Algorithmic Generalisations of Small Cancellation Theory

Max Neunhöffer

joint work with Jeffrey Burdges, Stephen Linton,

Richard Parker and Colva Roney-Dougal

2nd Annual Meeting of the DFG Priority Programm SPP1489,
Hannover, 27 February – 1 March 2012

Question

What can you tell me about the finitely presented group

$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle ?$$

Question

What can you tell me about the finitely presented group

$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle ?$$

(You may use a computer for this exercise!)

Question

What can you tell me about the finitely presented group

$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle ?$$

(You may use a computer for this exercise!)

- It is a quotient of the modular group $\mathrm{PSL}_2(\mathbb{Z}) \cong \langle S, T \mid S^3, T^2 \rangle$.

Question

What can you tell me about the finitely presented group

$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle?$$

(You may use a computer for this exercise!)

- It is a quotient of the modular group $\mathrm{PSL}_2(\mathbb{Z}) \cong \langle S, T \mid S^3, T^2 \rangle$.
- It has $\mathrm{PSL}_2(\mathbb{F}_{13})$ and $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotients.

Question

What can you tell me about the finitely presented group

$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle?$$

(You may use a computer for this exercise!)

- It is a quotient of the modular group $\mathrm{PSL}_2(\mathbb{Z}) \cong \langle S, T \mid S^3, T^2 \rangle$.
- It has $\mathrm{PSL}_2(\mathbb{F}_{13})$ and $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotients.
- Other finite quotients can be found (low index method).

Question

What can you tell me about the finitely presented group

$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle?$$

(You may use a computer for this exercise!)

- It is a quotient of the modular group $\mathrm{PSL}_2(\mathbb{Z}) \cong \langle S, T \mid S^3, T^2 \rangle$.
- It has $\mathrm{PSL}_2(\mathbb{F}_{13})$ and $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotients.
- Other finite quotients can be found (low index method).
- Eventually it turns out to be infinite (abelian invariants method).

Question

What can you tell me about the finitely presented group

$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle ?$$

(You may use a computer for this exercise!)

- It is a quotient of the modular group $\mathrm{PSL}_2(\mathbb{Z}) \cong \langle S, T \mid S^3, T^2 \rangle$.
- It has $\mathrm{PSL}_2(\mathbb{F}_{13})$ and $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotients.
- Other finite quotients can be found (low index method).
- Eventually it turns out to be infinite (abelian invariants method).
- Todd-Coxeter is not of much use here.

Question

What can you tell me about the finitely presented group

$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle?$$

(You may use a computer for this exercise!)

- It is a quotient of the modular group $\mathrm{PSL}_2(\mathbb{Z}) \cong \langle S, T \mid S^3, T^2 \rangle$.
- It has $\mathrm{PSL}_2(\mathbb{F}_{13})$ and $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotients.
- Other finite quotients can be found (low index method).
- Eventually it turns out to be infinite (abelian invariants method).
- Todd-Coxeter is not of much use here.
- Knuth-Bendix does not help either.

Question

What can you tell me about the finitely presented group

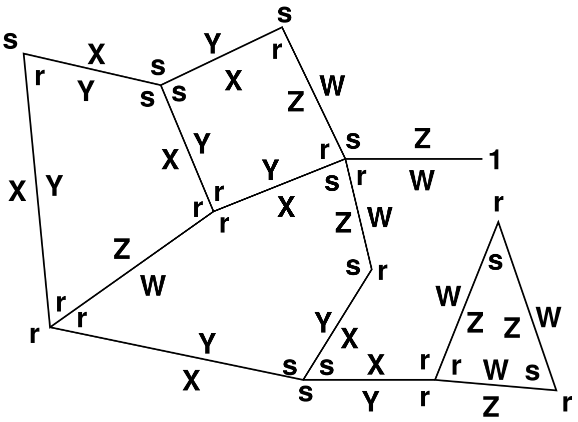
$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle ?$$

(You may use a computer for this exercise!)

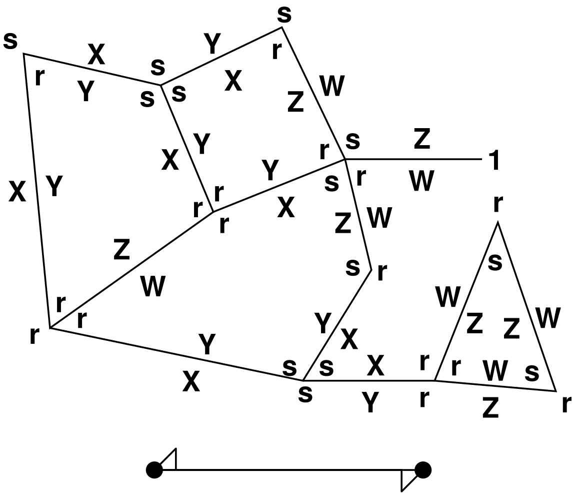
- It is a quotient of the modular group $\mathrm{PSL}_2(\mathbb{Z}) \cong \langle S, T \mid S^3, T^2 \rangle$.
- It has $\mathrm{PSL}_2(\mathbb{F}_{13})$ and $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotients.
- Other finite quotients can be found (low index method).
- Eventually it turns out to be infinite (abelian invariants method).
- Todd-Coxeter is not of much use here.
- Knuth-Bendix does not help either.

Can we solve the word problem?

We draw connected finite graphs in the plane and label them:



We draw connected finite graphs in the plane and label them:



Edges are pairs of directed edges which are labelled by 2 letters each.

Diagram problems

Let R be a finite set of cyclic words, called relators.

Diagram problems

Let R be a finite set of cyclic words, called relators.

Problem (Diagram boundary problem)

Devise (algorithmically) a procedure that decides for any cyclic word w, whether or not there is a diagram such that

- *every internal region is labelled by a relator, and*
- *the external boundary is labelled by w.*

Diagram problems

Let R be a finite set of cyclic words, called relators.

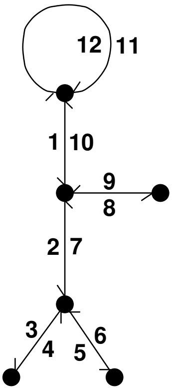
Problem (Diagram boundary problem)

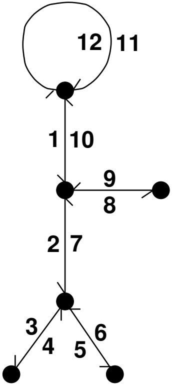
Devise (algorithmically) a procedure that decides for any cyclic word w, whether or not there is a diagram such that

- *every internal region is labelled by a relator, and*
- *the external boundary is labelled by w.*

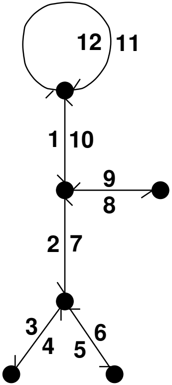
Problem (Isoperimetric inequality)

Find and prove (algorithmically) a function $f : \mathbb{N} \to \mathbb{N}$, such that for every cyclic word w of length k that is the boundary label of a diagram, there is one with at most $f(k)$ internal regions.


12
11
1
10
9
8
2
7
3
4
5
6

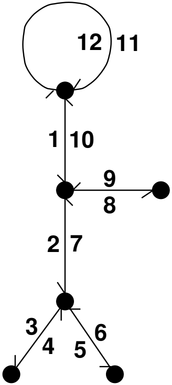


#	E	F	V
1	10	2	12
2	7	3	10
3	4	4	7
4	3	5	4
5	6	6	3
6	5	7	6
7	2	8	5
8	9	9	2
9	8	10	9
10	1	11	8
11	12	1	1
12	11	12	11



#	E	F	V
1	10	2	12
2	7	3	10
3	4	4	7
4	3	5	4
5	6	6	3
6	5	7	6
7	2	8	5
8	9	9	2
9	8	10	9
10	1	11	8
11	12	1	1
12	11	12	11

$\{$finite connected planar embedded graphs$\}/\sim$ is in bijection with



#	E	F	V
1	10	2	12
2	7	3	10
3	4	4	7
4	3	5	4
5	6	6	3
6	5	7	6
7	2	8	5
8	9	9	2
9	8	10	9
10	1	11	8
11	12	1	1
12	11	12	11

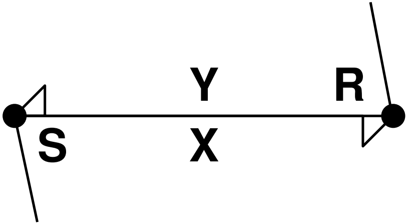
$\{$finite connected planar embedded graphs$\}/\sim$ is in bijection with

$$\{(E,F,V)\in S_n^3 \;\mid\; n\in\mathbb{N}, EFV=1, \langle E,F\rangle \text{ is transitive, } \#\text{cycles of } E, F \text{ and } V \text{ is } n+2, E \text{ is a fixed-point free involution}\}/\sim$$

Rules for the labels

We label every half-edge with two symbols,

- one for the corner to the right of where it starts, and
- one for the right hand side of it:



We now need rules for the corner labels and the edge labels.

Definition (Pongos)

A pongo is a set P with a subset $P_+ \subset P$, such that $P_0 := P \dot{\cup} \{0\}$ is a semigroup with 0 and:

$$\text{if } xy \in P_+ \text{ for } x, y \in P, \text{ then } yx \in P_+.$$

The elements in P_+ are called acceptors.

Definition (Pongos)

A pongo is a set P with a subset $P_+ \subset P$, such that $P_0 := P \dot{\cup} \{0\}$ is a semigroup with 0 and:

$$\text{if } xy \in P_+ \text{ for } x, y \in P, \text{ then } yx \in P_+.$$

The elements in P_+ are called acceptors.

Lemma (Cyclicity)

Let P be a pongo, if $p_1 p_2 \cdots p_k \in P_+$, then all rotations $p_i p_{i+1} \cdots p_k p_1 p_2 \cdots p_{i-1} \in P_+$.

Definition (Pongos)

A pongo is a set P with a subset $P_+ \subset P$, such that $P_0 := P \dot{\cup} \{0\}$ is a semigroup with 0 and:

$$\text{if } xy \in P_+ \text{ for } x, y \in P, \text{ then } yx \in P_+.$$

The elements in P_+ are called acceptors.

Lemma (Cyclicity)

Let P be a pongo, if $p_1 p_2 \cdots p_k \in P_+$, then all rotations $p_i p_{i+1} \cdots p_k p_1 p_2 \cdots p_{i-1} \in P_+$.

Vertex rules

The corner labels are from a pongo P,

Definition (Pongos)

A pongo is a set P with a subset $P_+ \subset P$, such that $P_0 := P \dot{\cup} \{0\}$ is a semigroup with 0 and:

$$\text{if } xy \in P_+ \text{ for } x, y \in P, \text{ then } yx \in P_+.$$

The elements in P_+ are called acceptors.

Lemma (Cyclicity)

Let P be a pongo, if $p_1 p_2 \cdots p_k \in P_+$, then all rotations $p_i p_{i+1} \cdots p_k p_1 p_2 \cdots p_{i-1} \in P_+$.

Vertex rules

The corner labels are from a pongo P,
a V-cycle is valid, if the product of its corner labels is an acceptor.

Definition (Pongos)

A pongo is a set P with a subset $P_+ \subset P$, such that $P_0 := P\dot{\cup}\{0\}$ is a semigroup with 0 and:

$$\text{if } xy \in P_+ \text{ for } x, y \in P, \text{ then } yx \in P_+.$$

The elements in P_+ are called acceptors.

Lemma (Cyclicity)

Let P be a pongo, if $p_1 p_2 \cdots p_k \in P_+$, then all rotations $p_i p_{i+1} \cdots p_k p_1 p_2 \cdots p_{i-1} \in P_+$.

Vertex rules

The corner labels are from a pongo P,
a V-cycle is valid, if the product of its corner labels is an acceptor.

Using a finite pongo is equivalent to using a finite state automaton.

Definition (Edge alphabet)

An edge alphabet is a set A with an involution $\overline{} : A \to A$.

Definition (Edge alphabet)

An edge alphabet is a set A with an involution $\overline{} : A \to A$.

(This is actually a special case of a pongo.)

Definition (Edge alphabet)

An edge alphabet is a set A with an involution $\overline{} : A \to A$.

(This is actually a special case of a pongo.)

Edge rules

The edge labels are from an edge alphabet,
an E-cycle ($=$ edge) is valid, if $Y = \overline{X}$ for the two labels X and Y.

Definition (Edge alphabet)

An edge alphabet is a set A with an involution $\overline{} : A \to A$.

(This is actually a special case of a pongo.)

Edge rules

The edge labels are from an edge alphabet,
an E-cycle ($=$ edge) is valid, if $Y = \overline{X}$ for the two labels X and Y.

(For the experts:
This is a generalisation of the rules of van Kampen diagrams.)

Definition (Valid diagram)

Let P be a pongo and A be an edge alphabet. A valid diagram is: an $n \in \mathbb{N}$ and three permutations $E, F, V \in S_{\{1,2,\ldots,n\}}$ and a labelling function $\ell : \{1, \ldots, n\} \to P \times A, x \mapsto (\ell_P(x), \ell_A(x))$, such that

- $EFV = 1$,
- E is a fixed point free involution,
- $\langle E, F\rangle$ is a transitive subgroup of S_n,
- the total number of cycles in E, F and V is $n + 2$,
- $\ell_P(x) \cdot \ell_P(xV) \cdot \ell_P(xV^2) \cdot \cdots \in P_+$ for every V-cycle $x\,\langle V\rangle$, and
- $\ell_A(xE) = \overline{\ell_A(x)}$ for all E-cycles (x, xE).

Let P be a pongo and A be an edge alphabet.

Definition (Set of relators)

A set of relators R is a finite set of cyclic words in $P \times A$.

Let P be a pongo and A be an edge alphabet and R a set of relators.

Definition (Set of relators)

A set of relators R is a finite set of cyclic words in $P \times A$.

Let P be a pongo and A be an edge alphabet and R a set of relators.

Definition (Set of relators)

A set of relators R is a finite set of cyclic words in $P \times A$.

Problem (Diagram boundary problem)

Devise (algorithmically) a procedure that decides for any cyclic word w in $P \times A$, whether or not there is a valid diagram such that

- *every internal F-cycle is labelled by a relator, and*
- *the external F-cycle is labelled by w.*

Let P be a pongo and A be an edge alphabet and R a set of relators.

Definition (Set of relators)

A set of relators R is a finite set of cyclic words in $P \times A$.

Problem (Diagram boundary problem)

Devise (algorithmically) a procedure that decides for any cyclic word w in $P \times A$, whether or not there is a valid diagram such that

- *every internal F-cycle is labelled by a relator, and*
- *the external F-cycle is labelled by w.*

Problem (Isoperimetric inequality)

Find and prove (algorithmically) a function $f : \mathbb{N} \to \mathbb{N}$, such that for every cyclic word $w \in P \times A$ of length k that is the boundary label of a valid diagram, there is one with at most $f(k)$ internal F-cycles.

Let P be a pongo and A be an edge alphabet and R a set of relators.

Definition (Set of relators)

A set of relators R is a finite set of cyclic words in $P \times A$.

Problem (Diagram boundary problem)

Devise (algorithmically) a procedure that decides for any cyclic word w in $P \times A$, whether or not there is a valid diagram such that

- *every internal F-cycle is labelled by a relator, and*
- *the external F-cycle is labelled by w.*

Problem (Isoperimetric inequality)

Find and prove (algorithmically) a function $f : \mathbb{N} \to \mathbb{N}$, such that for every cyclic word $w \in P \times A$ of length k that is the boundary label of a valid diagram, there is one with at most $f(k)$ internal F-cycles.

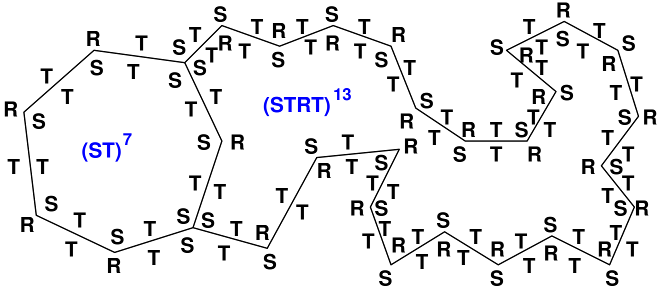
If f can be chosen linear, we call (P, A, R) hyperbolic.

$G := \langle S, R, T \mid SR, T^2, S^3, (ST)^7, (STS^2T)^{13} \rangle$ can be studied by:

$$
\begin{aligned}
P &= \{S, R, 1\} \text{ with } P_+ = \{1\} \text{ and } SR = RS = 1, SS = R, RR = S \\
A &= \{T\} \text{ with } \overline{T} = T \\
R &= \{((S,T),(S,T),(S,T),(S,T),(S,T),(S,T),(S,T)), \\
&\quad ((R,T),(R,T),(R,T),(R,T),(R,T),(R,T),(R,T)), \\
&\quad ((S,T),(R,T), \text{ repeated 13 times})\}
\end{aligned}
$$

$G := \langle S, R, T \mid SR, T^2, S^3, (ST)^7, (STS^2T)^{13} \rangle$ can be studied by:

$$
\begin{aligned}
P &= \{S, R, 1\} \text{ with } P_+ = \{1\} \text{ and } SR = RS = 1, SS = R, RR = S \\
A &= \{T\} \text{ with } \overline{T} = T \\
R &= \{((S,T),(S,T),(S,T),(S,T),(S,T),(S,T),(S,T)), \\
&\quad ((R,T),(R,T),(R,T),(R,T),(R,T),(R,T),(R,T)), \\
&\quad ((S,T),(R,T), \text{ repeated 13 times})\}
\end{aligned}
$$



$$(ST)^7(TRTS)^{13} = (ST)^5(STST)(TRTS)(TRTS)^{12} = (ST)^5R(TRTS)^{12}$$

Applications

This theory can be applied

Applications

This theory can be applied

- to solve the word problem in quotients of the free group,

Applications

This theory can be applied

- to solve the word problem in quotients of the free group,
- to solve the word problem in quotients of the modular group,

Applications

This theory can be applied

- to solve the word problem in quotients of the free group,
- to solve the word problem in quotients of the modular group,
- to decide if w can be rewritten to w' using a given rewrite system,

Applications

This theory can be applied

- to solve the word problem in quotients of the free group,
- to solve the word problem in quotients of the modular group,
- to decide if w can be rewritten to w' using a given rewrite system,
- to solve the word problem in monoids

Applications

This theory can be applied

- to solve the word problem in quotients of the free group,
- to solve the word problem in quotients of the modular group,
- to decide if w can be rewritten to w' using a given rewrite system,
- to solve the word problem in monoids
- to solve jigsaw-puzzles in which you can use arbitrarily many copies of each piece,

Applications

This theory can be applied

- to solve the word problem in quotients of the free group,
- to solve the word problem in quotients of the modular group,
- to decide if w can be rewritten to w' using a given rewrite system,
- to solve the word problem in monoids
- to solve jigsaw-puzzles in which you can use arbitrarily many copies of each piece,
- etc. ???

Applications

This theory can be applied

- to solve the word problem in quotients of the free group,
- to solve the word problem in quotients of the modular group,
- to decide if w can be rewritten to w' using a given rewrite system,
- to solve the word problem in monoids
- to solve jigsaw-puzzles in which you can use arbitrarily many copies of each piece,
- etc. ???

You only have to chose the right pongo and edge alphabet!

We first remove vertices of valency 1 and 2.

We first remove vertices of valency 1 and 2.

Theorem (Euler's formula)

In a planar embedded graph we have:

$$\#\textit{vertices} - \#\textit{edges} + \#\textit{bounded regions} = 1$$

(number of V-, E- and F-cycles, excluding the external one).

We first remove vertices of valency 1 and 2.

Theorem (Euler's formula)

In a planar embedded graph we have:

$$\#\textit{vertices} - \#\textit{edges} + \#\textit{bounded regions} = 1$$

(number of V-, E- and F-cycles, excluding the external one).

Combinatorical curvature: We endow

- each V-cycle with $+1$ unit of combinatorial curvature,
- each E-cycle with -1 unit of combinatorial curvature and
- each F-cycle with $+1$ unit of combinatorial curvature.

We first remove vertices of valency 1 and 2.

Theorem (Euler's formula)

In a planar embedded graph we have:

$$\#\textit{vertices} - \#\textit{edges} + \#\textit{bounded regions} = 1$$

(number of V-, E- and F-cycles, excluding the external one).

Combinatorical curvature: We endow

- each V-cycle with $+1$ unit of combinatorial curvature,
- each E-cycle with -1 unit of combinatorial curvature and
- each F-cycle with $+1$ unit of combinatorial curvature.

Observation

The total sum of our combinatorial curvature is always $+1$.

Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.

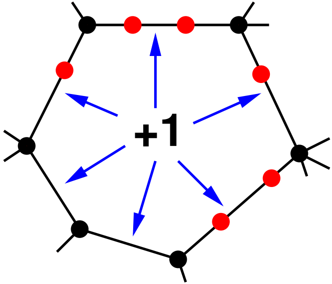
Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.
Purpose: To smear it out locally.

Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.
Purpose: To smear it out locally.

In Phase 1 we move the positive curvature to the edges:



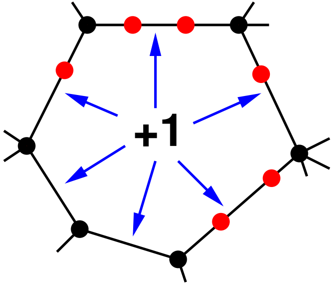
according to length

- Edges have different length (number of mini-edges).

Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.
Purpose: To smear it out locally.

In Phase 1 we move the positive curvature to the edges:



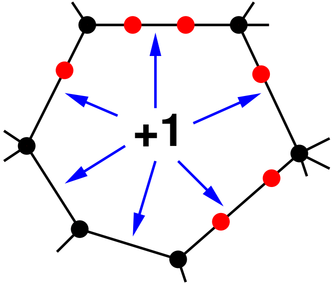
according to length

- Edges have different length (number of mini-edges).
  Both half-edges in an edge get an equal amount.

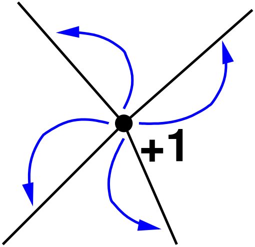
Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.
Purpose: To smear it out locally.

In Phase 1 we move the positive curvature to the edges:



according to length



evenly

- Edges have different length (number of mini-edges).
  Both half-edges in an edge get an equal amount.
- Vertices have different valency. Only outgoing half-edge receives.

All curvature is now on the half-edges.

All curvature is now on the half-edges.

Idea (Pubcrawl)

A pubcrawler crawls around (locally) from half-edge to half-edge and collects curvature. He deposits it on his orbit.

All curvature is now on the half-edges.

Idea (Pubcrawl)

A pubcrawler crawls around (locally) from half-edge to half-edge and collects curvature. He deposits it on his orbit.

- The path of the crawl is described in terms of E, F and F^{-1} steps.

All curvature is now on the half-edges.

Idea (Pubcrawl)

A pubcrawler crawls around (locally) from half-edge to half-edge and collects curvature. He deposits it on his orbit.

- The path of the crawl is described in terms of E, F and F^{-1} steps.
- We want "orbits", that is, some cyclic behaviour.

All curvature is now on the half-edges.

Idea (Pubcrawl)

A pubcrawler crawls around (locally) from half-edge to half-edge and collects curvature. He deposits it on his orbit.

- The path of the crawl is described in terms of E, F and F^{-1} steps.
- We want "orbits", that is, some cyclic behaviour.

Let $D := \{1, 2, \ldots, d\}$ and $\pi_D : \mathbb{Z} \to D$, with $z \equiv \pi_D(z) \pmod{d}$, and

All curvature is now on the half-edges.

Idea (Pubcrawl)

A pubcrawler crawls around (locally) from half-edge to half-edge and collects curvature. He deposits it on his orbit.

- The path of the crawl is described in terms of E, F and F^{-1} steps.
- We want "orbits", that is, some cyclic behaviour.

Let $D := \{1, 2, \ldots, d\}$ and $\pi_D : \mathbb{Z} \to D$, with $z \equiv \pi_D(z) \pmod{d}$, and

$$(C_1, C_2, \ldots, C_d) \in \{E, F, F^{-1}\}^d \qquad \text{(e.g. "EFEFE").}$$

All curvature is now on the half-edges.

Idea (Pubcrawl)

A pubcrawler crawls around (locally) from half-edge to half-edge and collects curvature. He deposits it on his orbit.

- The path of the crawl is described in terms of E, F and F^{-1} steps.
- We want "orbits", that is, some cyclic behaviour.

Let $D := \{1, 2, \ldots, d\}$ and $\pi_D : \mathbb{Z} \to D$, with $z \equiv \pi_D(z) \pmod{d}$, and

$$(C_1, C_2, \ldots, C_d) \in \{E, F, F^{-1}\}^d \qquad \text{(e.g. "EFEFE").}$$

Definition of the pubcrawl $(C_1, C_2, \ldots, C_d)$

Let $Y := X \times D$ and define $\Delta : Y \to Y, (x, i) \mapsto (xC_i, \pi_D(i+1))$.

All curvature is now on the half-edges.

Idea (Pubcrawl)

A pubcrawler crawls around (locally) from half-edge to half-edge and collects curvature. He deposits it on his orbit.

- The path of the crawl is described in terms of E, F and F^{-1} steps.
- We want "orbits", that is, some cyclic behaviour.

Let $D := \{1, 2, \ldots, d\}$ and $\pi_D : \mathbb{Z} \to D$, with $z \equiv \pi_D(z) \pmod{d}$, and

$$(C_1, C_2, \ldots, C_d) \in \{E, F, F^{-1}\}^d \qquad \text{(e.g. "EFEFE").}$$

Definition of the pubcrawl $(C_1, C_2, \ldots, C_d)$

Let $Y := X \times D$ and define $\Delta : Y \to Y, (x, i) \mapsto (xC_i, \pi_D(i+1))$.

$\Longrightarrow$ Δ is a permutation on Y, since E and F are permutations on X.

All curvature is now on the half-edges.

Idea (Pubcrawl)

A pubcrawler crawls around (locally) from half-edge to half-edge and collects curvature. He deposits it on his orbit.

- The path of the crawl is described in terms of E, F and F^{-1} steps.
- We want "orbits", that is, some cyclic behaviour.

Let $D := \{1, 2, \ldots, d\}$ and $\pi_D : \mathbb{Z} \to D$, with $z \equiv \pi_D(z) \pmod{d}$, and

$$(C_1, C_2, \ldots, C_d) \in \{E, F, F^{-1}\}^d \qquad \text{(e.g. "EFEFE").}$$

Definition of the pubcrawl $(C_1, C_2, \ldots, C_d)$

Let $Y := X \times D$ and define $\Delta : Y \to Y, (x, i) \mapsto (xC_i, \pi_D(i+1))$.

$\Longrightarrow$ Δ is a permutation on Y, since E and F are permutations on X.

Δ describes a step of the crawler, we sum curvature over $\langle\Delta\rangle$-orbits.

Let $L := \{1, 2, \ldots, \ell\}$ and $a_1, a_2, \ldots, a_\ell \in \mathbb{R}$ and $S := \sum_{m \in L} a_m$.

Let $L := \{1, 2, \ldots, \ell\}$ and $a_1, a_2, \ldots, a_\ell \in \mathbb{R}$ and $S := \sum_{m \in L} a_m$.
Define $\pi_L : \mathbb{Z} \to L$ such that $z \equiv \pi_L(z) \pmod{\ell}$.

Let $L := \{1, 2, \ldots, \ell\}$ and $a_1, a_2, \ldots, a_\ell \in \mathbb{R}$ and $S := \sum_{m \in L} a_m$.
Define $\pi_L : \mathbb{Z} \to L$ such that $z \equiv \pi_L(z) \pmod{\ell}$.

Lemma (Goes up and stays up)

If $S \geq 0$ then there is a $j \in L$ such that for all $i \in \mathbb{N}$ the partial sum

$$s_{j,i} := \sum_{m=0}^{i-1} a_{\pi_L(j+m)} \geq 0.$$

Let $L := \{1, 2, \ldots, \ell\}$ and $a_1, a_2, \ldots, a_\ell \in \mathbb{R}$ and $S := \sum_{m \in L} a_m$.
Define $\pi_L : \mathbb{Z} \to L$ such that $z \equiv \pi_L(z) \pmod{\ell}$.

Lemma (Goes up and stays up)

If $S \geq 0$ then there is a $j \in L$ such that for all $i \in \mathbb{N}$ the partial sum

$$s_{j,i} := \sum_{m=0}^{i-1} a_{\pi_L(j+m)} \geq 0.$$

i	1	2	3	4	5	6	7
a_i	2	-3	4	1	-5	3	2
$s_{1,i}$	2	-1	3	4	-1	2	4
$s_{6,i}$	3	5	7	4	8	9	4

Let $L := \{1, 2, \ldots, \ell\}$ and $a_1, a_2, \ldots, a_\ell \in \mathbb{R}$ and $S := \sum_{m \in L} a_m$.
Define $\pi_L : \mathbb{Z} \to L$ such that $z \equiv \pi_L(z) \pmod{\ell}$.

Lemma (Goes up and stays up)

If $S \geq 0$ then there is a $j \in L$ such that for all $i \in \mathbb{N}$ the partial sum

$$s_{j,i} := \sum_{m=0}^{i-1} a_{\pi_L(j+m)} \geq 0.$$

i	1	2	3	4	5	6	7
a_i	2	-3	4	1	-5	3	2
$s_{1,i}$	2	-1	3	4	-1	2	4
$s_{6,i}$	3	5	7	4	8	9	4

Corollary

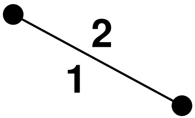
Assume that there are $k \in \mathbb{N}$ and $\varepsilon \leq 0$ such that for all $j \in L$ there is an $i \leq k$ with $s_{j,i} < \varepsilon$, then $S < \varepsilon \cdot \ell / k$.

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2			*
2	1			*
				*
				*
				*
				*

Illustration



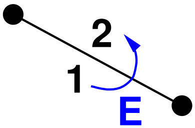
We trace the pubcrawl and disjoin cases if stuck, until:

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2			*
2	1			*
				*
				*
				*
				*

Illustration



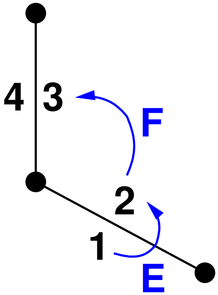
We trace the pubcrawl and disjoin cases if stuck, until:

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2			*
2	1	3		*
3	4		2	*
4	3			*
				*
				*

Illustration



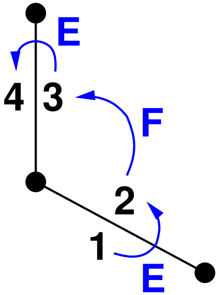
We trace the pubcrawl and disjoin cases if stuck, until:

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2			*
2	1	3		*
3	4		2	*
4	3			*
				*
				*

Illustration



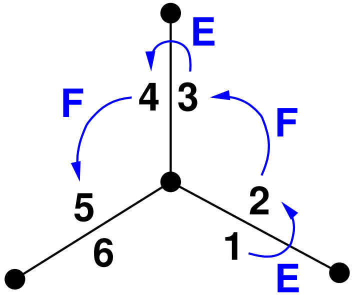
We trace the pubcrawl and disjoin cases if stuck, until:

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2			*
2	1	3		*
3	4		2	*
4	3	5		*
5	6		4	*
6	5			*

Illustration



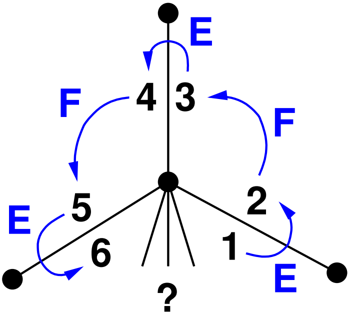
We trace the pubcrawl and disjoin cases if stuck, until:

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2			*
2	1	3		*
3	4		2	*
4	3	5		*
5	6		4	*
6	5			*

Illustration



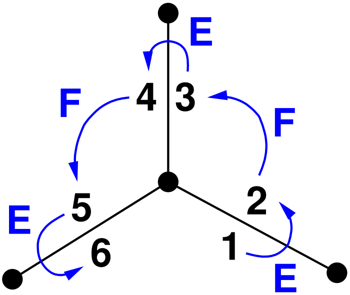
We trace the pubcrawl and disjoin cases if stuck, until:

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2		6	*
2	1	3		*
3	4		2	*
4	3	5		*
5	6		4	*
6	5	1		*

Illustration



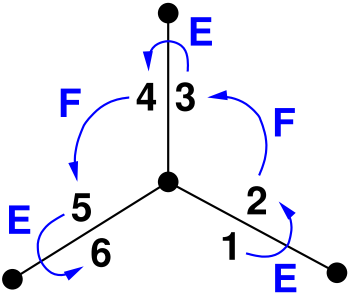
We trace the pubcrawl and disjoin cases if stuck, until:

Search for bad orbits of pubcrawl "EFEFE"

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2		6	*
2	1	3		*
3	4		2	*
4	3	5		*
5	6		4	*
6	5	1		*

Illustration



We trace the pubcrawl and disjoin cases if stuck, until:

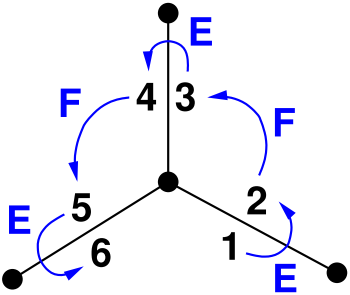
- we find a bad cycle (if we return to 1 with starting letter), or

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2		6	*
2	1	3		*
3	4		2	*
4	3	5		*
5	6		4	*
6	5	1		*

Illustration



We trace the pubcrawl and disjoin cases if stuck, until:

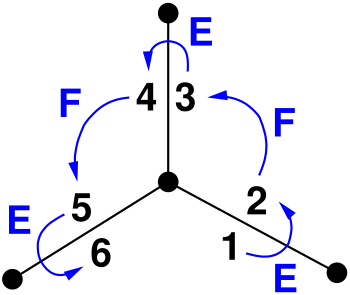
- we find a bad cycle (if we return to 1 with starting letter), or
- a partial sum is negative (keep value!), or

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2		6	*
2	1	3		*
3	4		2	*
4	3	5		*
5	6		4	*
6	5	1		*

Illustration



We trace the pubcrawl and disjoin cases if stuck, until:

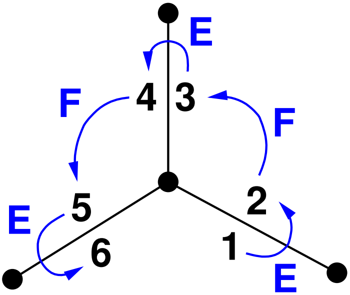
- we find a bad cycle (if we return to 1 with starting letter), or
- a partial sum is negative (keep value!), or
- we lose patience.

Search for bad orbits of pubcrawl "EFEFE"

Data structure in computer

Id	E	F	F^{-1}	Rel
1	2		6	*
2	1	3		*
3	4		2	*
4	3	5		*
5	6		4	*
6	5	1		*

Illustration



We trace the pubcrawl and disjoin cases if stuck, until:

- we find a bad cycle (if we return to 1 with starting letter), or
- a partial sum is negative (keep value!), or
- we lose patience.

Note that we use lower bounds for the vertex valencies!

An isoperimetric inequality

What have we proved?

An isoperimetric inequality

What have we proved?

If this terminates, we have

- either found a bad cycle with non-negative curvature sum, or

An isoperimetric inequality

What have we proved?

If this terminates, we have

- either found a bad cycle with non-negative curvature sum, or
- proved, that for every starting position in a crawl orbit

  $\boxed{\text{the partial sum after at most } k \text{ steps is } < \varepsilon}$

  for some global $k \in \mathbb{N}$ and some $\varepsilon < 0$.

An isoperimetric inequality

What have we proved?

If this terminates, we have

- either found a bad cycle with non-negative curvature sum, or
- proved, that for every starting position in a crawl orbit

$$\boxed{\text{the partial sum after at most } k \text{ steps is } < \varepsilon}$$

for some global $k \in \mathbb{N}$ and some $\varepsilon < 0$.

In the second case, the "Goes up and stays up" corollary tells us that

$$\boxed{\text{the sum over every interior crawl orbit of length } \ell \text{ is } < \ell \cdot \varepsilon / k < 0.}$$

An isoperimetric inequality

What have we proved?

If this terminates, we have

- either found a bad cycle with non-negative curvature sum, or
- proved, that for every starting position in a crawl orbit

  $\boxed{\text{the partial sum after at most } k \text{ steps is } < \varepsilon}$

  for some global $k \in \mathbb{N}$ and some $\varepsilon < 0$.

In the second case, the "Goes up and stays up" corollary tells us that

$\boxed{\text{the sum over every interior crawl orbit of length } \ell \text{ is } < \ell \cdot \varepsilon / k < 0.}$

Since the amount of positive curvature close to the boundary can be bounded from above by an expression in the boundary length, we get a

$\boxed{\text{linear isoperimetric inequality}}$

and thus have proved hyperbolicity. ■

An example GAP session

Outlook

We want to

- tune our program.
- investigate lots of groups.
- do algorithmic analysis to solve the word problem in practice.
- prove that for every hyperbolic group presentation there is a successful pubcrawl.
- investigate applications to monoids and rewrite systems.
- find more interesting pongos — what do they do?
- use this technology to tackle relative hyperbolicity computationally.
- write everything up and publish the theory.
- publish the software as open source.