

Generalisations of
Small Cancellation
Theory

Max Neunhöffer

Van Kampen
Diagrams

The Word Problem

Diagrams

Isoperimetric Functions

Small Cancellation
Theory

Conditions

Theorems

Generalisations

Algorithmic approach

Forbidden Subdiagrams

Curvature redistribution

2-dimensional analysis

1-dimensional analysis

Changing the presentation

The Prototype

Generalisations of Small Cancellation Theory

Max Neunhöffer



University of St Andrews

Allahabad 11.9.2010

Status of the Project

Joint work with:

Stephen Linton

Richard Parker

Colva Roney-Dougal

Richard already worked for several years on this.

The whole project is still in its early stages.

The Word Problem

Problem

Let $G = \langle X \mid R \rangle$ be a **finitely presented group**,

The Word Problem

Problem

Let $G = \langle X \mid R \rangle$ be a **finitely presented group**,
i.e. $G \cong F(X)/N$, where $N := \langle\langle R \rangle\rangle$ is the **normal closure**
of R in the **free group** $F(X)$ on X .

The Word Problem

Problem

Let $G = \langle X \mid R \rangle$ be a **finitely presented group**,
i.e. $G \cong F(X)/N$, where $N := \langle\langle R \rangle\rangle$ is the **normal closure**
of R in the **free group** $F(X)$ on X .

Decide whether or not a word $w \in F(X)$ lies in N , that is,
whether or not it represents the identity in G .

The Word Problem

Problem

Let $G = \langle X \mid R \rangle$ be a **finitely presented group**,
i.e. $G \cong F(X)/N$, where $N := \langle\langle R \rangle\rangle$ is the **normal closure**
of R in the **free group** $F(X)$ on X .

Decide whether or not a word $w \in F(X)$ lies in N , that is,
whether or not it represents the identity in G .

Of course:

$$w \in N \quad \iff \quad w = \prod_{i=1}^k u_i r_i u_i^{-1} \quad \text{in } F(X)$$

for some $k \in \mathbb{N} \cup \{0\}$, $r_i \in R \cup R^{-1}$ and $u_i \in F(X)$.

Want: An **algorithm** to solve this **Word Problem** in G .

Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

Van Kampen Diagram example

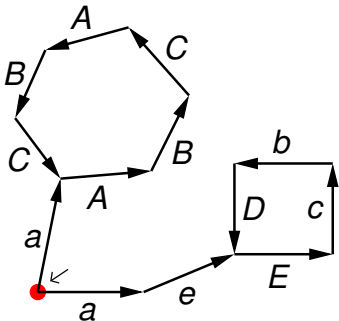
$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)

Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

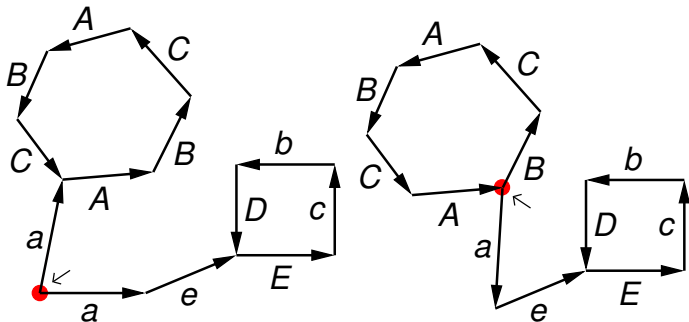
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

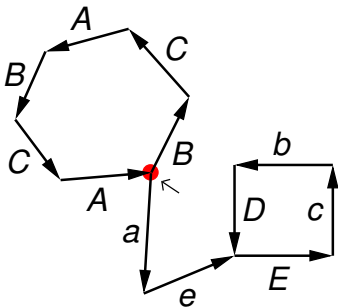
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

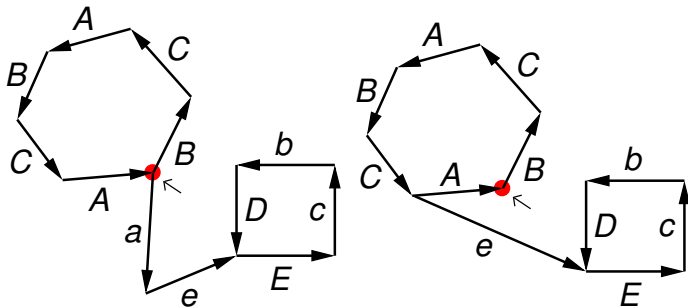
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

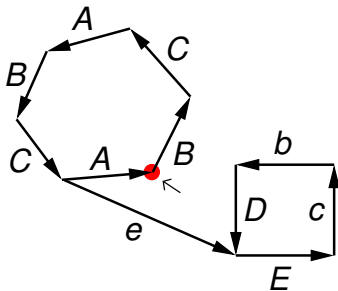
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

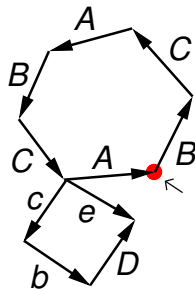
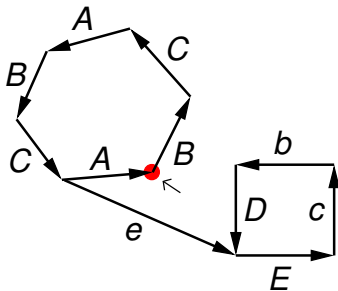
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

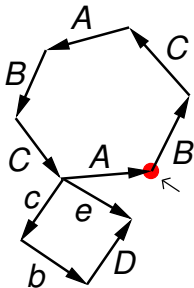
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

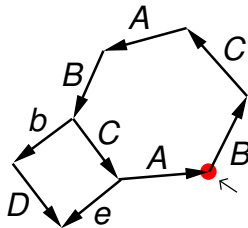
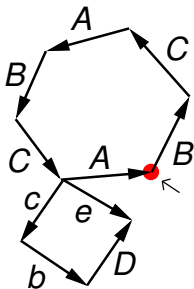
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

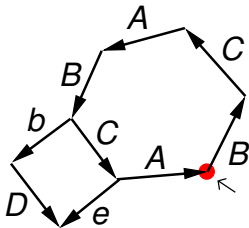
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

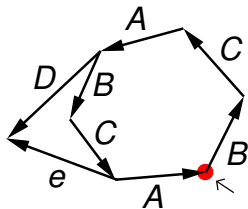
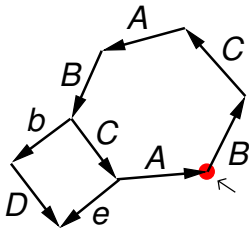
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

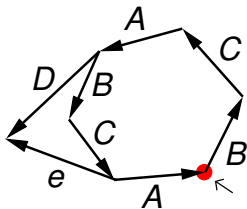
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

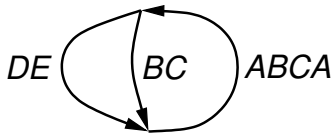
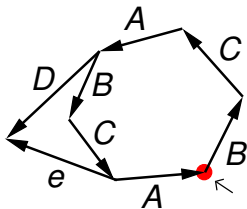
Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)

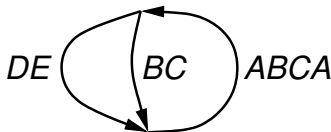


"reduced"

Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



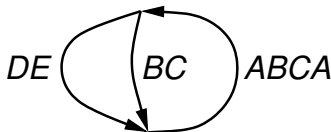
Definition

A **van Kampen diagram** is a connected, simply-connected 2-complex with oriented and labelled edges.

Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Definition

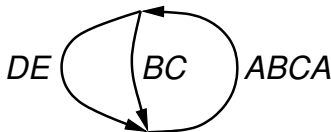
A **van Kampen diagram** is a connected, simply-connected 2-complex with oriented and labelled edges.

It is a topological proof for a word to be in N .

Van Kampen Diagram example

$$G = \langle A, B, C, D, E \mid ABCABC = EcbD = 1 \rangle$$

Thus $BCADEA = (a \cdot ABCABC \cdot A) \cdot (ae \cdot EcbD \cdot EA)$.
(small letters are the inverses of corresponding capitals)



Definition

A **van Kampen diagram** is a connected, simply-connected 2-complex with oriented and labelled edges.

It is a **topological proof** for a word to be in N .

It is a **sphere** if the boundary word is a relator.

Van Kampen Diagrams II

Theorem (Van Kampen, 1933)

Let $G = \langle X \mid R \rangle$ be a finitely presented group.

Then a word $w \in F(X)$ is contained in the normal closure $N := \langle\langle R \rangle\rangle$ of R in $F(X)$ if and only if there is a van Kampen diagram with w as boundary word.

Van Kampen Diagrams II

Theorem (Van Kampen, 1933)

Let $G = \langle X \mid R \rangle$ be a finitely presented group.
Then a word $w \in F(X)$ is *contained in the normal closure*
 $N := \langle\langle R \rangle\rangle$ of R in $F(X)$ if and only if there is a *van
Kampen diagram with w as boundary word*.

Definition (Isoperimetric (or Dehn) Function)

A function $f : \mathbb{N} \rightarrow \mathbb{N}$ is called an *isoperimetric function* or
Dehn function for $G = \langle X \mid R \rangle$ if the following holds:

for every reduced word w of length at most n
representing 1 in G there is a van Kampen diagram
proving this *with at most $f(n)$ faces*.

Isoperimetric Functions

Theorem (Gersten)

*The word problem for $G = \langle X \mid R \rangle$ is **solvable** if and only if $\langle X \mid R \rangle$ has a **recursive isoperimetric function**.*

Isoperimetric Functions

Theorem (Gersten)

The word problem for $G = \langle X \mid R \rangle$ is *solvable* if and only if $\langle X \mid R \rangle$ has a *recursive isoperimetric function*.

Theorem (Gromov)

For a *finitely generated group* G are equivalent:

- G has a *finite presentation* $\langle X \mid R \rangle$ with a *linear isoperimetric function*.

Isoperimetric Functions

Theorem (Gersten)

The word problem for $G = \langle X \mid R \rangle$ is *solvable* if and only if $\langle X \mid R \rangle$ has a *recursive isoperimetric function*.

Theorem (Gromov)

For a *finitely generated group* G are equivalent:

- G has a *finite presentation* $\langle X \mid R \rangle$ with a *linear isoperimetric function*.
- G is a *word hyperbolic group*.

Isoperimetric Functions

Theorem (Gersten)

The word problem for $G = \langle X \mid R \rangle$ is *solvable* if and only if $\langle X \mid R \rangle$ has a *recursive isoperimetric function*.

Theorem (Gromov)

For a *finitely generated group* G are equivalent:

- G has a *finite presentation* $\langle X \mid R \rangle$ with a *linear isoperimetric function*.
- G is a *word hyperbolic group*.
- G has a *finite presentation* $\langle X \mid R \rangle$ for which *Dehn's algorithm* solves the word problem.

Isoperimetric Functions

Theorem (Gersten)

The word problem for $G = \langle X \mid R \rangle$ is *solvable* if and only if $\langle X \mid R \rangle$ has a *recursive isoperimetric function*.

Theorem (Gromov)

For a *finitely generated group* G are equivalent:

- G has a *finite presentation* $\langle X \mid R \rangle$ with a *linear isoperimetric function*.
- G is a *word hyperbolic group*.
- G has a *finite presentation* $\langle X \mid R \rangle$ for which *Dehn's algorithm* solves the word problem.

Dehn's algorithm to solve the word problem

Whenever a cyclic word w contains *more than half of a relator*, replace this part by the inverse of the other, smaller part.

Conditions

Let $G = \langle X \mid R \rangle$ be a finitely presented group.

Let \overline{R} be R together with **all cyclic rotations and their inverses** of all relators.

Conditions

Let $G = \langle X \mid R \rangle$ be a finitely presented group.
Let \bar{R} be R together with **all cyclic rotations and their inverses** of all relators.

Condition $T(k)$ for $k \geq 3$

If the presentation fulfills condition $T(k)$, then every **internal vertex** of a reduced van Kampen diagram has **valency $\geq k$** .

Conditions

Let $G = \langle X \mid R \rangle$ be a finitely presented group.
Let \bar{R} be R together with **all cyclic rotations and their inverses** of all relators.

Condition $T(k)$ for $k \geq 3$

If the presentation fulfills condition $T(k)$, then every **internal vertex** of a reduced van Kampen diagram has **valency $\geq k$** .

Definition (Piece)

A word $1 \neq b \in F(X)$ is called a **piece** if there are relators $r_1, r_2 \in \bar{R}$ with $r_1 = bc_1$ and $r_2 = bc_2$ for $c_1, c_2 \in F(X)$ with $c_1 \neq c_2$.

Conditions

Let $G = \langle X \mid R \rangle$ be a finitely presented group.
Let \bar{R} be R together with **all cyclic rotations and their inverses** of all relators.

Condition $T(k)$ for $k \geq 3$

If the presentation fulfills condition $T(k)$, then every **internal vertex** of a reduced van Kampen diagram has **valency $\geq k$** .

Definition (Piece)

A word $1 \neq b \in F(X)$ is called a **piece** if there are relators $r_1, r_2 \in \bar{R}$ with $r_1 = bc_1$ and $r_2 = bc_2$ for $c_1, c_2 \in F(X)$ with $c_1 \neq c_2$.

Condition $C'(\lambda)$ for $0 < \lambda < 1$

If $bc \in \bar{R}$ with a piece b , then **$\ell(b) < \lambda \ell(bc)$** .

Small Cancellation Theorems

Theorem (Small Cancellation)

If a presentation $G = \langle X \mid R \rangle$ fulfills

- $T(3)$ and $C'(1/6)$, or
- $T(4)$ and $C'(1/4)$,

then *Dehn's algorithm* solves the word problem in G .

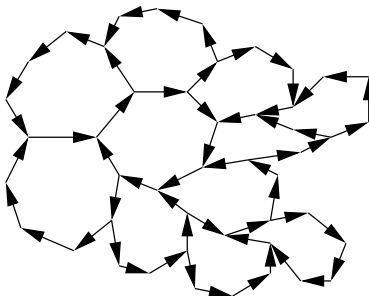
Small Cancellation Theorems

Theorem (Small Cancellation)

If a presentation $G = \langle X \mid R \rangle$ fulfills

- $T(3)$ and $C'(1/6)$, or
- $T(4)$ and $C'(1/4)$,

then *Dehn's algorithm* solves the word problem in G .



Algorithmic approach

Small Cancellation Theory is **very much static**.

Algorithmic approach

Small Cancellation Theory is **very much static**.

Idea 1: Analyse presentation dynamically

We want to write programs that analyse presentations and — if they succeed — come up with **new, possibly more complicated local conditions** which show that **some word problem solving algorithm** works.

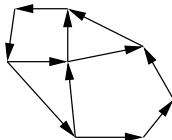
This is a very **dynamic and flexible** approach.

Forbidden Subdiagrams

Already forbidden: two faces touching with inverse words.

Forbidden Subdiagrams

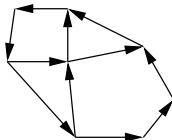
Already forbidden: two faces touching with inverse words.



Assume this is a **sphere**, i.e. the **boundary is a relator**.

Forbidden Subdiagrams

Already forbidden: two faces touching with inverse words.

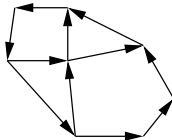


Assume this is a **sphere**, i.e. the **boundary** is a **relator**.

⇒ Can replace this subdiagram by **one heptagon!**

Forbidden Subdiagrams

Already forbidden: two faces touching with inverse words.



Assume this is a **sphere**, i.e. the **boundary** is a **relator**.

⇒ Can replace this subdiagram by **one heptagon!**

Idea 2: Forbidden regions

We simply **forbid** certain subdiagrams (once we proved that they are not needed!) and check the local conditions **only on the vK diagrams without forbidden regions.**

Curvature redistribution

Idea 3:

To make a **sphere**, the surface must be **curved**.

Curvature redistribution

Idea 3:

To make a **sphere**, the surface must be **curved**.

In a planar graph, Euler's formula holds:

$$(\# \text{Vertices}) - (\# \text{Edges}) + (\# \text{Regions}) = 1$$

(not counting the “outside region”).

Curvature redistribution

Idea 3:

To make a **sphere**, the surface must be **curved**.

In a planar graph, Euler's formula holds:

$$(\# \text{Vertices}) - (\# \text{Edges}) + (\# \text{Regions}) = 1$$

(not counting the “outside region”).

Combinatorial Curvature

If we endow every vertex with $+1$, every edge with -1 and every face with $+1$ units of “**curvature**”, the total sum in a diagram is always equal to 1.

This way, **curvature** is a **local phenomenon**, about which we have **global information**.

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to move curvature around **locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to move curvature around **locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).
- Distribute the curvature of each **edge** equally to its **adjacent vertices** (E).

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to **move curvature around locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).
- Distribute the curvature of each **edge** equally to its **adjacent vertices** (E).
- Distribute the curvature of each **vertex** equally to its **adjacent faces** (A).

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to **move curvature around locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).
- Distribute the curvature of each **edge** equally to its **adjacent vertices** (E).
- Distribute the curvature of each **vertex** equally to its **adjacent faces** (A).

Or any sequence of them, e.g. LE or LEA or LEALEA.

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to move curvature around **locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).
- Distribute the curvature of each **edge** equally to its **adjacent vertices** (E).
- Distribute the curvature of each **vertex** equally to its **adjacent faces** (A).

Or any sequence of them, e.g. LE or LEA or LEALEA.

To which end?

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to move curvature around **locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).
- Distribute the curvature of each **edge** equally to its **adjacent vertices** (E).
- Distribute the curvature of each **vertex** equally to its **adjacent faces** (A).

Or any sequence of them, e.g. LE or LEA or LEALEA.

To which end?

- In the end, the sum is still 1.

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to move curvature around **locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).
- Distribute the curvature of each **edge** equally to its **adjacent vertices** (E).
- Distribute the curvature of each **vertex** equally to its **adjacent faces** (A).

Or any sequence of them, e.g. LE or LEA or LEALEA.

To which end?

- In the end, the sum is still 1.
- **If we can show that after the move, the curvature in the interior is always non-positive then**

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to move curvature around **locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).
- Distribute the curvature of each **edge** equally to its **adjacent vertices** (E).
- Distribute the curvature of each **vertex** equally to its **adjacent faces** (A).

Or any sequence of them, e.g. LE or LEA or LEALEA.

To which end?

- In the end, the sum is still 1.
- **If we can show that after the move, the curvature in the interior is always non-positive** then
 - there are no spheres, and

Curvature redistribution

Idea 4: Move curvature around — Officers

We now start to move curvature around **locally**, e.g.:

- Distribute the curvature of each **face** to the **adjacent edges** according to their length (L).
- Distribute the curvature of each **edge** equally to its **adjacent vertices** (E).
- Distribute the curvature of each **vertex** equally to its **adjacent faces** (A).

Or any sequence of them, e.g. LE or LEA or LEALEA.

To which end?

- In the end, the sum is still 1.
- **If we can show that after the move, the curvature in the interior is always non-positive** then
 - there are no spheres, and
 - in all vK diagrams, **the curvature is at the boundary.**

2-dimensional analysis by example

This analyses the **interior** of possible diagrams to conclude that an officer leaves **no curvature in the interior**.

Example: LE officer with $T(4)$ and $C'(1/4)$

Assume $T(4)$ and $C'(1/4)$ and run officer LE.

Curvature distribution (interior):

when	int. vertex	int. edge	face
initial	1	-1	1

2-dimensional analysis by example

This analyses the **interior** of possible diagrams to conclude that an officer leaves **no curvature in the interior**.

Example: LE officer with $T(4)$ and $C'(1/4)$

Assume $T(4)$ and $C'(1/4)$ and run officer LE.

Curvature distribution (interior):

when	int. vertex	int. edge	face
initial	1	-1	1
after L	1	$< -1/2 (*)$	0

2-dimensional analysis by example

This analyses the **interior** of possible diagrams to conclude that an officer leaves **no curvature in the interior**.

Example: LE officer with $T(4)$ and $C'(1/4)$

Assume $T(4)$ and $C'(1/4)$ and run officer LE.

Curvature distribution (interior):

when	int. vertex	int. edge	face
initial	1	-1	1
after L	1	$< -1/2$ (*)	0
after E	< 0 (**)	0	0

(*) since $-1/2 = -1 + 2 \cdot \frac{1}{4}$ using $C'(1/4)$.

(**) since $0 = 1 + 4 \cdot \frac{-1/2}{2}$ using $T(4)$.

1-dimensional analysis

This analyses the **boundary** of possible diagrams to use positive curvature near the boundary to **prove a word problem solver**.

1-dimensional analysis

This analyses the **boundary** of possible diagrams to use positive curvature near the boundary to **prove a word problem solver**.

Basically construct **all possible regions** around a boundary vertex with **positive curvature**.

1-dimensional analysis

This analyses the **boundary** of possible diagrams to use positive curvature near the boundary to **prove a word problem solver**.

Basically construct **all possible regions** around a boundary vertex with **positive curvature**.

Show that your favourite word problem solver can deal with all situations arising.

Changing the presentation

Idea 5:

As part of our dynamic, algorithmic approach, we **change the presentation** (without changing the group!).

Changing the presentation

Idea 5:

As part of our dynamic, algorithmic approach, we **change the presentation** (without changing the group!).

We might for example:

- Add **implied relators** to **finish spheres** and thereby **create new forbidden regions**.

Changing the presentation

Idea 5:

As part of our dynamic, algorithmic approach, we **change the presentation** (without changing the group!).

We might for example:

- Add **implied relators** to **finish spheres** and thereby **create new forbidden regions**.
- Leave out **generators shown to be not needed** to **simplify relators and diagrams**.

Changing the presentation

Idea 5:

As part of our dynamic, algorithmic approach, we **change the presentation** (without changing the group!).

We might for example:

- Add **implied relators** to **finish spheres** and thereby **create new forbidden regions**.
- Leave out **generators shown to be not needed** to **simplify relators and diagrams**.
- Add **generators and relations** to **change the geometry of diagrams** (e.g. triangulation).

Changing the presentation

Idea 5:

As part of our dynamic, algorithmic approach, we **change the presentation** (without changing the group!).

We might for example:

- Add **implied relators** to **finish spheres** and thereby **create new forbidden regions**.
- Leave out **generators shown to be not needed** to **simplify relators and diagrams**.
- Add **generators and relations** to **change the geometry of diagrams** (e.g. triangulation).
- Do **all of these** to make **new local small cancellation conditions** applicable.

The Prototype

- Only allows **triangle relators**.

The Prototype

- Only allows **triangle relators**.
- Tries to verify that **no internal vertices of valency < 6** are needed in van Kampen diagrams.

The Prototype

- Only allows **triangle relators**.
- Tries to verify that **no internal vertices of valency < 6** are needed in van Kampen diagrams.
- **Adds gens and rels** until conditions fulfilled.

The Prototype

- Only allows **triangle relators**.
- Tries to verify that **no internal vertices of valency < 6** are needed in van Kampen diagrams.
- **Adds gens and rels** until conditions fulfilled.
- For **some groups** it **explodes** and does not work.

The Prototype

- Only allows **triangle relators**.
- Tries to verify that **no internal vertices of valency < 6** are needed in van Kampen diagrams.
- **Adds gens and rels** until conditions fulfilled.
- For **some groups** it **explodes** and does not work.
- **If it terminates**, it proves that a certain **non-length-reducing word problem solver** works for the output presentation.

The Prototype

- Only allows **triangle relators**.
- Tries to verify that **no internal vertices of valency < 6** are needed in van Kampen diagrams.
- **Adds gens and rels** until conditions fulfilled.
- For **some groups** it **explodes** and does not work.
- **If it terminates**, it proves that a certain **non-length-reducing word problem solver** works for the output presentation.
- Highly optimised **C implementation**.

The Prototype

- Only allows **triangle relators**.
- Tries to verify that **no internal vertices of valency < 6** are needed in van Kampen diagrams.
- **Adds gens and rels** until conditions fulfilled.
- For **some groups** it **explodes** and does not work.
- **If it terminates**, it proves that a certain **non-length-reducing word problem solver** works for the output presentation.
- Highly optimised **C implementation**.

# gens	# random rels	time [seconds]
1 000	1 140	0.03
10 000	20 010	0.77
100 000	366 500	33
500 000	2 830 000	584
1 000 000	6 850 000	2145