Algorithmic Generalisations of Small Cancellation Theory

Max Neunhöffer

joint work with Jeffrey Burdges, Stephen Linton,

Richard Parker and Colva Roney-Dougal

Aberdeen, 29 November 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 $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotient.

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 $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotient.
- 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 $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotient.
- 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 $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotient.
- 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 $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotient.
- 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.
- It is not small cancellation.

Question

What can you tell me about the finitely presented group

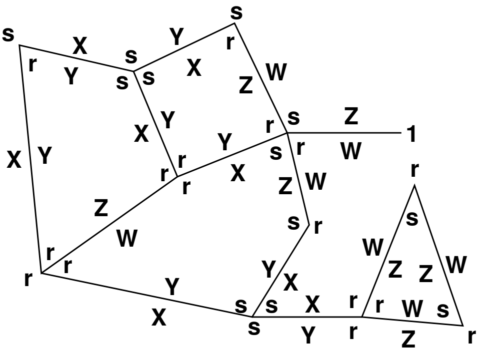
$$G := \left\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \right\rangle ?$$

(You may use a computer for this exercise!)

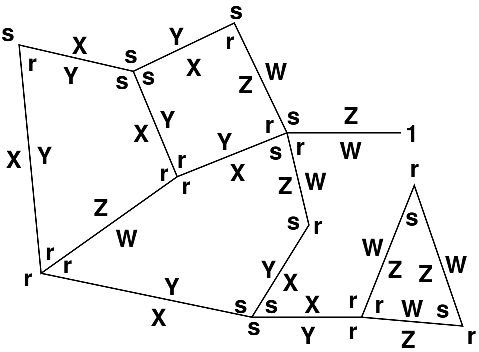
- It is a quotient of the modular group $\mathrm{PSL}_2(\mathbb{Z}) \cong \langle S, T \mid S^3, T^2 \rangle$.
- It has $3^7.\mathrm{PSL}_2(\mathbb{F}_{13})$ as quotient.
- Other finite quotients can be found (low index method).
- Eventually it turns out to be infinite (abelian invariants method).
-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

Can we solve the word problem?

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

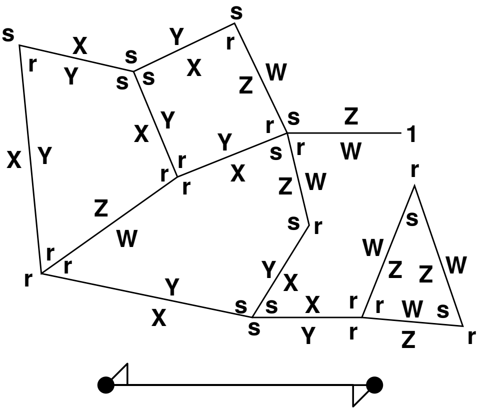


We draw connected finite graphs in the plane and label them:



Faces are oriented clockwise.

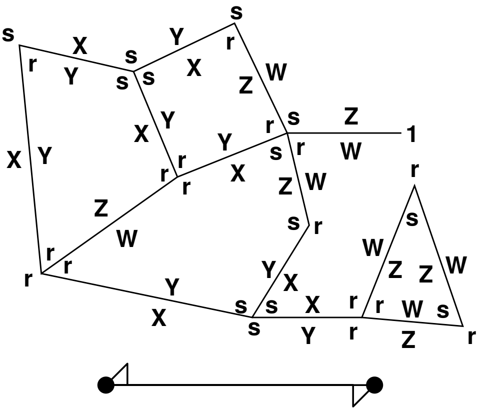
We draw connected finite graphs in the plane and label them:



Faces are oriented clockwise.

We view each edge as a pair of opposite directed edges: half-edges.

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



Faces are oriented clockwise.

We view each edge as a pair of opposite directed edges: half-edges.

Each half-edge is labelled at the start vertex and along the half-edge.

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)

Algorithmically devise 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)

Algorithmically devise 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)

Algorithmically find and prove a function $\mathcal{D} : \mathbb{N} \to \mathbb{N}$, such that for every cyclic word w of length k that is the boundary label of a diagram,

Diagram problems

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

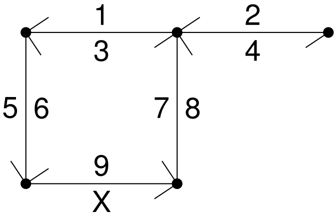
Problem (Diagram boundary problem)

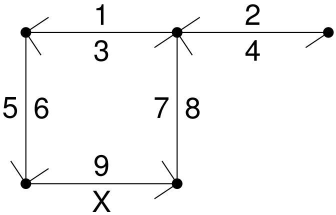
Algorithmically devise a procedure that decides for any cyclic word w, whether or not there is a diagram such that

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

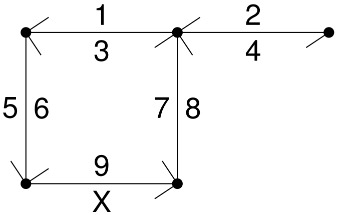
Problem (Isoperimetric inequality)

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


1
2
3
4
5
6
7
8
9
X



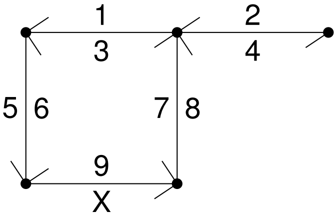
is stored as
$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$



is stored as

$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$

$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$

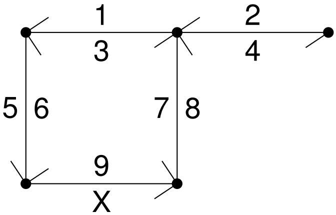


is stored as

$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$

$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$

$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9).$

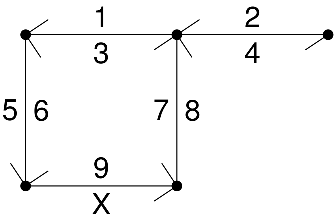


is stored as

$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$
$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$
$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9).$

Lemma (Graph equivalence)

Finite connected planar embedded graphs with $n/2$ edges



is stored as

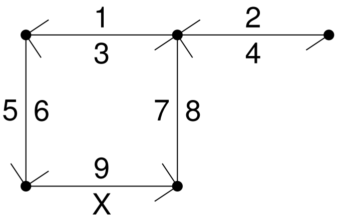
$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$

$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$

$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9).$

Lemma (Graph equivalence)

Finite connected planar embedded graphs with $n/2$ edges (up to equivalence)

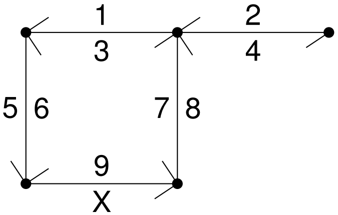


is stored as

$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$
$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$
$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9).$

Lemma (Graph equivalence)

Finite connected planar embedded graphs with $n/2$ edges (up to equivalence)
are in bijection with
ordered triples $(E, F, V) \in S_n^3$

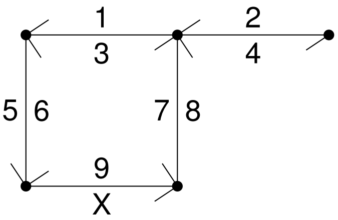


is stored as

$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$
$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$
$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9)$.

Lemma (Graph equivalence)

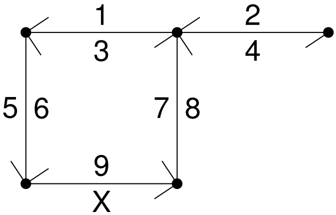
Finite connected planar embedded graphs with $n/2$ edges (up to equivalence)
are in bijection with
ordered triples $(E, F, V) \in S_n^3$ such that $EFV = 1$



is stored as
$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$
$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$
$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9)$.

Lemma (Graph equivalence)

Finite connected planar embedded graphs with $n/2$ edges (up to equivalence)
are in bijection with
ordered triples $(E, F, V) \in S_n^3$ such that $EFV = 1$, the group $\langle E, F\rangle$ is transitive



is stored as

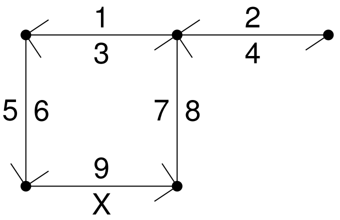
$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$
$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$
$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9).$

Lemma (Graph equivalence)

Finite connected planar embedded graphs with $n/2$ edges (up to equivalence)

are in bijection with

ordered triples $(E, F, V) \in S_n^3$ such that $EFV = 1$, the group $\langle E, F\rangle$ is transitive, $\#$cycles of E, F and V sums to $n+2$



is stored as

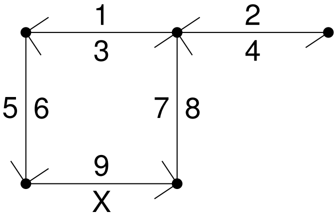
$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$
$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$
$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9)$.

Lemma (Graph equivalence)

Finite connected planar embedded graphs with $n/2$ edges (up to equivalence)

are in bijection with

ordered triples $(E, F, V) \in S_n^3$ such that $EFV = 1$, the group $\langle E, F \rangle$ is transitive, $\#$cycles of E, F and V sums to $n + 2$ and E is a fixed-point free involution



is stored as

$E := (1\,3)(2\,4)(5\,6)(7\,8)(9\,X)$
$F := (1\,5\,X\,8\,4\,2)(3\,7\,9\,6)$
$V := (1\,4\,7)(2)(3\,5)(6\,X)(8\,9)$.

Lemma (Graph equivalence)

Finite connected planar embedded graphs with $n/2$ edges (up to equivalence)

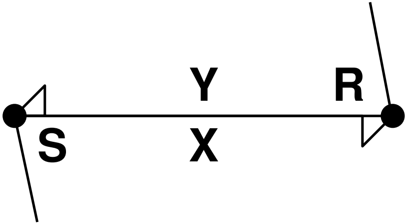
are in bijection with

ordered triples $(E, F, V) \in S_n^3$ such that $EFV = 1$, the group $\langle E, F \rangle$ is transitive, #cycles of E, F and V sums to $n + 2$ and E is a fixed-point free involution (up to S_n-conjugacy).

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.

Examples of pongos

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.
- Let $G_1, \ldots, G_k$ be groups. Let $Q := \dot{\bigcup} G_i$

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.
- Let $G_1, \ldots, G_k$ be groups. Let $Q := \dot{\bigcup} G_i$ and $Q_+ := \{1_{G_i} \mid 1 \leq i \leq k\}$.

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.
- Let $G_1, \ldots, G_k$ be groups. Let $Q := \dot{\bigcup} G_i$ and $Q_+ := \{1_{G_i} \mid 1 \leq i \leq k\}$. Elements of a single G_i multiply as before.

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.
- Let $G_1, \ldots, G_k$ be groups. Let $Q := \dot{\bigcup} G_i$ and $Q_+ := \{1_{G_i} \mid 1 \le i \le k\}$. Elements of a single G_i multiply as before. Products across factors are all 0.

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.
- Let $G_1, \ldots, G_k$ be groups. Let $Q := \dot{\bigcup} G_i$ and $Q_+ := \{1_{G_i} \mid 1 \leq i \leq k\}$. Elements of a single G_i multiply as before. Products across factors are all 0.
- Take any groupoid, undefined products are 0.

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.
- Let $G_1, \ldots, G_k$ be groups. Let $Q := \dot{\bigcup} G_i$ and $Q_+ := \{1_{G_i} \mid 1 \leq i \leq k\}$. Elements of a single G_i multiply as before. Products across factors are all 0.
- Take any groupoid, undefined products are 0.
- $K_6 := \{s, e, t, b, r, l\}$,

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.
- Let $G_1, \ldots, G_k$ be groups. Let $Q := \dot{\bigcup} G_i$ and $Q_+ := \{1_{G_i} \mid 1 \leq i \leq k\}$. Elements of a single G_i multiply as before. Products across factors are all 0.
- Take any groupoid, undefined products are 0.
- $K_6 := \{s, e, t, b, r, l\}$, $K_{6+} := \{s, e\}$,

Examples of pongos

- Let G be a group. Let $P := G$ and $P_+ := \{1\}$.
- Let $G_1, \ldots, G_k$ be groups. Let $Q := \dot{\bigcup} G_i$ and $Q_+ := \{1_{G_i} \mid 1 \leq i \leq k\}$. Elements of a single G_i multiply as before. Products across factors are all 0.
- Take any groupoid, undefined products are 0.
- $K_6 := \{s, e, t, b, r, l\}$, $K_{6+} := \{s, e\}$,

	s	e	t	b	r	l
s	0	0	s	0	0	0
e	0	0	0	e	0	0
t	s	0	t	0	0	l
b	0	e	0	b	r	0
r	0	0	r	0	0	e
l	0	0	0	l	s	0

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 (i, j), i.e. an edge, with labels X and Y is valid, if $Y = \overline{X}$.

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 (i, j), i.e. an edge, with labels X and Y is valid, if $Y = \overline{X}$.

(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 $E, F, V \in S_n$ 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)

Algorithmically devise 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)

Algorithmically devise 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)

Algorithmically find and prove a function $\mathcal{D} : \mathbb{N} \to \mathbb{N}$, such that for every cyclic word $w \in P \times A$ of length k that is the boundary label of a valid diagram,

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)

Algorithmically devise 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)

Algorithmically find and prove a function $\mathcal{D} : \mathbb{N} \to \mathbb{N}$, 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 $\mathcal{D}(k)$ 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)

Algorithmically devise 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)

Algorithmically find and prove a function $\mathcal{D} : \mathbb{N} \to \mathbb{N}$, 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 $\mathcal{D}(k)$ internal F-cycles.

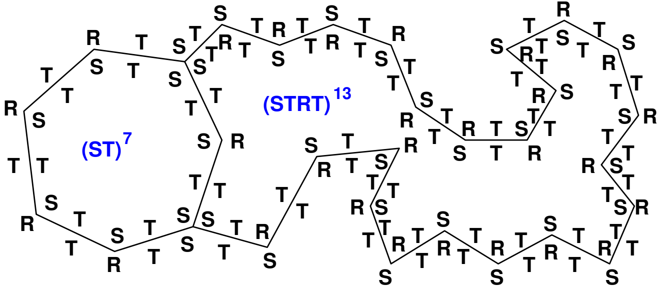
If there is a linear $\mathcal{D}$, we call (P, A, R) hyperbolic.

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

These diagrams and their two fundamental problems encode

Applications

These diagrams and their two fundamental problems encode

- the word problem in quotients of the free group,

Applications

These diagrams and their two fundamental problems encode

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

Applications

These diagrams and their two fundamental problems encode

- the word problem in quotients of the free group,
- the word problem in quotients of the modular group,
- the word problem for relative presentations (relative to one subgroup gives a Howie diagram)

Applications

These diagrams and their two fundamental problems encode

- the word problem in quotients of the free group,
- the word problem in quotients of the modular group,
- the word problem for relative presentations (relative to one subgroup gives a Howie diagram)
- the rewrite decision problem for arbitrary rewrite systems,

Applications

These diagrams and their two fundamental problems encode

- the word problem in quotients of the free group,
- the word problem in quotients of the modular group,
- the word problem for relative presentations (relative to one subgroup gives a Howie diagram)
- the rewrite decision problem for arbitrary rewrite systems,
- the word problem in monoids,

Applications

These diagrams and their two fundamental problems encode

- the word problem in quotients of the free group,
- the word problem in quotients of the modular group,
- the word problem for relative presentations (relative to one subgroup gives a Howie diagram)
- the rewrite decision problem for arbitrary rewrite systems,
- the word problem in monoids,
- jigsaw-puzzles

Applications

These diagrams and their two fundamental problems encode

- the word problem in quotients of the free group,
- the word problem in quotients of the modular group,
- the word problem for relative presentations (relative to one subgroup gives a Howie diagram)
- the rewrite decision problem for arbitrary rewrite systems,
- the word problem in monoids,
- jigsaw-puzzles in which you can use arbitrarily many copies of each piece,

Applications

These diagrams and their two fundamental problems encode

- the word problem in quotients of the free group,
- the word problem in quotients of the modular group,
- the word problem for relative presentations (relative to one subgroup gives a Howie diagram)
- the rewrite decision problem for arbitrary rewrite systems,
- the word problem in monoids,
- jigsaw-puzzles in which you can use arbitrarily many copies of each piece,
- etc. ???

Applications

These diagrams and their two fundamental problems encode

- the word problem in quotients of the free group,
- the word problem in quotients of the modular group,
- the word problem for relative presentations (relative to one subgroup gives a Howie diagram)
- the rewrite decision problem for arbitrary rewrite systems,
- the word problem in monoids,
- jigsaw-puzzles in which you can use arbitrarily many copies of each piece,
- etc. ???

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

Combinatorial Curvature

Find “pieces”, and remove vertices of valency 1 and 2:

Combinatorial Curvature

Find “pieces”, and remove vertices of valency 1 and 2:

- compute the finite list of all possible edges,

Combinatorial Curvature

Find “pieces”, and remove vertices of valency 1 and 2:

- compute the finite list of all possible edges,
- this produces a new edge alphabet,

Combinatorial Curvature

Find "pieces", and remove vertices of valency 1 and 2:

- compute the finite list of all possible edges,
- this produces a new edge alphabet, as edges now have different lengths,

Combinatorial Curvature

Find “pieces”, and remove vertices of valency 1 and 2:

- compute the finite list of all possible edges,
- this produces a new edge alphabet, as edges now have different lengths, refer to original edges as mini-edges,

Combinatorial Curvature

Find “pieces”, and remove vertices of valency 1 and 2:

- compute the finite list of all possible edges,
- this produces a new edge alphabet, as edges now have different lengths, refer to original edges as mini-edges,
- denote set of half-edges in a diagram by X, so that the diagram is now given by $E, F, V \in S_X$.

Combinatorial Curvature

Find “pieces”, and remove vertices of valency 1 and 2:

- compute the finite list of all possible edges,
- this produces a new edge alphabet, as edges now have different lengths, refer to original edges as mini-edges,
- denote set of half-edges in a diagram by X, so that the diagram is now given by $E, F, V \in S_X$.

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.

Combinatorial Curvature

Find “pieces”, and remove vertices of valency 1 and 2:

- compute the finite list of all possible edges,
- this produces a new edge alphabet, as edges now have different lengths, refer to original edges as mini-edges,
- denote set of half-edges in a diagram by X, so that the diagram is now given by $E, F, V \in S_X$.

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.

Euler’s formula/genus condition

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

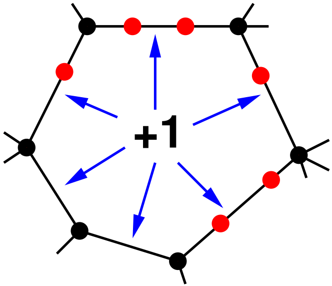
Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.

Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.

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



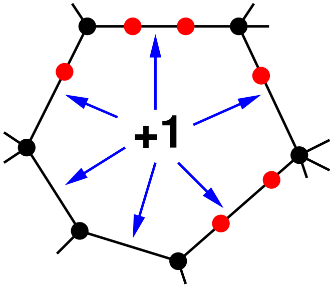
according to length

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

Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.

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



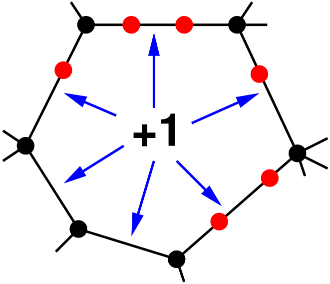
according to length

- 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.

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



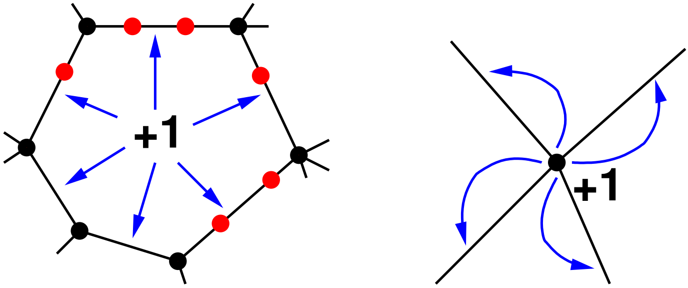
according to length

- 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.

Idea (Curvature redistribution)

We redistribute the curvature locally in a conservative way.

In Phase 1 we move the positive curvature to the half-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, the sum is still $+1$.

All curvature is now on the half-edges, the sum is still $+1$.

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, the sum is still $+1$.

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, the sum is still $+1$.

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, the sum is still $+1$.

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, the sum is still $+1$.

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

$$(C_1, C_2, \ldots, C_d) \in \{E, F, F^{-1}\}^d \qquad \text{(e.g. “EFEFE”, } d = 5).$$

All curvature is now on the half-edges, the sum is still $+1$.

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

$$(C_1, C_2, \ldots, C_d) \in \{E, F, F^{-1}\}^d \qquad \text{(e.g. "EFEFE", } d = 5\text{).}$$

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, the sum is still $+1$.

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

$$(C_1, C_2, \ldots, C_d) \in \{E, F, F^{-1}\}^d \qquad \text{(e.g. "EFEFE", } d = 5\text{).}$$

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, the sum is still $+1$.

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

$$(C_1, C_2, \ldots, C_d) \in \{E, F, F^{-1}\}^d \qquad \text{(e.g. "EFEFE", } d = 5\text{).}$$

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

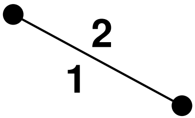
Assume that there are $k \in \mathbb{N}$ and $\varepsilon \leq 0$ such that for all $j \in L$ there is an $i \leq k$ with $s_{j,i} < \varepsilon$, then $S < \varepsilon \cdot \ell / k$.

Search for bad orbits of pubcrawl “EFEFE”

Data structure in computer

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

Illustration



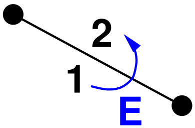
We trace the pubcrawl and disjoin cases if stuck, until:

Search for bad orbits of pubcrawl "EFEFE"

Data structure in computer

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

Illustration



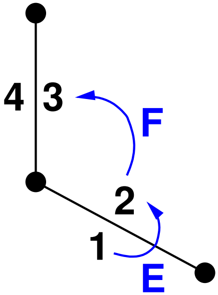
We trace the pubcrawl and disjoin cases if stuck, until:

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



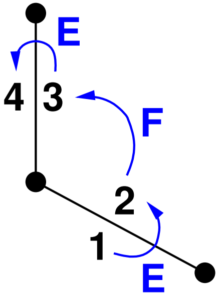
We trace the pubcrawl and disjoin cases if stuck, until:

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



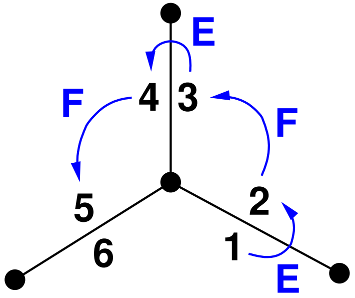
We trace the pubcrawl and disjoin cases if stuck, until:

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



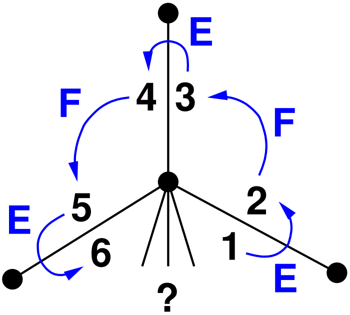
We trace the pubcrawl and disjoin cases if stuck, until:

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



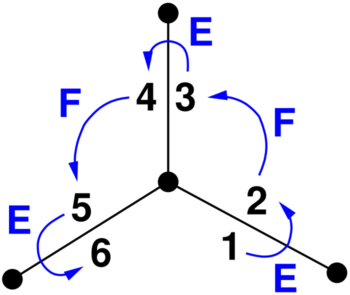
We trace the pubcrawl and disjoin cases if stuck, until:

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



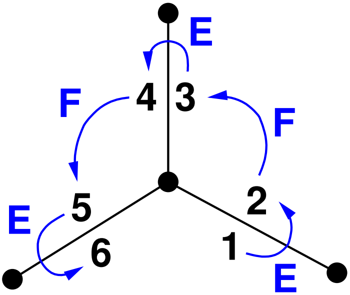
We trace the pubcrawl and disjoin cases if stuck, until:

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:

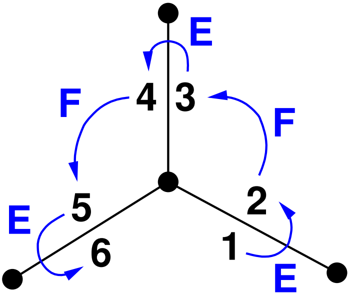
- we find a bad cycle (if we return to 1 with starting letter), or

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:

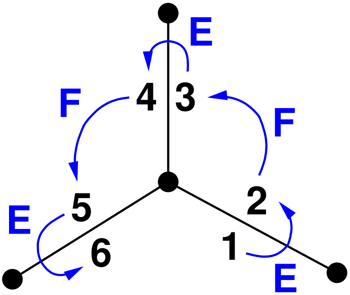
- we find a bad cycle (if we return to 1 with starting letter), or
- a partial sum is negative (keep value!), or

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:

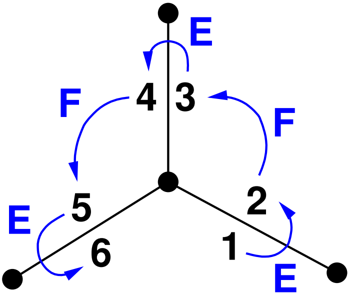
- we find a bad cycle (if we return to 1 with starting letter), or
- a partial sum is negative (keep value!), or
- we lose patience.

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!

What have we proved?

What have we proved?

If this terminates, we have

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

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 $\varepsilon < 0$.

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

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

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. ■

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. ■

e.g.: $\langle S, T \mid S^3, T^2, (ST)^7, (STS^2T)^{13} \rangle$ is hyperbolic.

Outlook and plans

We want to

- investigate more ways of redistributing curvature.
- determine whether for every presentation of a hyperbolic group there is a successful curvature-redistribution scheme
  - easy for random presentations with low Gromov density.
- sort out details for a version for relative hyperbolicity.
- investigate applications to monoids and rewrite systems.
- find more interesting pongos — what do they do?
- generalise to "flat" jigsaw puzzles.
- develop further algorithms to solve the word problem, after proving the isoperimetric inequality.
- investigate lots of examples: send us your groups!