rangle$$

we can apply the following series of transformations:

- $\langle a, b \mid bab = aba \rangle$
- $\langle a, b, c \mid bab = aba, ba = c \rangle$

For instance, consider the presentation

$$\langle a, b \mid bab = aba \rangle$$

we can apply the following series of transformations:

- $\langle a, b \mid bab = aba \rangle$
- $\langle a, b, c \mid bab = aba, ba = c \rangle$
- $\langle a, b, c \mid bab = aba, ab = c, cb = ac \rangle$

For instance, consider the presentation

$$\langle a, b \mid bab = aba \rangle$$

we can apply the following series of transformations:

- $\langle a, b \mid bab = aba \rangle$
- $\langle a, b, c \mid bab = aba, ba = c \rangle$
- $\langle a, b, c \mid bab = aba, ab = c, cb = ac \rangle$
- $\langle a, b, c \mid ab = c, cb = ac \rangle$

For instance, consider the presentation

$$\langle a, b \mid bab = aba \rangle$$

we can apply the following series of transformations:

- $\langle a, b \mid bab = aba \rangle$
- $\langle a, b, c \mid bab = aba, ba = c \rangle$
- $\langle a, b, c \mid bab = aba, ab = c, cb = ac \rangle$
- $\langle a, b, c \mid ab = c, cb = ac \rangle$

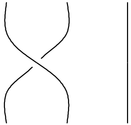
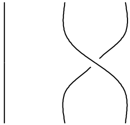
And we obtain a convergent rewriting system:

$$\langle a, b, c \mid ab \Rightarrow c, cb \Rightarrow ac \rangle$$

We can deduce that the presentation

$$\langle a, b \mid bab = aba \rangle$$

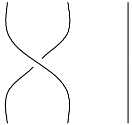
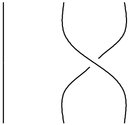
corresponds to B_3, the monoid of braids on 3 strands:

$a =$  $b =$ 

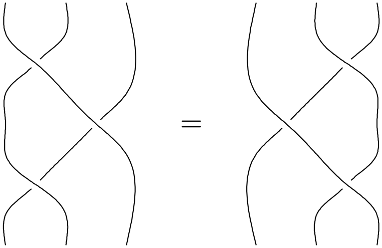
We can deduce that the presentation

$$\langle a, b \mid bab = aba \rangle$$

corresponds to B_3, the monoid of braids on 3 strands:

$a =$  $b =$ 

We have the relation $bab = aba$:



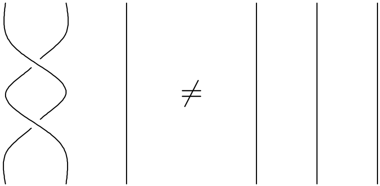
Braids

We can deduce that the presentation

$$\langle a, b \mid bab = aba \rangle$$

corresponds to B_3, the monoid of braids on 3 strands:

But not the relation $aa = 1$:

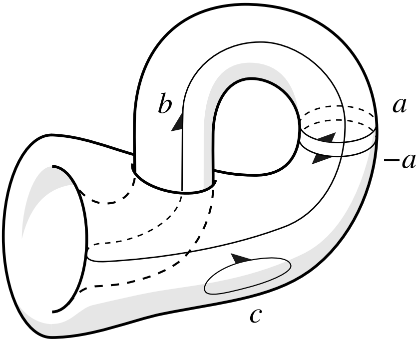


Studying all the presentations
of a given monoid
to determine whether there is
a *convergent* one
is difficult!

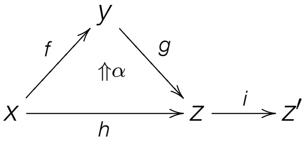
Let's switch to something else...

Suppose that you have a space (e.g. a simplicial complex) and you want to compute the number of “holes” in it. There is a very efficient way of doing this:

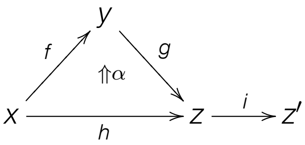
homology



Suppose that our space looks like this:



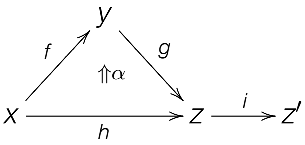
Suppose that our space looks like this:



- we allow taking linear combinations of “building blocks”

Homology

Suppose that our space looks like this:

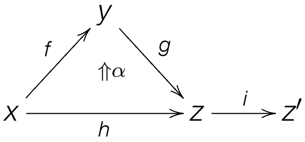


- we allow taking linear combinations of "building blocks"
- we define the boundary of a block as target - source:

$$\partial(f) = y - x \qquad \partial(\alpha) = f + g - h$$

Homology

Suppose that our space looks like this:



- we allow taking linear combinations of “building blocks”
- we define the boundary of a block as target - source:

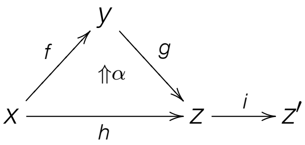
$$\partial(f) = y - x \qquad\qquad \partial(\alpha) = f + g - h$$

- “potential holes” can be detected as those with empty boundary:

$$\begin{aligned}\partial(f + g - h) &= \partial(f) + \partial(g) - \partial(h)\\ &= (y - x) + (z - y) - (z - x) = 0\end{aligned}$$

Homology

Suppose that our space looks like this:



- we allow taking linear combinations of “building blocks”
- we define the boundary of a block as target - source:

$$\partial(f) = y - x \qquad\qquad \partial(\alpha) = f + g - h$$

- “potential holes” can be detected as those with empty boundary:

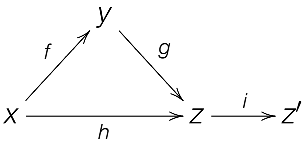
$$\begin{aligned}\partial(f + g - h) &= \partial(f) + \partial(g) - \partial(h)\\ &= (y - x) + (z - y) - (z - x) = 0\end{aligned}$$

- we have to remove those that are boundaries

$$\partial(\alpha) = f + g - h$$

Homology

Suppose that our space looks like this:



- we allow taking linear combinations of “building blocks”
- we define the boundary of a block as target - source:

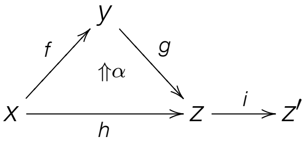
$$\partial(f) = y - x \qquad\qquad \partial(\alpha) = f + g - h$$

- “potential holes” can be detected as those with empty boundary:

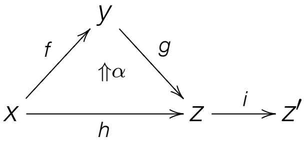
$$\begin{aligned}\partial(f + g - h) &= \partial(f) + \partial(g) - \partial(h)\\ &= (y - x) + (z - y) - (z - x) = 0\end{aligned}$$

- we have to remove those that are boundaries

Formally, given our space X:



Formally, given our space X:



we consider the **chain complex**

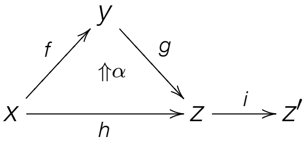
$$\ldots \xrightarrow{\partial_2} \mathbb{k}\{\alpha\} \xrightarrow{\partial_1} \mathbb{k}\{f,g,h,i\} \xrightarrow{\partial_0} \mathbb{k}\{x,y,z,z'\}$$

$$\begin{matrix} \mathbb{k}\{\alpha\} & & \mathbb{k}\{f,g,h,i\} & & \mathbb{k}\{x,y,z,z'\} \\ \| & & \| & & \| \\ C_2 & & C_1 & & C_0 \end{matrix}$$

which means that

- the C_i are $\mathbb{k}$-vector spaces,
- the $\partial_i : C_{i+1} \to C_i$ are linear maps,
- we have $\partial_{i-1} \circ \partial_i = 0$ and thus $\operatorname{im} \partial_i \subseteq \ker \partial_{i-1}$.

Formally, given our space X:



we consider the **chain complex**

$$\dots \xrightarrow{\partial_2} \Bbbk\{\alpha\} \xrightarrow{\partial_1} \Bbbk\{f, g, h, i\} \xrightarrow{\partial_0} \Bbbk\{x, y, z, z'\}$$

$$\begin{array}{ccc} \| & \| & \| \\ C_2 & C_1 & C_0 \end{array}$$

and we can compute i-th **homology groups**:

$$H_i(X) \quad = \quad \ker \partial_{i-1} / \operatorname{im} \partial_i$$

The intuition is that the rank of $H_i(X)$ counts the number of holes in dimension i.

Theorem

Homology is invariant under homotopy equivalences (= continuous deformations of the space).

The classifying space

Given a convergent presentation

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

we can build a space with

0. one point $\bullet$

The classifying space

Given a convergent presentation

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

we can build a space with

0. one point $\bullet$
1. one segment $\bullet \xrightarrow{a} \bullet$ for each generator a

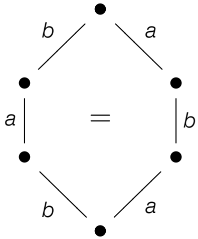
The classifying space

Given a convergent presentation

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

we can build a space with

0. one point $\bullet$
1. one segment $\bullet \overset{a}{\longrightarrow} \bullet$ for each generator a
2. one surface for each relation, e.g.



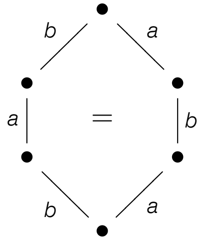
The classifying space

Given a convergent presentation

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

we can build a space with

0. one point $\bullet$
1. one segment $\bullet \overset{a}{\text{———}} \bullet$ for each generator a
2. one surface for each relation, e.g.



3. one volume for each critical pair

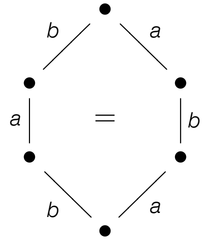
The classifying space

Given a convergent presentation

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

we can build a space with

0. one point •
1. one segment $\bullet \overset{a}{\text{———}} \bullet$ for each generator *a*
2. one surface for each relation, e.g.



3. one volume for each critical pair
4. one 4-volume for each critical triple

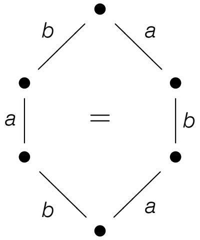
The classifying space

Given a convergent presentation

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

we can build a space with

0. one point $\bullet$
1. one segment $\bullet \xrightarrow{a} \bullet$ for each generator a
2. one surface for each relation, e.g.



3. one volume for each critical pair
4. one 4-volume for each critical triple
5. etc.

The classifying space

Theorem (Squier'87)

The homology of this space only depends on the presented monoid (not on the actual convergent presentation!).

Invariance under homotopy equivalence translates into this setting into invariance under (convergent) presentation!

The classifying space

Theorem (Squier'87)

The homology of this space only depends on the presented monoid (not on the actual convergent presentation!).

Invariance under homotopy equivalence translates into this setting into invariance under (convergent) presentation!

Remark

Actually, all these computations can be performed purely algebraically, without ever using topological spaces...

The counter-example

Example (Squier'87-Lafont-Prouté'91)

Consider the monoid M presented by

$$\langle a, b, c, d, d' \mid ab = a, da = ac, d'a = ac \rangle$$

1. has decidable word problem

The counter-example

Example (Squier'87-Lafont-Prouté'91)

Consider the monoid M presented by

$$\langle a, b, c, d, d' \mid ab = a, da = ac, d'a = ac \rangle$$

1. has decidable word problem
2. admits an infinite convergent presentation

The counter-example

Example (Squier'87-Lafont-Prouté'91)

Consider the monoid M presented by

$$\langle a, b, c, d, d' \mid ab = a, da = ac, d'a = ac \rangle$$

1. has decidable word problem
2. admits an infinite convergent presentation
3. from which we can compute that $H_3(M)$ is infinite

The counter-example

Example (Squier'87-Lafont-Prouté'91)

Consider the monoid M presented by

$$\langle a, b, c, d, d' \mid ab = a, da = ac, d'a = ac \rangle$$

1. has decidable word problem
2. admits an infinite convergent presentation
3. from which we can compute that $H_3(M)$ is infinite
4. $H_3(M)$ is a subquotient of $\Bbbk P$ where P are the critical pairs

The counter-example

Example (Squier'87-Lafont-Prouté'91)

Consider the monoid M presented by

$$\langle a, b, c, d, d' \mid ab = a, da = ac, d'a = ac \rangle$$

1. has decidable word problem
2. admits an infinite convergent presentation
3. from which we can compute that $H_3(M)$ is infinite
4. $H_3(M)$ is a subquotient of $\Bbbk P$ where P are the critical pairs
5. if there was a finite convergent presentation,
   it would have a finite number of critical pairs

The counter-example

Example (Squier'87-Lafont-Prouté'91)

Consider the monoid M presented by

$$\langle a, b, c, d, d' \mid ab = a, da = ac, d'a = ac \rangle$$

1. has decidable word problem
2. admits an infinite convergent presentation
3. from which we can compute that $H_3(M)$ is infinite
4. $H_3(M)$ is a subquotient of $\mathbb{k}P$ where P are the critical pairs
5. if there was a finite convergent presentation,
   it would have a finite number of critical pairs
6. there is no finite convergent presentation of the monoid!

Now, something *new*:
this can be extended to
term rewriting systems!

Algebraic theories

An **algebraic theory**

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

consists of

1. G: operations with given arities
2. R: equations between terms generated by operations

Example

- the theory of groups is given by $m : 2$, $e : 0$, $i : 1$ and

$$m(m(x_1, x_2), x_3) = m(x_1, m(x_2, x_3))$$

$$m(e, x_1) = x_1 \qquad m(x_1, e) = x_1$$

$$m(i(x_1), x_1) = e \qquad m(x_1, i(x_1)) = e$$

- rings, fields, etc.
- (semi)lattices, booleans algebras, etc.

Models

A **model** of an algebraic theory consists of

- a set X,
- an interpretation $[\![f]\!] : X^n \to X$ for each operation f of arity n,
- such that the axioms are satisfied.

Example

Models of the theory of groups are groups.

Equivalence between theories

Two theories are **equivalent** when they have the same models.

Example

Consider the theory of groups, given by $m : 2$, $e : 0$, $i : 1$ and

$$m(m(x_1, x_2), x_3) = m(x_1, m(x_2, x_3))$$
$$m(e, x_1) = x_1 \qquad m(x_1, e) = x_1$$
$$m(i(x_1), x_1) = e \qquad m(x_1, i(x_1)) = e$$

The equations in red are derivable from the other.

Equivalence between theories

Two theories are **equivalent** when they have the same models.

Example

Consider the theory of groups, given by $m : 2$, $e : 0$, $i : 1$ and

$$m(m(x_1, x_2), x_3) = m(x_1, m(x_2, x_3))$$

$$m(e, x_1) = x_1 \qquad\qquad m(x_1, e) = x_1$$

$$m(i(x_1), x_1) = e \qquad\qquad m(x_1, i(x_1)) = e$$

The equations in red are derivable from the other.

$$\begin{aligned} xe &= (ex)e = ((x^{--}x^{-})x)e = (x^{--}(x^{-}x))e = (x^{--}e)e \\ &= x^{--}(ee) = x^{--}e = x^{--}(x^{-}x) = (x^{--}x^{-})x = ex = x \end{aligned}$$

Equivalence between theories

Two theories are **equivalent** when they have the same models.

Example

Consider the theory of groups, given by $m : 2$, $e : 0$, $i : 1$ and

$$m(m(x_1, x_2), x_3) = m(x_1, m(x_2, x_3))$$
$$m(e, x_1) = x_1$$
$$m(i(x_1), x_1) = e$$

The equations in red are derivable from the other.

$$xe = (ex)e = ((x^{--}x^{-})x)e = (x^{--}(x^{-}x))e = (x^{--}e)e$$
$$= x^{--}(ee) = x^{--}e = x^{--}(x^{-}x) = (x^{--}x^{-})x = ex = x$$

`Can we find minimal (or small) axiomatizations for theories?`

One relation for (abelian) groups

In 1938, Tarski observed that the theory of abelian groups can be axiomatized with two operations $d : 2$, $a : 0$ and one relation

$$d(x_1, d(x_2, d(x_3, d(x_1, x_2)))) = x_3$$

where a ensure that the model is not empty.

A **one-based** theory is a theory which can be axiomatized with only one axiom.

The quest for one-based theories

There is an interesting line of efforts to find one-based theories:

- 1938: abelian groups is one-based
- 1952: groups is one-based
- 1965: semi-lattices is not one-based
- 1970: distributive lattices is not one-based
  lattices is one-based (300 000 sym. / 34 var.)
- 1973: boolean algebras is one-based ($\geq$ 40 000 000 symb.)
- 2002: boolean algebras is one-based (12 symb.)
- 2003: lattices is one-based (29 symb. / 8 var.)
- …

AXIOMS FOR SEMI-LATTICES

D. H. Potts

A <u>semi-lattice</u> (Birkhoff, Lattice Theory, p. 18, Ex. 1) is an algebra $\langle A, . \rangle$ with a single binary operation satisfying: (1) $x = xx$, (2) $xy = yx$, and (3) $(xy)z = x(yz)$. In this note we show that the three identities may be reduced to two but cannot be reduced to one.

It is easy to see that (2), (3) imply (4) $(uv)((wx)(yz)) = ((vu)(xw))(zy)$. Setting $w = y = u$ and $x = z = v$ in (4) and using (1) we get $uv = vu$. Setting $v = u$, $x = w$, and $z = y$ in (4) and using (1) we get $u(wy) = (uw)y$. And so (1) and (4) imply (2) and (3).

If a single identity is sufficient to define the notion of <u>semi-lattice</u> it must be of form $x = \ldots$. Any identity not of that form is satisfied by, e. g. the algebra $\langle \{0, 1\}, . \rangle$ where $00 = 01 = 10 = 11 = 0$, which is not a semi-lattice.

Now suppose we have a semi-lattice with two distinct elements a, b. Let $c = ab$. Either $c \neq a$ or $c \neq b$. We suppose the latter. Then $bb = b$ and $bc = cb = cc = c$. Thus any identity holding in a semi-lattice with at least two elements must have the same variables occurring on each side of the equality sign. For suppose "x" occurs on the left but not on the right. Setting $x = c$ and all other variables equal to b yields the contradiction $c = b$.

Thus a single sufficing identity would have to be of form $x = f(x)$. Clearly such an identity will not imply (2), for the algebra $\langle \{0, 1\}, . \rangle$ where $00 = 01 = 0$ and $10 = 11 = 1$ satisfies $x = f(x)$ for any f but is not commutative.

University of California, Berkeley

Axioms for semi-lattices

A **semi-lattice** is a set equipped with a multiplication such that

$$(xy)z = x(yz) \qquad xy = yx \qquad xx = x$$

1. any axiom should be of the form $x = t$ otherwise the non-semi-lattice

$\cdot$	0	1
0	0	0
1	0	0

   would be a model

Axioms for semi-lattices

A **semi-lattice** is a set equipped with a multiplication such that

$$(xy)z = x(yz) \qquad xy = yx \qquad xx = x$$

1. any axiom should be of the form $x = t$
2. any axiom $t = u$ should have $\mathrm{FV}(t) = \mathrm{FV}(u)$ otherwise the semi-lattice

$\cdot$	0	1
0	0	1
1	1	1

   would not be a model

Axioms for semi-lattices

A **semi-lattice** is a set equipped with a multiplication such that

$$(xy)z = x(yz) \qquad xy = yx \qquad xx = x$$

1. any axiom should be of the form $x = t$
2. any axiom $t = u$ should have $\mathrm{FV}(t) = \mathrm{FV}(u)$
3. the axiom cannot be of the form $x = t(x)$ otherwise the non-semi-lattice

$\cdot$	0	1
0	0	0
1	1	1

would be a model

Axioms for semi-lattices

A **semi-lattice** is a set equipped with a multiplication such that

$$(xy)z = x(yz) \qquad xy = yx \qquad xx = x$$

1. any axiom should be of the form $x = t$
2. any axiom $t = u$ should have $\mathrm{FV}(t) = \mathrm{FV}(u)$
3. the axiom cannot be of the form $x = t(x)$
4. we can also show that any other choice of generators suffers from the same problem!

Not one-based theories

We are interested in showing that theories are *not* one-based:

- existing proofs are tricky and specific to particular theories
- they rely on finding counter-examples using some models

Here, instead

- we provide a method which is entirely automatic
- but it does not provide an answer in every case

The general method

Algorithm (Malbos-Mimram'16)

1. start from a theory $\mathcal{T}$,

The general method

Algorithm (Malbos-Mimram'16)

1. start from a theory $\mathcal{T}$,
2. orient it so that you get a terminating and confluent term rewriting system,

Algorithm (Malbos-Mimram'16)

1. start from a theory $\mathcal{T}$,
2. orient it so that you get a terminating and confluent term rewriting system,
3. feed it to the computer and compute

$$H_2(\mathcal{T}) \qquad \in \qquad \mathbb{N}$$

The general method

Algorithm (Malbos-Mimram'16)

1. start from a theory $\mathcal{T}$,
2. orient it so that you get a terminating and confluent term rewriting system,
3. feed it to the computer and compute

$$H_2(\mathcal{T}) \qquad \in \qquad \mathbb{N}$$

4. we know that we need at least $H_2(\mathcal{T})$ relations.

The general method

Algorithm (Malbos-Mimram'16)

1. start from a theory $\mathcal{T}$,
2. orient it so that you get a terminating and confluent term rewriting system,
3. feed it to the computer and compute

$$H_2(\mathcal{T}) \qquad \in \qquad \mathbb{N}$$

4. we know that we need at least $H_2(\mathcal{T})$ relations.

Note that:

- the theory might not be orientable as a convergent rs,
- we might compute $H_2(\mathcal{T}) = 0$,
- we have examples where it works though :)

Thanks!