Monoids (and more) as bridges

Samuel Mimram

Presentations

Algebraic structures are presented by generators and relations.

Presentations

Algebraic structures are presented by generators and relations.

The presentation is usually not unique, which is a good thing:
by varying the presentation, we can reach ones with nice properties.

Presentations

Algebraic structures are presented by generators and relations.

The presentation is usually not unique, which is a good thing:
by varying the presentation, we can reach ones with nice properties.

Every element of the algebraic structure can be obtained as a composite of generators. A natural question is:

when do two composite represent the same element?

Presentations

Algebraic structures are presented by generators and relations.

The presentation is usually not unique, which is a good thing:
by varying the presentation, we can reach ones with nice properties.

Every element of the algebraic structure can be obtained as a composite of generators. A natural question is:

when do two composite represent the same element?

We will see this is answered effectively (= we can implement it) by

rewriting theory

Presentations

$$\mathbf{Sh}(\mathcal{C}, J) \simeq \mathbf{Sh}(\mathcal{D}, K)$$

$(\mathcal{C}, J)$

$(\mathcal{D}, K)$

Presentations

$$G^*/R \simeq H^*/S$$

$\langle G \mid R \rangle$ $\langle H \mid S \rangle$

Part I

Abstract rewriting systems

Abstract rewriting systems

An abstract rewriting system consists of a set G together with a rewriting relation $R \subseteq G \times G$.

$$a \rightleftarrows b \longrightarrow c \qquad\qquad d \longrightarrow e$$

Abstract rewriting systems

An abstract rewriting system consists of a set G together with a rewriting relation $R \subseteq G \times G$.

$$a \rightleftarrows b \longrightarrow c \qquad\qquad d \longrightarrow e$$

We write

$$[-] : G \to G/R$$

and think of $a, b \in G$ such that $[a] = [b]$ as two possible descriptions of the corresponding element of G/R.

Abstract rewriting systems

An abstract rewriting system consists of a set G together with a rewriting relation $R \subseteq G \times G$.

$$a \rightleftarrows b \longrightarrow c \qquad\qquad d \longrightarrow e$$

We write

$$[-] : G \to G/R$$

and think of $a, b \in G$ such that $[a] = [b]$ as two possible descriptions of the corresponding element of G/R.

We can decide whether two elements are in the same equivalence class when

- we have a canonical representative in each equivalence class,
- can compute the canonical representative of an element.

Normal forms

If we think that $a \longrightarrow b$ means b is "more canonical" than a then canonical representatives should be given by normal forms: elements which are not the source of any reduction.

$$a \rightleftarrows b \longrightarrow c \qquad\qquad d \longrightarrow e$$

An equivalence class can however have

Normal forms

If we think that $a \longrightarrow b$ means b is “more canonical” than a then canonical representatives should be given by normal forms: elements which are not the source of any reduction.

$$a \rightleftarrows b \longrightarrow c \qquad\qquad d \longrightarrow e$$

An equivalence class can however have

- more than one normal form:

$$a \longleftarrow b \longrightarrow c$$

Normal forms

If we think that $a \longrightarrow b$ means b is “more canonical” than a then canonical representatives should be given by normal forms: elements which are not the source of any reduction.

$$a \rightleftarrows b \longrightarrow c \qquad\qquad d \longrightarrow e$$

An equivalence class can however have

- more than one normal form:

$$a \longleftarrow b \longrightarrow c$$

- no normal form:

$$a \rightleftarrows b \qquad\qquad a_0 \longrightarrow a_1 \longrightarrow a_2 \longrightarrow \ldots$$

Termination

An ARS is terminating if there is no infinite path

$$a_0 \longrightarrow a_1 \longrightarrow a_2 \longrightarrow \dots$$

Termination

An ARS is terminating if there is no infinite path

$$a_0 \longrightarrow a_1 \longrightarrow a_2 \longrightarrow \dots$$

Proposition
In a terminating ARS, every equivalence class contains a normal form.

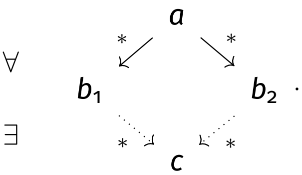
Proof.
Given a, consider a maximal path

$$a = a_0 \longrightarrow a_1 \longrightarrow a_2 \longrightarrow \dots \longrightarrow a_n$$

□

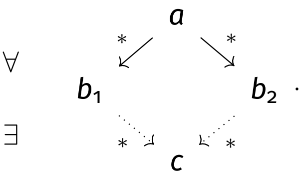
Confluence

An ARS is confluent when



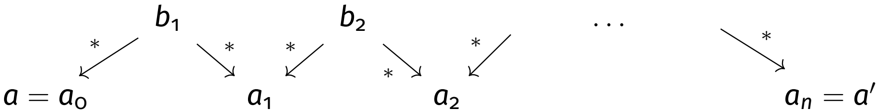
Confluence

An ARS is confluent when



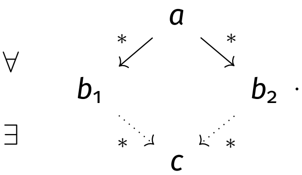
Proposition (Church-Rosser'36)
In a confluent ARS, two equivalent terms rewrite to a common element.

Proof.
Suppose $a \overset{*}{\leftrightarrow} a'$. This means that



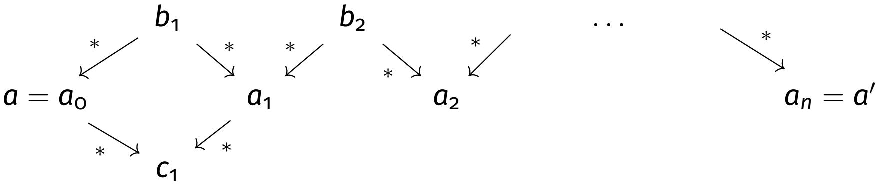
Confluence

An ARS is confluent when

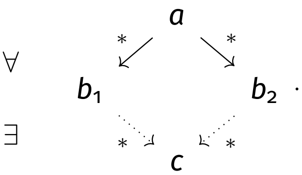


Proposition (Church-Rosser'36)
In a confluent ARS, two equivalent terms rewrite to a common element.

Proof.
Suppose $a \overset{*}{\leftrightarrow} a'$. This means that

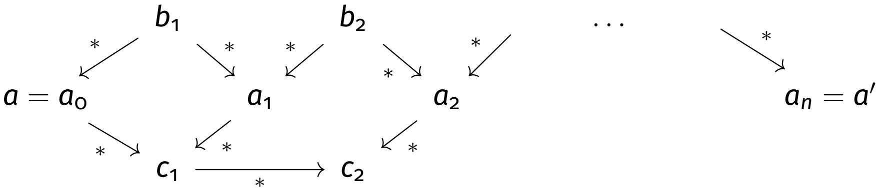


An ARS is confluent when



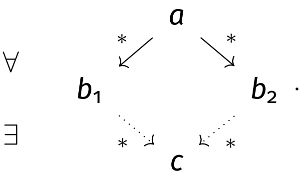
Proposition (Church-Rosser'36)
In a confluent ARS, two equivalent terms rewrite to a common element.

Proof.
Suppose $a \xleftrightarrow{*} a'$. This means that



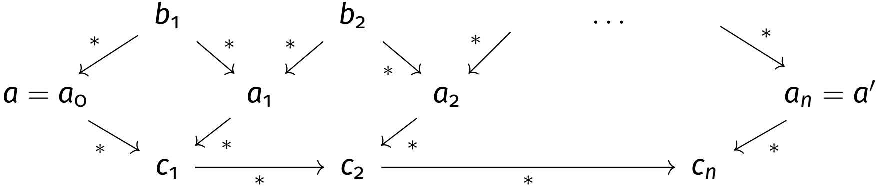
Confluence

An ARS is confluent when



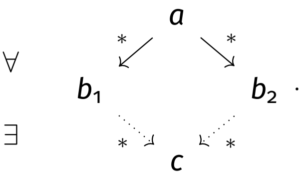
Proposition (Church-Rosser'36)
In a confluent ARS, two equivalent terms rewrite to a common element.

Proof.
Suppose $a \overset{*}{\leftrightarrow} a'$. This means that



Confluence

An ARS is confluent when



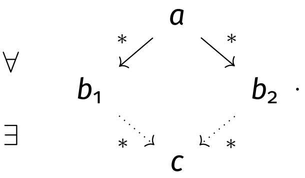
Proposition
In a confluent ARS, every equivalence class contains at most one normal form.

Proof.
Suppose that a and a' are equivalent normal forms.

$$a \overset{*}{\longleftrightarrow} a'$$

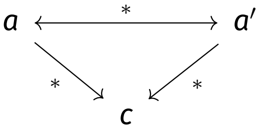
□

Confluence

An ARS is confluent when 

Proposition
In a confluent ARS, every equivalence class contains at most one normal form.

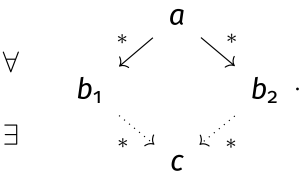
Proof.
Suppose that a and a' are equivalent normal forms. By previous theorem we have,



□

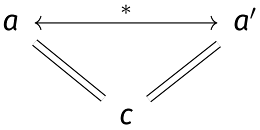
Confluence

An ARS is confluent when



Proposition
In a confluent ARS, every equivalence class contains at most one normal form.

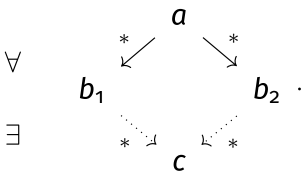
Proof.
Suppose that a and a' are equivalent normal forms. By previous theorem we have,



since a and a' are normal forms. □

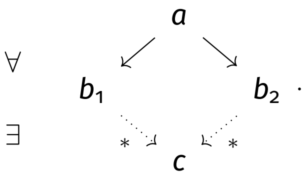
Local confluence

An ARS is confluent when



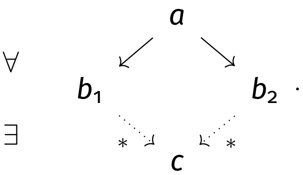
Local confluence

An ARS is locally confluent when



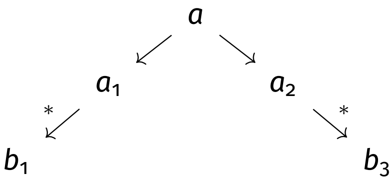
Local confluence

An ARS is locally confluent when



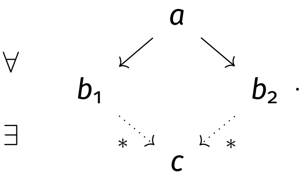
Proposition (Newman'42)
A terminating ARS is confluent if and only if it is locally confluent.

Proof.
By well-founded induction, locally confluent implies confluent:



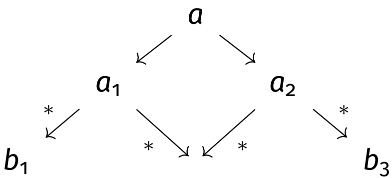
Local confluence

An ARS is locally confluent when

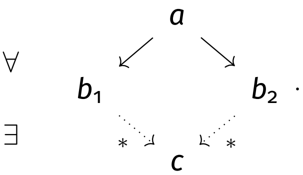


Proposition (Newman'42)
A terminating ARS is confluent if and only if it is locally confluent.

Proof.
By well-founded induction, locally confluent implies confluent:

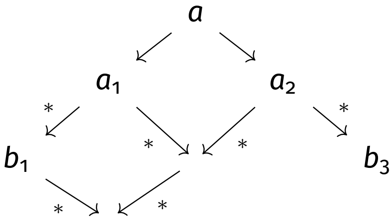


An ARS is locally confluent when

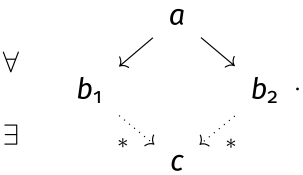


Proposition (Newman'42)
A terminating ARS is confluent if and only if it is locally confluent.

Proof.
By well-founded induction, locally confluent implies confluent:

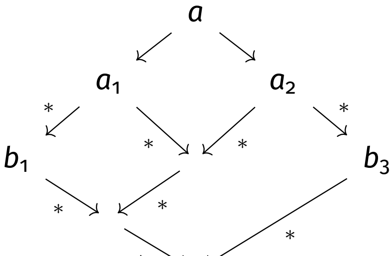


An ARS is locally confluent when



Proposition (Newman'42)
A terminating ARS is confluent if and only if it is locally confluent.

Proof.
By well-founded induction, locally confluent implies confluent:



Local confluence

An ARS is locally confluent when

$$\begin{array}{ccccc} \forall & & a & & \\ & \swarrow & & \searrow & \\ & b_1 & & b_2 & . \\ \exists & {}_{*}\searrow & & \swarrow_{*} & \\ & & c & & \end{array}$$

Remark (Huet'80)
Without termination, this does not hold

$$a \longleftarrow b \rightleftarrows c \longrightarrow d$$

Deciding equality

An ARS (G, R) is convergent when both terminating an (locally) confluent.

Deciding equality

An ARS (G, R) is convergent when both terminating an (locally) confluent.

By previous propositions,

Proposition
In a convergent ARS, every equivalence class contains a unique normal form.

Deciding equality

An ARS (G, R) is convergent when both terminating an (locally) confluent.

By previous propositions,

Proposition
In a convergent ARS, every equivalence class contains a unique normal form.

For $a, b \in G$, we can decide whether $[a] = [b]$ holds as follows:

$$a \overset{*}{\leftrightarrow} b$$

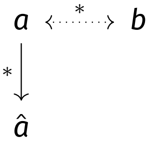
Deciding equality

An ARS (G, R) is convergent when both terminating an (locally) confluent.

By previous propositions,

Proposition
In a convergent ARS, every equivalence class contains a unique normal form.

For $a, b \in G$, we can decide whether $[a] = [b]$ holds as follows:



Deciding equality

An ARS (G, R) is convergent when both terminating an (locally) confluent.

By previous propositions,

Proposition
In a convergent ARS, every equivalence class contains a unique normal form.

For $a, b \in G$, we can decide whether $[a] = [b]$ holds as follows:

$$\begin{array}{ccc} a & \overset{*}{\leftrightarrow} & b \\ {\scriptstyle *}\downarrow & & \downarrow{\scriptstyle *} \\ \hat{a} & & \hat{b} \end{array}$$

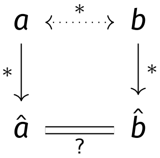
Deciding equality

An ARS (G, R) is convergent when both terminating an (locally) confluent.

By previous propositions,

Proposition
In a convergent ARS, every equivalence class contains a unique normal form.

For $a, b \in G$, we can decide whether $[a] = [b]$ holds as follows:



Part II

String rewriting systems

This is the core of rewriting theory.

Let's apply this to presentations of monoids.

Presentations of monoids

A monoid presentation / string rewriting system is a pair $\langle G \mid R \rangle$ consisting of

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

where G^* is the free monoid on G. It presents the monoid

$$G^*/R$$

where we quotient by the *congruence* generated by R.

Example

- $\mathbb{N} \simeq \langle a \mid \rangle$

Presentations of monoids

A monoid presentation / string rewriting system is a pair $\langle G \mid R \rangle$ consisting of

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

where G^* is the free monoid on G. It presents the monoid

$$G^*/R$$

where we quotient by the *congruence* generated by R.

Example

- $\mathbb{N} \simeq \langle a \mid \rangle$
- $\mathbb{N}/2\mathbb{N} \simeq \langle a \mid aa = 1 \rangle$

Presentations of monoids

A monoid presentation / string rewriting system is a pair $\langle G \mid R \rangle$ consisting of

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

where G^* is the free monoid on G. It presents the monoid

$$G^*/R$$

where we quotient by the *congruence* generated by R.

Example

- $\mathbb{N} \simeq \langle a \mid \rangle$
- $\mathbb{N}/2\mathbb{N} \simeq \langle a \mid aa = 1 \rangle$
- $\mathbb{N} \times \mathbb{N}/2\mathbb{N} \simeq \langle a, b \mid ba = ab, bb = 1 \rangle$

Presentations of monoids

A monoid presentation / string rewriting system is a pair $\langle G \mid R\rangle$ consisting of

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

where G^* is the free monoid on G. It presents the monoid

$$G^*/R$$

where we quotient by the *congruence* generated by R.

Example

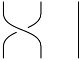
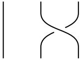
- $\mathbb{N} \simeq \langle a \mid \rangle$
- $\mathbb{N}/2\mathbb{N} \simeq \langle a \mid aa = 1\rangle$
- $\mathbb{N} \times \mathbb{N}/2\mathbb{N} \simeq \langle a, b \mid ba = ab, bb = 1\rangle$
- $S_n \simeq \langle a_0, \ldots, a_{n-1} \mid a_i a_i = 1, a_i a_{i+1} a_i = a_{i+1} a_i a_{i+1}, a_i a_j = a_j a_i\rangle$

Presenting of S_3

A presentation of S_3 is

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

where

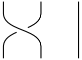
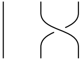
$a =$  $b =$ 

Presenting of S_3

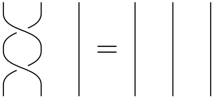
A presentation of S_3 is

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

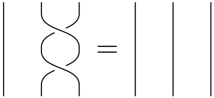
where

$a =$  $b =$ 

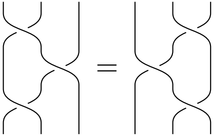
and the relations are



$aa = 1$



$bb = 1$



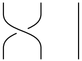
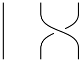
$aba = bab$

Presenting of S_3

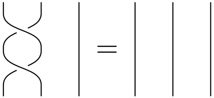
A presentation of S_3 is

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

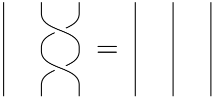
where

$a =$  $b =$ 

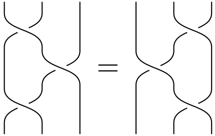
and the relations are



$aa = 1$



$bb = 1$



$aba = bab$

Note: it is clear that the relations are valid, but not that they are complete…
(rewriting can help here)

String rewriting systems

Given a presentation $\langle G \mid R\rangle$, a rewriting step is

$$uvw \to uv'w$$

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

For instance with $S_3 \simeq \langle a, b \mid aa \to 1, aba \to bab, bb \to 1\rangle$, we have

$$bbabaab \to bbbabab$$

String rewriting systems

Given a presentation $\langle G \mid R \rangle$, a rewriting step is

$$uvw \to uv'w$$

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

For instance with $S_3 \simeq \langle a, b \mid aa \to 1, aba \to bab, bb \to 1 \rangle$, we have

$$bbabaab \to bbbabab$$

The presentation induces an ARS with

- elements of G^* as vertices,
- rewriting steps as edges,

thus allowing to re-use previous notions.

String rewriting systems

Given a presentation $\langle G \mid R \rangle$, a rewriting step is

$$uvw \to uv'w$$

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

For instance with $S_3 \simeq \langle a, b \mid aa \to 1, aba \to bab, bb \to 1 \rangle$, we have

$$bbabaab \to bbbabab$$

Note that the following are equivalent:

- $u \overset{*}{\leftrightarrow} v$
- u and v are related by the congruence generated by R
- $[u] = [v]$ in G^*/R.

Deciding equality in presentations

Given a presentation $\langle G \mid R \rangle$ and words $u, v \in G^*$, we want to decide equality, i.e. answer

$$\text{do we have } [u] = [v]?$$

Deciding equality in presentations

Given a presentation $\langle G \mid R \rangle$ and words $u, v \in G^*$, we want to decide equality, i.e. answer

$$\text{do we have } [u] = [v]?$$

Proposition
When the presentation is terminating and (locally) confluent, $[u] = [v]$ holds if and only if u and v have the same normal form:

$$\begin{array}{ccc} u & & v \\ {\scriptstyle *}\big\downarrow & & \big\downarrow{\scriptstyle *} \\ \hat{u} & \overset{?}{=\!=\!=} & \hat{v} \end{array}$$

Given a presentation $\langle G \mid R \rangle$, how do we show

- that it is terminating?
- that it is confluent?

Showing termination

Termination can usually be shown by showing that rules (and thus rewriting steps) make something decrease in a well-founded order.

For instance, with

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

we have that

- the rules make the length of words decrease (strictly for the first two),
- the third rule make the number of a's strictly decrease.

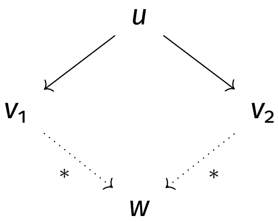
They are thus strictly decreasing under

$$>_{\text{len}} \times_{\text{lex}} >_{a}$$

which is well-founded.

Showing (local) confluence

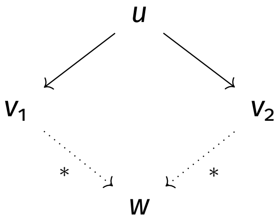
Since we are interested in terminating rewriting systems, it is enough to show that a presentation is locally confluent:



i.e. that every branching can be closed.

Showing (local) confluence

Since we are interested in terminating rewriting systems, it is enough to show that a presentation is locally confluent:



i.e. that every branching can be closed.

Problem: we have to check for all possible triples (v_1, u, v_2)...

We should remove "obviously commuting" diagrams from our search.

Independent branchings

Suppose that we have a branching rewriting two “independent” parts:

$$aaababbababb$$

$$aababbbababb \qquad \qquad aaababbbabbb$$

Independent branchings

Suppose that we have a branching rewriting two “independent” parts:

$$aaababbababb$$

$$aababbbababb \qquad aaababbbabbb$$

$$aababbbbabbb$$

It can always be closed!

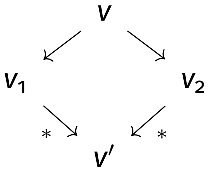
We can thus restrict to situation where the changed parts are overlapping.

Critical branchings

Suppose that we have a branching

$$
\begin{array}{ccc}
 & v & \\
\swarrow & & \searrow \\
v_1 & & v_2
\end{array}
$$

Suppose that we have a branching which can be closed



Critical branchings

Suppose that we have a branching which can be closed

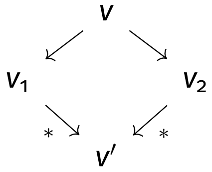
$$\begin{array}{ccccc} & & v & & \\ & \swarrow & & \searrow & \\ v_1 & & & & v_2 \\ & {}^{*}\searrow & & \swarrow{}^{*} & \\ & & v' & & \end{array}$$

then the branching

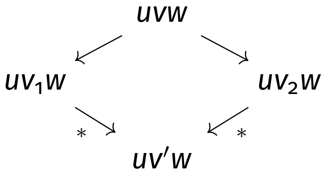
$$\begin{array}{ccccc} & & uvw & & \\ & \swarrow & & \searrow & \\ uv_1w & & & & uv_2w \end{array}$$

We can thus restrict to situations where the context is minimal.

Suppose that we have a branching which can be closed



then the branching



can also be closed.

We can thus restrict to situations where the context is minimal.

Critical branchings

A critical branching is a situation

$$\begin{array}{ccc} & u & \\ \swarrow & & \searrow \\ v_1 & & v_2 \end{array}$$

which not “independent” and with minimal context.

Critical branchings

A critical branching is a situation

$$\begin{array}{ccc} & u & \\ \swarrow & & \searrow \\ v_1 & & v_2 \end{array}$$

which not “independent” and with minimal context.

For instance, the critical branchings generated by

$$aa \to 1 \qquad \text{and} \qquad aba \to bab$$

are

$$ba \longleftarrow aaba \longrightarrow abab \qquad\qquad baba \longleftarrow abaa \longrightarrow ab$$

Critical branchings

A critical branching is a situation

$$\begin{array}{ccc} & u & \\ \swarrow & & \searrow \\ v_1 & & v_2 \end{array}$$

which not "independent" and with minimal context.

For instance, the critical branchings generated by

$$aa \to 1 \qquad \text{and} \qquad aba \to bab$$

are

$$ba \longleftarrow aaba \longrightarrow abab \qquad\qquad baba \longleftarrow abaa \longrightarrow ab$$

but not

$$baba \longleftarrow aababa \longrightarrow aabbab \qquad\qquad ababab \longleftarrow aabaab \longrightarrow aabb$$

Proposition
A rewriting system with a finite number of rules has a finite number of critical branchings (and we can compute them).

Proof.
Try to make the left side of all pairs of rules overlap in a non-trivial way. □

Proposition
A rewriting system with a finite number of rules has a finite number of critical branchings (and we can compute them).

Proof.
Try to make the left side of all pairs of rules overlap in a non-trivial way. □

Proposition
If all the critical branchings are confluent then the system is locally confluent.

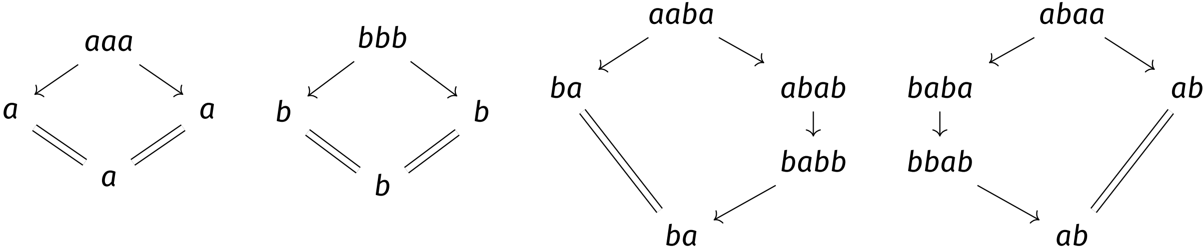
Proof.
By definition of critical branchings. □

Example

With our favorite presentation

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

we can check that the critical branchings are confluent:



Example

With our favorite presentation

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

we can thus test equality by comparing normal forms:

- $aabaabb$
- $ababa$
- $bbab$

Example

With our favorite presentation

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

we can thus test equality by comparing normal forms:

- $aabaabb \to baabb \to bbb \to b$
- $ababa$
- $bbab$

Example

With our favorite presentation

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

we can thus test equality by comparing normal forms:

- $aabaabb \to baabb \to bbb \to b$
- $ababa \to babba \to baa \to b$
- $bbab$

Example

With our favorite presentation

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

we can thus test equality by comparing normal forms:

- $aabaabb \to baabb \to bbb \to b$
- $ababa \to babba \to baa \to b$
- $bbab \to ab$

Non-confluent presentations

Of course, presentations are not always confluent:

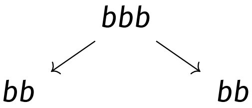
$$\langle a, b \mid bb \to b, aa \to bb \qquad \rangle$$

Non-confluent presentations

Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb \qquad\qquad \rangle$$

has critical branchings

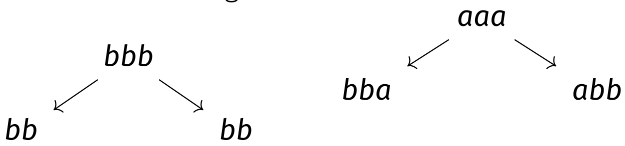


Non-confluent presentations

Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb \qquad\qquad \rangle$$

has critical branchings

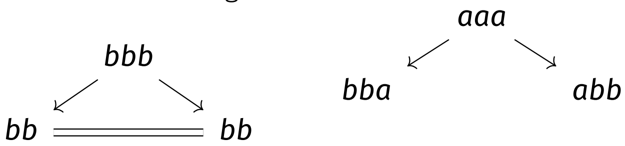


Non-confluent presentations

Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb \qquad\qquad \rangle$$

has critical branchings

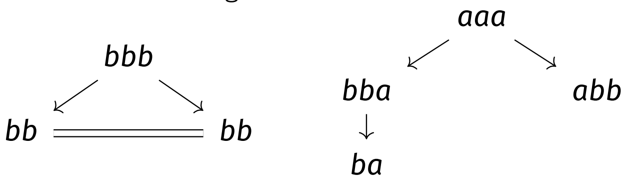


Non-confluent presentations

Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb \qquad\qquad \rangle$$

has critical branchings

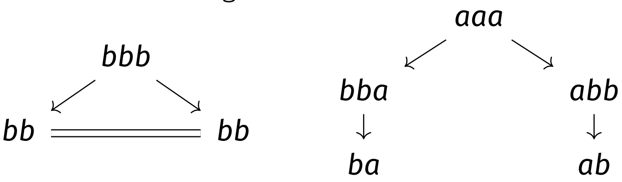


Non-confluent presentations

Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb \rangle$$

has critical branchings

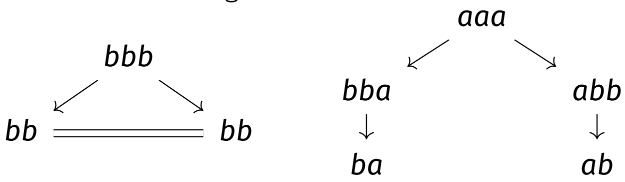


Non-confluent presentations

Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb \rangle$$

has critical branchings



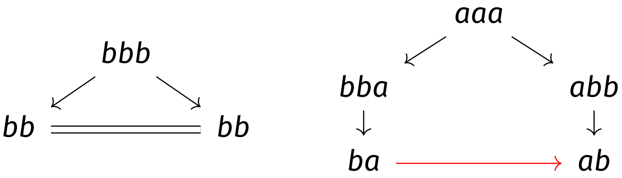
What do we do from there?

Non-confluent presentations

Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb, ba \to ab \rangle$$

has critical branchings



What do we do from there?

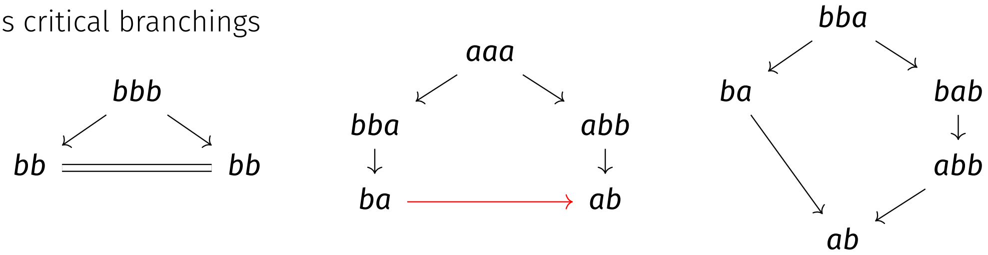
Knuth-Bendix completion procedure
Iteratively compute critical branchings and add new rules between normal forms when they are not confluent (the orientation is chosen according to a fixed order)

Non-confluent presentations

Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb, ba \to ab \rangle$$

has critical branchings



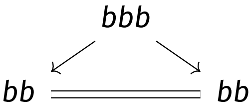
(and another from *baa*)

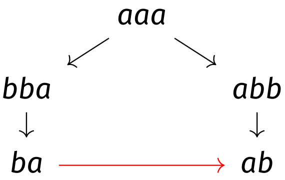
Knuth-Bendix completion procedure
Iteratively compute critical branchings and add new rules between normal forms when they are not confluent (the orientation is chosen according to a fixed order)

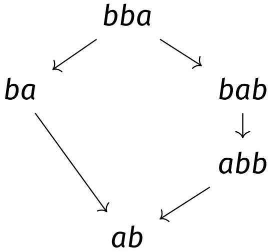
Of course, presentations are not always confluent:

$$\langle a, b \mid bb \to b, aa \to bb, {\color{red} ba \to ab} \rangle$$

has critical branchings







(and another from *baa*)

Knuth-Bendix completion procedure
Iteratively compute critical branchings and add new rules between normal forms when they are not confluent (the orientation is chosen according to a fixed order) [+ simplify rules using newly added ones].

Knuth-Bendix completion procedure

Note that the Knuth-Bendix is not guaranteed to end after a finite amount of time (this is not an *algorithm*).

For instance,

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

might get completed to

$$\langle a, b, c, d \mid ac^n b \to ac^n, da \to ac \rangle$$

The inductive limit is always locally confluent.

Note that the Knuth-Bendix is not guaranteed to end after a finite amount of time (this is not an *algorithm*).

For instance,

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

might get completed to

$$\langle a, b, c, d \mid ac^n b \to ac^n, da \to ac \rangle$$

Remark
The presentation

$$\langle a, b, c, d \mid ab \to a, da \leftarrow ac \rangle$$

has not critical branching and is thus locally confluent!

Tietze equivalence

Two presentations $\langle G \mid R \rangle$ and $\langle G' \mid R' \rangle$ are equivalent when they present the same monoid:

$$G/R \simeq G'/R'$$

Can we come up with some elementary characterization of this equivalence?

Tietze transformations

Given a presentation $\langle G \mid R \rangle$ the Tietze transformations are

1. add a definable generator:

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

   for some $a \notin G$ and $u \in G^*$,

Tietze transformations

Given a presentation $\langle G \mid R \rangle$ the Tietze transformations are

1. add a definable generator:

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

   for some $a \notin G$ and $u \in G^*$,

2. add a definable relation:

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

   for $u, v \in G^*$ which are related by the congruence generated by R.

Tietze transformations

Given a presentation $\langle G \mid R \rangle$ the Tietze transformations are

1. add a definable generator:

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

   for some $a \notin G$ and $u \in G^*$,

2. add a definable relation:

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

   for $u, v \in G^*$ which are related by the congruence generated by R.

Lemma
Tietze transformations preserve the presented monoid.

Tietze transformations

Given a presentation $\langle G \mid R \rangle$ the Tietze transformations are

1. add a definable generator:

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

   for some $a \notin G$ and $u \in G^*$,

2. add a definable relation:

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

   for $u, v \in G^*$ which are related by the congruence generated by R.

Lemma
Tietze transformations preserve the presented monoid.

Note that KB algorithm only uses transformations of the second kind.

Tietze transformations

Proposition
Two finite presentations $\langle G \mid R\rangle$ and $\langle G' \mid R'\rangle$ present the same monoid if and only if they are related by a finite series of Tietze transformations:

$$\langle G \mid R\rangle = \langle G_0 \mid R_0\rangle \leftrightsquigarrow \langle G_1 \mid R_1\rangle \leftrightsquigarrow \ldots \leftrightsquigarrow \langle G_n \mid R_n\rangle = \langle G' \mid R'\rangle$$

Tietze transformations

Proposition
Two finite presentations $\langle G \mid R \rangle$ and $\langle G' \mid R' \rangle$ present the same monoid if and only if they are related by a finite series of Tietze transformations:

$$\langle G \mid R \rangle \overset{*}{\rightsquigarrow} \langle G'' \mid R'' \rangle \overset{*}{\leftsquigarrow} \langle G' \mid R' \rangle$$

Proposition
Two finite presentations $\langle G \mid R\rangle$ and $\langle G' \mid R'\rangle$ present the same monoid if and only if they are related by a finite series of Tietze transformations:

$$\langle G \mid R\rangle \overset{*}{\rightsquigarrow} \langle G'' \mid R''\rangle \overset{*}{\leftsquigarrow} \langle G' \mid R'\rangle$$

Proof.
Writing $G = \{a_1, \ldots, a_n\}$ and $G' = \{b_1, \ldots, b_m\}$, we define

$$\langle G'' \mid R''\rangle = \langle G, G' \mid R, R', a_i = u'_i, b_i = u_i\rangle$$

where and $u'_i \in G'^*$ such that $[u'_i] = a_i$ and $v'_i \in G^*$ such that $[u_i] = b_j$. □

Proposition
Two finite presentations $\langle G \mid R\rangle$ and $\langle G' \mid R'\rangle$ present the same monoid if and only if they are related by a finite series of Tietze transformations:

$$\langle G \mid R\rangle \overset{*}{\rightsquigarrow} \langle G'' \mid R''\rangle \overset{*}{\leftsquigarrow} \langle G' \mid R'\rangle$$

Proof.
Writing $G = \{a_1, \ldots, a_n\}$ and $G' = \{b_1, \ldots, b_m\}$, we define

$$\langle G'' \mid R''\rangle = \langle G, G' \mid R, R', a_i = u_i', b_i = u_i\rangle$$

where and $u_i' \in G'^*$ such that $[u_i'] = a_i$ and $v_i' \in G^*$ such that $[u_i] = b_j$. □

This can be suitably generalized to infinite presentations.

Universality of rewriting

We have seen that for a monoid with a finite terminating and confluent presentation we can decide equality.

Universality of rewriting

We have seen that for a monoid with a finite terminating and confluent presentation we can decide equality.

Conversely, we wonder

is rewriting is universal?

which means

given a finitely presented monoid with decidable equality,
does it always admit a finite convergent presentation?

Universality of rewriting

The braid monoid admits the presentation

$$B_3^+ = \langle a, b \mid aba = bab \rangle$$

Universality of rewriting

The braid monoid admits the presentation

$$B_3^+ = \langle a, b \mid aba = bab \rangle$$

Lemma
The monoid has decidable equality.

Proof.
Since the only relation preserves length, equivalence classes are finite. □

Universality of rewriting

The braid monoid admits the presentation

$$B_3^+ = \langle a, b \mid aba = bab \rangle$$

Lemma
The monoid has decidable equality.

Proof.
Since the only relation preserves length, equivalence classes are finite. □

Proposition (Kapur-Narendran'85)
There is no convergent presentation of B_3^+ on the same generators.

The braid monoid admits the presentation

$$B_3^+ = \langle a, b \mid aba = bab \rangle$$

Lemma
The monoid has decidable equality.

Proof.
Since the only relation preserves length, equivalence classes are finite. $\square$

Proposition (Kapur-Narendran'85)
There is no convergent presentation of B_3^+ on the same generators.

This does not entirely solve the question since we are using only the second type of Tietze transformation (but it does for KB algorithm).

Universality of rewriting

However, we did not exploit Tietze transformations of first kind:

- $B_3^+ = \langle a, b \mid aba \to bab \rangle$

Universality of rewriting

However, we did not exploit Tietze transformations of first kind:

- $B_3^+ = \langle a, b \mid aba \to bab \rangle$
- $B_3^+ = \langle a, b, c \mid aba \to bab, ba \to c \rangle$

However, we did not exploit Tietze transformations of first kind:

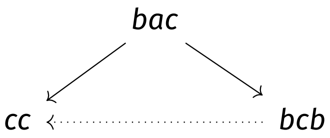
- $B_3^+ = \langle a, b \mid aba \to bab \rangle$
- $B_3^+ = \langle a, b, c \mid aba \to bab, ba \to c \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c \rangle$

$$\begin{array}{ccc} aba & \longrightarrow & bab \\ \downarrow & & \downarrow \\ ac & \dashrightarrow & cb \end{array}$$

Universality of rewriting

However, we did not exploit Tietze transformations of first kind:

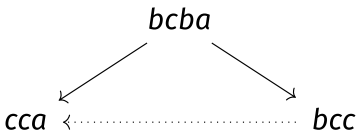
- $B_3^+ = \langle a, b \mid aba \to bab \rangle$
- $B_3^+ = \langle a, b, c \mid aba \to bab, ba \to c \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c, bcb \to cc \rangle$



Universality of rewriting

However, we did not exploit Tietze transformations of first kind:

- $B_3^+ = \langle a, b \mid aba \to bab \rangle$
- $B_3^+ = \langle a, b, c \mid aba \to bab, ba \to c \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c, bcb \to cc \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c, bcb \to cc, bcc \to cca \rangle$



Universality of rewriting

However, we did not exploit Tietze transformations of first kind:

- $B_3^+ = \langle a, b \mid aba \to bab \rangle$
- $B_3^+ = \langle a, b, c \mid aba \to bab, ba \to c \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c, bcb \to cc \rangle$
- $B_3^+ = \langle a, b, c \mid ac \to cb, ba \to c, bcb \to cc, bcc \to cca \rangle$

This is a convergent presentation!

Universality of rewriting

We are going to show that there is a monoid which admits no finite convergent presentation.

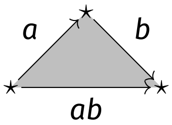
The strategy is that we are going to compute from a presentation of the monoid, a property which depends only on the monoid.

Moreover this property will be such that not finite convergent presentation can lead to it.

Monoids as geometric objects

The intuition is that a monoid can be considered as some form of geometric object with

- one point $\star$,
- the elements a of the monoid as 1-cells $a : \star \to \star$,
- equalities between products of elements of the monoid as 2-cells $\alpha : u \Rightarrow v$,



- trivial higher-dimensional information.

Monoids as geometric objects

With this point of view, it is natural to define the homology of a monoid M as follows.

1. Construct a resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$ by projective $\mathbb{Z}M$ modules:

$$\cdots \xrightarrow{d_4} C_3 \xrightarrow{d_3} C_2 \xrightarrow{d_2} C_1 \xrightarrow{d_1} C_0 \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

Monoids as geometric objects

With this point of view, it is natural to define the homology of a monoid M as follows.

1. Construct a resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$ by projective $\mathbb{Z}M$ modules:

$$\cdots \xrightarrow{d_4} C_3 \xrightarrow{d_3} C_2 \xrightarrow{d_2} C_1 \xrightarrow{d_1} C_0 \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

2. Tensor it by $\mathbb{Z}$ over $\mathbb{Z}M$:

$$\cdots \xrightarrow{\partial_4} C_3 \otimes \mathbb{Z} \xrightarrow{\partial_3} C_2 \otimes \mathbb{Z} \xrightarrow{\partial_2} C_1 \otimes \mathbb{Z} \xrightarrow{\partial_1} C_0 \otimes \mathbb{Z}$$

Monoids as geometric objects

With this point of view, it is natural to define the homology of a monoid M as follows.

1. Construct a resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$ by projective $\mathbb{Z}M$ modules:

$$\cdots \xrightarrow{d_4} C_3 \xrightarrow{d_3} C_2 \xrightarrow{d_2} C_1 \xrightarrow{d_1} C_0 \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

2. Tensor it by $\mathbb{Z}$ over $\mathbb{Z}M$:

$$\cdots \xrightarrow{\partial_4} C_3 \otimes \mathbb{Z} \xrightarrow{\partial_3} C_2 \otimes \mathbb{Z} \xrightarrow{\partial_2} C_1 \otimes \mathbb{Z} \xrightarrow{\partial_1} C_0 \otimes \mathbb{Z}$$

3. Compute

$$H_i(M) = \ker \partial_i / \operatorname{im} \partial_{i+1}$$

Monoids as geometric objects

With this point of view, it is natural to define the homology of a monoid M as follows.

1. Construct a resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$ by projective $\mathbb{Z}M$ modules:

$$\cdots \xrightarrow{d_4} C_3 \xrightarrow{d_3} C_2 \xrightarrow{d_2} C_1 \xrightarrow{d_1} C_0 \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

2. Tensor it by $\mathbb{Z}$ over $\mathbb{Z}M$:

$$\cdots \xrightarrow{\partial_4} C_3 \otimes \mathbb{Z} \xrightarrow{\partial_3} C_2 \otimes \mathbb{Z} \xrightarrow{\partial_2} C_1 \otimes \mathbb{Z} \xrightarrow{\partial_1} C_0 \otimes \mathbb{Z}$$

3. Compute

$$H_i(M) = \ker \partial_i / \operatorname{im} \partial_{i+1}$$

Lemma
Between any two projective resolutions there is a morphism, which is unique up to homotopy.

With this point of view, it is natural to define the homology of a monoid M as follows.

1. Construct a resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$ by projective $\mathbb{Z}M$ modules:

$$\cdots \xrightarrow{d_4} C_3 \xrightarrow{d_3} C_2 \xrightarrow{d_2} C_1 \xrightarrow{d_1} C_0 \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

2. Tensor it by $\mathbb{Z}$ over $\mathbb{Z}M$:

$$\cdots \xrightarrow{\partial_4} C_3 \otimes \mathbb{Z} \xrightarrow{\partial_3} C_2 \otimes \mathbb{Z} \xrightarrow{\partial_2} C_1 \otimes \mathbb{Z} \xrightarrow{\partial_1} C_0 \otimes \mathbb{Z}$$

3. Compute

$$H_i(M) = \ker \partial_i / \operatorname{im} \partial_{i+1}$$

Lemma
Between any two projective resolutions there is a morphism, which is unique up to homotopy. The homology thus does not depend on the choice of the resolution.

Constructing a tractable resolution

We can always construct a projective resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$:

$$\mathbb{Z} \longrightarrow 0$$

Constructing a tractable resolution

We can always construct a projective resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$:

$$\mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

Constructing a tractable resolution

We can always construct a projective resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$:

$$\mathbb{Z}M[\ker \varepsilon] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

Constructing a tractable resolution

We can always construct a projective resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$:

$$\mathbb{Z}M[\ker d_1] \xrightarrow{d_2} \mathbb{Z}M[\ker \varepsilon] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

Constructing a tractable resolution

We can always construct a projective resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$:

$$\cdots \xrightarrow{d_3} \mathbb{Z}M[\ker d_1] \xrightarrow{d_2} \mathbb{Z}M[\ker \varepsilon] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

Constructing a tractable resolution

We can always construct a projective resolution of the trivial $\mathbb{Z}M$-module $\mathbb{Z}$:

$$\cdots \xrightarrow{d_3} \mathbb{Z}M[\ker d_1] \xrightarrow{d_2} \mathbb{Z}M[\ker \varepsilon] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

However we cannot compute much from this, we need a smaller resolution!

Squier's theorem

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z} \longrightarrow 0$$

where

Squier's theorem

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

where

Squier's theorem

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z}M[G] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

where

Squier's theorem

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z}M[R] \xrightarrow{d_2} \mathbb{Z}M[G] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

where

Squier's theorem

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z}M[P] \xrightarrow{d_3} \mathbb{Z}M[R] \xrightarrow{d_2} \mathbb{Z}M[G] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

where

- P *is the set of critical branchings,*

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z}M[T] \xrightarrow{d_4} \mathbb{Z}M[P] \xrightarrow{d_3} \mathbb{Z}M[R] \xrightarrow{d_2} \mathbb{Z}M[G] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

where

- P *is the set of critical branchings,*
- T *is the set of critical triples.*

Squier's theorem

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z}M[T] \xrightarrow{d_4} \mathbb{Z}M[P] \xrightarrow{d_3} \mathbb{Z}M[R] \xrightarrow{d_2} \mathbb{Z}M[G] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

Proof.
Construct a contracting homotopy

$$\mathbb{Z}M[T] \underset{s_3}{\overset{d_4}{\rightleftarrows}} \mathbb{Z}M[P] \underset{s_2}{\overset{d_3}{\rightleftarrows}} \mathbb{Z}M[R] \underset{s_1}{\overset{d_2}{\rightleftarrows}} \mathbb{Z}M[G] \underset{s_0}{\overset{d_1}{\rightleftarrows}} \mathbb{Z}M \underset{\eta}{\overset{\varepsilon}{\rightleftarrows}} \mathbb{Z}$$

where the s_i are $\mathbb{Z}$-linear maps such that $\partial_{i+1}s_i + s_{i-1}\partial_i = 0$. □

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z}M[T] \xrightarrow{d_4} \mathbb{Z}M[P] \xrightarrow{d_3} \mathbb{Z}M[R] \xrightarrow{d_2} \mathbb{Z}M[G] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

Corollary
$H_3(M) = \ker \partial_3 / \operatorname{im} \partial_4$ *is finitely generated.*

Suppose given a monoid M with a finite convergent presentation $\langle G \mid R \rangle$.

Theorem (Squier'87)
One can construct a (partial) resolution

$$\mathbb{Z}M[T] \xrightarrow{d_4} \mathbb{Z}M[P] \xrightarrow{d_3} \mathbb{Z}M[R] \xrightarrow{d_2} \mathbb{Z}M[G] \xrightarrow{d_1} \mathbb{Z}M \xrightarrow{\varepsilon} \mathbb{Z} \longrightarrow 0$$

Corollary
$H_3(M) = \ker \partial_3 / \operatorname{im} \partial_4$ *is finitely generated.*

Corollary
If M is such that $H_3(M)$ is not finitely generated then M admits no finite convergent presentation.

A counter-example [Squier'87,Lafont-Prouté'91]

Consider the monoid M presented by

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

A counter-example [Squier'87,Lafont-Prouté'91]

Consider the monoid M presented by

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

by Knuth-Bendix, it can be completed into the infinite convergent presentation

$$\langle a, b, c, d, d' \mid A_n : ac^n b \to ac^n, B : da \to ac, B' : d'a \to ac \rangle$$

A counter-example [Squier'87,Lafont-Prouté'91]

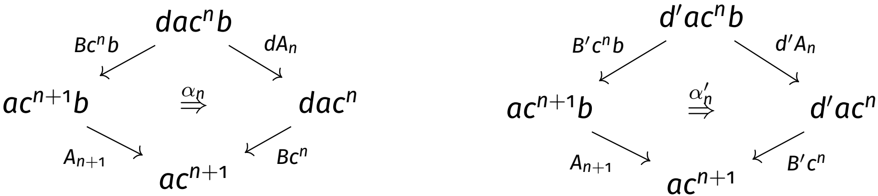
Consider the monoid M presented by

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

by Knuth-Bendix, it can be completed into the infinite convergent presentation

$$\langle a, b, c, d, d' \mid A_n : ac^n b \to ac^n, B : da \to ac, B' : d'a \to ac \rangle$$

there are two families of critical branchings



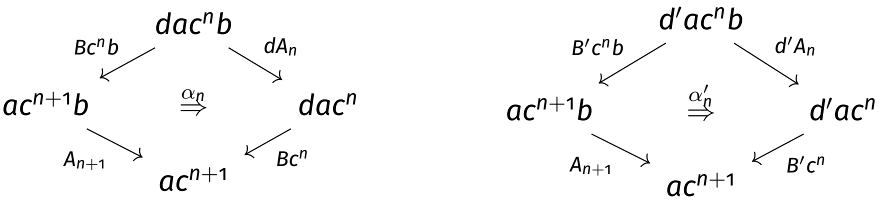
Consider the monoid M presented by

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

by Knuth-Bendix, it can be completed into the infinite convergent presentation

$$\langle a, b, c, d, d' \mid A_n : ac^n b \to ac^n, B : da \to ac, B' : d'a \to ac \rangle$$

there are two families of critical branchings



and no critical triple.

A counter-example [Squier'87,Lafont-Prouté'91]

The homology of M is thus the homology of

$$0 \xrightarrow{\partial_4} \mathbb{Z}[\alpha_n, \alpha'_n] \xrightarrow{\partial_3} \mathbb{Z}[A_n, B, B'] \xrightarrow{\partial_2} \mathbb{Z}[a, b, c, d, d'] \xrightarrow{\partial_1} \mathbb{Z}$$

A counter-example [Squier'87,Lafont-Prouté'91]

The homology of M is thus the homology of

$$0 \xrightarrow{\partial_4} \mathbb{Z}[\alpha_n, \alpha'_n] \xrightarrow{\partial_3} \mathbb{Z}[A_n, B, B'] \xrightarrow{\partial_2} \mathbb{Z}[a, b, c, d, d'] \xrightarrow{\partial_1} \mathbb{Z}$$

with, since $A_n : ac^n b \to ac^n$,

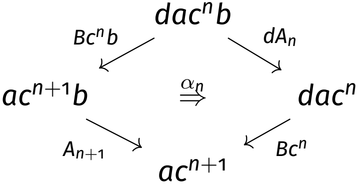
$$\partial_2(A_n) = [a] + n[c] - ([a] + n[c] + [b]) = [b]$$

A counter-example [Squier'87,Lafont-Prouté'91]

The homology of M is thus the homology of

$$0 \xrightarrow{\partial_4} \mathbb{Z}[\alpha_n, \alpha'_n] \xrightarrow{\partial_3} \mathbb{Z}[A_n, B, B'] \xrightarrow{\partial_2} \mathbb{Z}[a, b, c, d, d'] \xrightarrow{\partial_1} \mathbb{Z}$$

with, since



we have

$$\partial_3(\alpha_n) = [A_n] + [B] - ([B] - [A_{n+1}]) = [A_n] - [A_{n+1}]$$

and similarly

$$\partial(\alpha'_n) = [A_n] - [A_{n+1}]$$

A counter-example [Squier'87,Lafont-Prouté'91]

The homology of M is thus the homology of

$$0 \xrightarrow{\partial_4} \mathbb{Z}[\alpha_n, \alpha'_n] \xrightarrow{\partial_3} \mathbb{Z}[A_n, B, B'] \xrightarrow{\partial_2} \mathbb{Z}[a, b, c, d, d'] \xrightarrow{\partial_1} \mathbb{Z}$$

where $\ker \partial_3$ is infinitely generated by

$$[\alpha'_n] - [\alpha_n]$$

and thus

$$H_3(M)$$

is not finitely generated!

A counter-example [Squier'87,Lafont-Prouté'91]

The homology of M is thus the homology of

$$0 \xrightarrow{\partial_4} \mathbb{Z}[\alpha_n, \alpha'_n] \xrightarrow{\partial_3} \mathbb{Z}[A_n, B, B'] \xrightarrow{\partial_2} \mathbb{Z}[a, b, c, d, d'] \xrightarrow{\partial_1} \mathbb{Z}$$

where $\ker \partial_3$ is infinitely generated by

$$[\alpha'_n] - [\alpha_n]$$

and thus

$$H_3(M)$$

is not finitely generated!

Corollary
The monoid M cannot be presented by a finite convergent presentation.

An analogy with topos theory

monoid	topos
presentation $\langle G \mid R \rangle$	site $(\mathcal{C}, J)$
presented monoid G^*/R	sheaves $\mathbf{Sh}(\mathcal{C}, J)$
Tietze equivalence	Morita equivalence
Tietze transformation	$\sim$ comparison lemma
convergent presentation	?
$\vdots$	$\vdots$

It seems that Tietze transformations are “deformations” of presentations.

Can we make this formal?

A model structure on presentations

Theorem (Henry-M.)
There is a (cofibrantly generated) model structure on the category of (reflexive) presentations of monoids where weak equivalences are morphisms $f : \langle G \mid R \rangle \to \langle G' \mid R' \rangle$ *inducing isomorphism of presented monoids, i.e.* $G^*/R \simeq G'^*/R'$.

A model structure on presentations

The generating cofibrations are

- $\langle | \rangle \hookrightarrow \langle a \mid \rangle$
- $\langle a_1, \ldots, a_n, b_1, \ldots, b_m \mid \rangle \hookrightarrow \langle a_1, \ldots, a_n, b_1, \ldots, b_m \mid a_1 \ldots a_n = b_1 \ldots b_m \rangle$

Proposition
Every object is cofibrant and cofibrations are precisely monomorphisms.

A model structure on presentations

We expect that the generating trivial cofibrations are

- $\langle a_1, \ldots, a_n \mid \rangle \hookrightarrow \langle a_1, \ldots, a_n, b \mid a_1 \ldots a_n = b \rangle$,
- $\langle a_1, \ldots, a_n \mid \rangle \hookrightarrow \langle a_1, \ldots, a_n \mid a_1 \ldots a_n = a_1 \ldots a_n \rangle$,
- + transitivity, symmetry and congruence

so that generated trivial cofibrations are precisely (retracts of) Tietze transformations.

This is “almost” the case, in the sense that those generate trivial cofibrations when the target is fibrant, and we can recover abstractly Tietze theorem.

Part III

Generalization to higher categories

Generalizations

The technology of rewriting extends to many other settings:

- universal algebra / Lawvere theories / clones (term rewriting systems)
- operads
- commutative (or not) rings (Gröbner basis)
- etc.

Can we come up with a general definition of higher-dimensional rewriting system?

Generalizations

The technology of rewriting extends to many other settings:

- universal algebra / Lawvere theories / clones (term rewriting systems)
- operads
- commutative (or not) rings (Gröbner basis)
- etc.

Can we come up with a general definition of higher-dimensional rewriting system?

higher-dimensional rewriting

=

rewriting between rewriting paths between rewriting paths between …

0-dimensional rewriting systems

Recall that an abstract rewriting system is a graph

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

with $R \subseteq G \times G$.

0-dimensional rewriting systems

Recall that an abstract rewriting system is a graph

$$\langle P_0 \mid P_1 \rangle$$

with $P_1 \subseteq P_0 \times P_0$.

0-dimensional rewriting systems

Recall that an abstract rewriting system is a graph

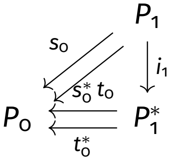
$$P_0 \underset{t_0}{\overset{s_0}{\leftleftarrows}} P_1$$

0-dimensional rewriting systems

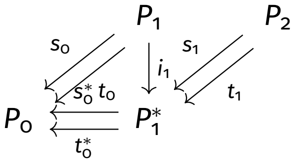
Recall that an abstract rewriting system is a graph

$$P_0 \underset{t_0}{\overset{s_0}{\leftleftarrows}} P_1$$

We write P_1^* for the set of rewriting paths:

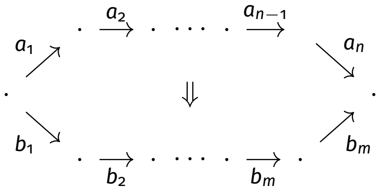


An string rewriting system is

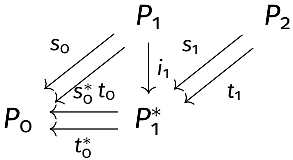


such that $s_0^* s_1 = s_0^* t_1$ and $t_0^* s_1 = t_0^* t_1$.

An element of P_2 is seen as



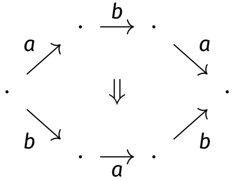
An string rewriting system is



such that $s_0^* s_1 = s_0^* t_1$ and $t_0^* s_1 = t_0^* t_1$.

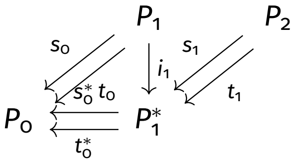
For instance,

$$\{\star\} \underset{t_0^*}{\overset{s_0^*}{\leftleftarrows}} \{a, b\}^*$$



2-dimensional rewriting systems

A 1-dimensional rewriting system is



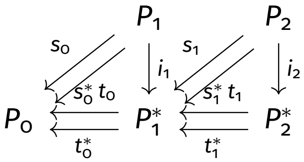
It presents the category

$$P_1^*/P_2$$

...but it can also be seen as a generating system for a 2-category!

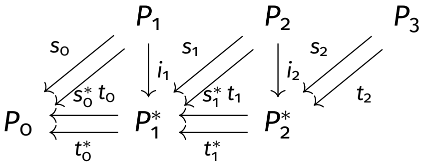
2-dimensional rewriting systems

A 1-dimensional rewriting system is



where P_2^* is the set of rewriting paths / 2-cells.

A 2-dimensional rewriting system is



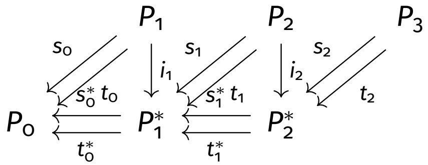
together with the structure of 2-category on the diagram on the bottom line.

(aka polygraph or computad)

It presents a 2-category.

Monoids

For instance, we can take

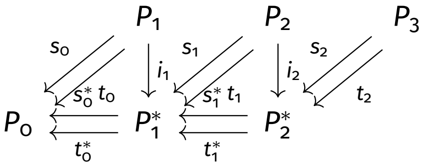


with

- $P_0 = \{\star\}$

Monoids

For instance, we can take

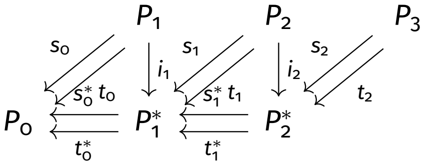


with

- $P_0 = \{\star\}$
- $P_1 = \{1\}$

Monoids

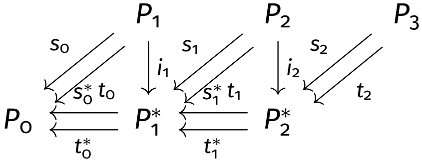
For instance, we can take



with

- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{m : 2 \to 1, e : 0 \to 1\}$

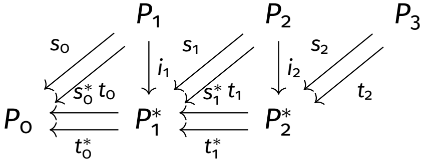
For instance, we can take



with

- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{m : 2 \to 1, e : 0 \to 1\}$
- $P_3 = \{\alpha : (m * 1) * m \Rightarrow (1 * m) * m, \lambda : (e * 1) * m \Rightarrow 1, \rho : (1 * e) * m \Rightarrow 1\}$

For instance, we can take



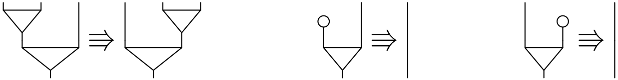
with

- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{m : 2 \to 1, e : 0 \to 1\}$
- $P_3 = \{\alpha : (m * 1) * m \Rightarrow (1 * m) * m, \lambda : (e * 1) * m \Rightarrow 1, \rho : (1 * e) * m \Rightarrow 1\}$

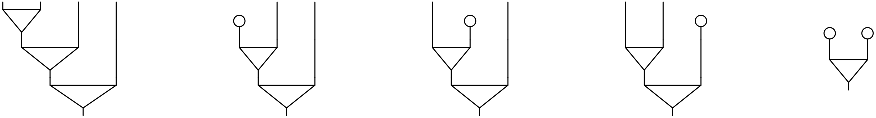
A functor $\overline{P} \to$ **Cat** is a strict monoidal category.

Monoids

The rules are



and there are 5 critical branchings:



A finite rewriting system can lead to an infinite number of critical branchings.

Example (Guiraud-Malbos'09)
Consider

- $P_0 = \{\star\}$

A finite rewriting system can lead to an infinite number of critical branchings.

Example (Guiraud-Malbos'09)
Consider

- $P_0 = \{\star\}$
- $P_1 = \{1\}$

2-dimensional rewriting systems

A finite rewriting system can lead to an infinite number of critical branchings.

Example (Guiraud-Malbos'09)
Consider

- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{n : 0 \to 2, u : 0 \to 1, o : 1 \to 1\}$

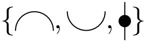
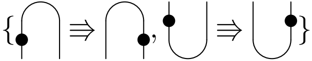
A finite rewriting system can lead to an infinite number of critical branchings.

Example (Guiraud-Malbos'09)
Consider

- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{n : 0 \to 2, u : 0 \to 1, o : 1 \to 1\} = \{\frown, \smile, \bullet\}$

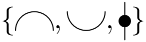
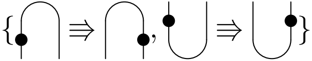
A finite rewriting system can lead to an infinite number of critical branchings.

Example (Guiraud-Malbos'09)
Consider

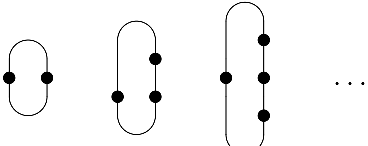
- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{n : 0 \to 2, u : 0 \to 1, o : 1 \to 1\} =$ 
- $P_3 =$ 

A finite rewriting system can lead to an infinite number of critical branchings.

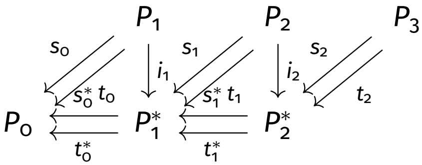
Example (Guiraud-Malbos'09)
Consider

- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{n : 0 \to 2, u : 0 \to 1, o : 1 \to 1\} =$ 
- $P_3 =$ 

We have an infinite family of critical branchings:

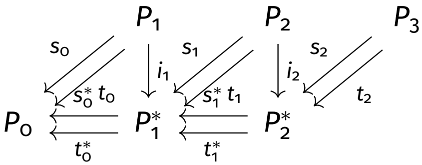


An 2-dimensional rewriting system is



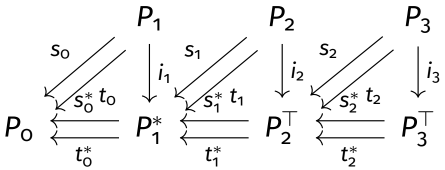
Coherent 1-dimensional rewriting systems

An extended 1-dimensional rewriting system is



Coherent 1-dimensional rewriting systems

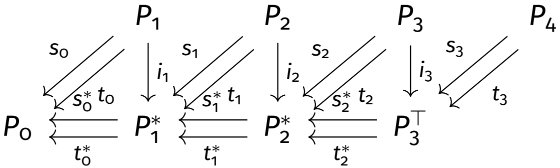
An extended 1-dimensional rewriting system is



Theorem (Squier)
If we take $\mathbf{P}_3$ generated by critical branchings then the extended rs is coherent: there is an invertible 3-cell in $\mathbf{P}_3^\top$ between any parallel pair of 2-cells in $\mathbf{P}_2^*$. Thus, $\mathbf{P}$ has finite derivation type.

Coherent 2-dimensional rewriting systems

Similarly, an extended 2-dimensional rewriting system



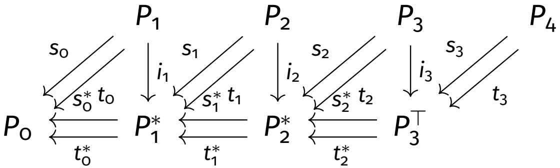
where P_4 is generated by critical branchings is coherent.

Applying this to the rs of monoids, we can recover MacLane's coherence theorem:

- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{m : 2 \to 1, e : 0 \to 1\}$
- $P_3 = \{\alpha : (m * 1) * m \Rightarrow (1 * m) * m, \lambda : (e * 1) * m \Rightarrow 1, \rho : (1 * e) * m \Rightarrow 1\}$
- $P_4 = \{5 \text{ elements}\}$

Coherent 2-dimensional rewriting systems

Similarly, an extended 2-dimensional rewriting system



where P_4 is generated by critical branchings is coherent.

Applying this to the rs of monoids, we can recover MacLane's coherence theorem:

- $P_0 = \{\star\}$
- $P_1 = \{1\}$
- $P_2 = \{m : 2 \to 1, e : 0 \to 1\}$
- $P_3 = \{\alpha : (m * 1) * m \Rightarrow (1 * m) * m, \lambda : (e * 1) * m \Rightarrow 1, \rho : (1 * e) * m \Rightarrow 1\}$
- $P_4 = \{2 \text{ elements}\}$

Theorem (Lafont-Métayer-Worytkiewicz'10)
*There is a model structure on ω-**Cat** where*

- *equivalences are categorical equivalences,*
- *every object is fibrant,*
- *cofibrant objects are categories generated by polygraphs.*