On the isomorphism problem for virtually free groups

Armin Weiß[1]

Universität Stuttgart, Germany

Online, May 2020

[1]Based on joint work with Géraud Sénizergues and Volker Diekert

Outline

- Introduction
- Main result: complexity of the isomorphism problem for virtually free groups

Outline

- Introduction
- Main result: complexity of the isomorphism problem for virtually free groups
- Proof part 1: Graphs of groups and formal languages

Outline

- Introduction
- Main result: complexity of the isomorphism problem for virtually free groups
- Proof part 1: Graphs of groups and formal languages
- Cuts and structure trees

Isomorphism and Word problem

Isomorphism problem: Given finite presentations $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$, are the groups isomorphic?

Isomorphism and Word problem

Isomorphism problem: Given finite presentations $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$, are the groups isomorphic?

Special cases:

- virtually free presentation.
- context-free grammar for the word problem.

Isomorphism and Word problem

Isomorphism problem: Given finite presentations $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$, are the groups isomorphic?

Special cases:

- virtually free presentation.
- context-free grammar for the word problem.

Word problem for a group $G = \langle \Sigma \mid R \rangle$: Given a word $w \in \Sigma^*$, is $w =_G 1$?

$$\mathrm{WP}(G) = \{\, w \in \Sigma^* \mid w =_G 1 \,\}$$

Warm-Up: Isomorphism problem for free groups

Input: Finite presentations $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$.

Promise: $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$ are free groups.

Question: Is $\langle \Sigma \mid R \rangle \cong \langle \Sigma' \mid R' \rangle$?

Input: Finite presentations $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$.

Promise: $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$ are free groups.

Question: Is $\langle \Sigma \mid R \rangle \cong \langle \Sigma' \mid R' \rangle$?

This is decidable:

Warm-Up: Isomorphism problem for free groups

Input: Finite presentations $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$.

Promise: $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$ are free groups.

Question: Is $\langle \Sigma \mid R \rangle \cong \langle \Sigma' \mid R' \rangle$?

This is decidable:

- Add relations $ab = ba$ for $a, b \in \Sigma$ and $a^2 = 1$ for $a \in \Sigma$.

Warm-Up: Isomorphism problem for free groups

Input: Finite presentations $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$.

Promise: $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$ are free groups.

Question: Is $\langle \Sigma \mid R \rangle \cong \langle \Sigma' \mid R' \rangle$?

This is decidable:

- Add relations $ab = ba$ for $a, b \in \Sigma$ and $a^2 = 1$ for $a \in \Sigma$.
- Same for $\langle \Sigma' \mid R' \rangle$.

Warm-Up: Isomorphism problem for free groups

Input: Finite presentations $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$.

Promise: $\langle \Sigma \mid R \rangle$ and $\langle \Sigma' \mid R' \rangle$ are free groups.

Question: Is $\langle \Sigma \mid R \rangle \cong \langle \Sigma' \mid R' \rangle$?

This is decidable:

- Add relations $ab = ba$ for $a, b \in \Sigma$ and $a^2 = 1$ for $a \in \Sigma$.
- Same for $\langle \Sigma' \mid R' \rangle$.
- Use linear algebra to check isomorphism of $\mathbb{F}_2$ vector spaces.

Context-free Languages

A context-free grammar is a tuple (V, Σ, P, S) with $P \subseteq V \times (V \cup \Sigma)^*$:

Start with S and rewrite A to α if $A \to \alpha \in P$.

Context-free Languages

A context-free grammar is a tuple (V, Σ, P, S) with $P \subseteq V \times (V \cup \Sigma)^*$:

Start with S and rewrite A to α if $A \to \alpha \in P$.

- Introduced by Noam Chomsky (1950s) to describe natural language
- heavily studied in theoretical computer science
- application in programming languages/compilers etc.

Context-free Languages

A context-free grammar is a tuple (V, Σ, P, S) with $P \subseteq V \times (V \cup \Sigma)^*$:

Start with S and rewrite A to α if $A \to \alpha \in P$.

- Introduced by Noam Chomsky (1950s) to describe natural language
- heavily studied in theoretical computer science
- application in programming languages/compilers etc.

$\Sigma = \{:=, ;, \textbf{if}, \textbf{then}, \textbf{else}, \textbf{endif}, \textbf{while}, \textbf{do}, \textbf{endwhile}, (,), +, *, =, \neg, \wedge\}$,
$V = \{ A, B, C, X \}$,

$C \to X := A \mid C; C \mid \textbf{if } B \textbf{ then } C \textbf{ else } C \textbf{ endif} \mid \textbf{while } B \textbf{ do } C \textbf{ endwhile}$

$A \to X \mid (A + A) \mid (A * A)$

$B \to A = A \mid \neg B \mid B \wedge B$

Context-free Languages

A context-free grammar is a tuple (V, Σ, P, S) with $P \subseteq V \times (V \cup \Sigma)^*$:

Start with S and rewrite A to α if $A \to \alpha \in P$.

A group G is context-free iff $\mathrm{WP}(G)$ is context-free

Context-free Languages

A context-free grammar is a tuple (V, Σ, P, S) with $P \subseteq V \times (V \cup \Sigma)^*$:

Start with S and rewrite A to α if $A \to \alpha \in P$.

A group G is context-free iff $\mathrm{WP}(G)$ is context-free

Example

Free groups are context-free:

$$S \to aS\overline{a}S \mid \overline{a}SaS \mid bS\overline{b}S \mid \overline{b}SbS \mid 1$$

Context-free Languages

A context-free grammar is a tuple (V, Σ, P, S) with $P \subseteq V \times (V \cup \Sigma)^*$:

Start with S and rewrite A to α if $A \to \alpha \in P$.

A group G is context-free iff $\mathrm{WP}(G)$ is context-free

Example

Free groups are context-free:

$$S \to aS\overline{a}S \mid \overline{a}SaS \mid bS\overline{b}S \mid \overline{b}SbS \mid 1$$

Fact

If K is context-free and L is regular, then $K \cap L$ is context-free.

Context-free Languages

A context-free grammar is a tuple (V, Σ, P, S) with $P \subseteq V \times (V \cup \Sigma)^*$:

Start with S and rewrite A to α if $A \to \alpha \in P$.

A group G is context-free iff $\mathrm{WP}(G)$ is context-free

Example

Free groups are context-free:

$$S \to aS\overline{a}S \mid \overline{a}SaS \mid bS\overline{b}S \mid \overline{b}SbS \mid 1$$

Fact

If K is context-free and L is regular, then $K \cap L$ is context-free.

Example

$K = \mathrm{WP}(F_2)$

$L =$ freely reduced words $\qquad \rightsquigarrow K \cap L = \{1\}$ is context-free

Virtually free groups

Virtually free = free subgroup of finite index

Virtually free groups

Virtually free = free subgroup of finite index

Theorem (Muller, Schupp, 1983)

A group is finitely generated virtually free iff it is context-free.

Virtually free groups

Virtually free = free subgroup of finite index

Theorem (Muller, Schupp, 1983)

A group is finitely generated virtually free iff it is context-free.

$$1 \to F \to G \to Q \to 1 \qquad \text{with } F \text{ free, } Q \text{ finite}$$

Virtually free groups

Virtually free = free subgroup of finite index

Theorem (Muller, Schupp, 1983)

A group is finitely generated virtually free iff it is context-free.

$$1 \to F \to G \to Q \to 1 \qquad \text{with } F \text{ free, } Q \text{ finite}$$

Virtually free presentation:

- basis X of F,
- a system of representatives $R \subseteq G$ of $F \backslash G$
- multiplication rules: for $q \in R$, $a \in R \cup X$ there are $f \in F$, $r \in R$ with

$$qa = fr.$$

Virtually free groups

Virtually free $=$ free subgroup of finite index

Theorem (Muller, Schupp, 1983)

A group is finitely generated virtually free iff it is context-free.

$$1 \to F \to G \to Q \to 1 \qquad \text{with } F \text{ free, } Q \text{ finite}$$

Virtually free presentation:

- basis X of F,
- a system of representatives $R \subseteq G$ of $F \backslash G$
- multiplication rules: for $q \in R$, $a \in R \cup X$ there are $f \in F$, $r \in R$ with

$$qa = f\,r.$$

Example

Let $F = \mathbb{Z} = \langle x \rangle$, $Q = \mathbb{Z}/2\mathbb{Z}$, $R = \{1, a\}$

with rules $ax = xa$, $aa = 1$.

The isomorphism problem for virtually free groups

G_1

$$F_1 = \mathbb{Z} = \langle x \rangle, \qquad Q_1 = \mathbb{Z}/2\mathbb{Z}, \qquad R_1 = \{ 1, a \}$$

with rules $$ax = xa, \qquad aa = 1.$$

The isomorphism problem for virtually free groups

G_1

$$F_1 = \mathbb{Z} = \langle x \rangle, \qquad Q_1 = \mathbb{Z}/2\mathbb{Z}, \qquad R_1 = \{ 1, a \}$$

with rules $ax = xa, \qquad aa = 1.$

G_2

$$F_2 = \mathbb{Z} = \langle y \rangle, \qquad Q_2 = \mathbb{Z}/2\mathbb{Z}, \qquad R_2 = \{ 1, b \}$$

with rules $by = yb, \qquad bb = y,$

The isomorphism problem for virtually free groups

G_1

$$F_1 = \mathbb{Z} = \langle x \rangle\,, \qquad Q_1 = \mathbb{Z}/2\mathbb{Z}, \qquad R_1 = \{\,1, a\,\}$$

with rules $\qquad ax = xa, \qquad aa = 1.$

G_2

$$F_2 = \mathbb{Z} = \langle y \rangle\,, \qquad Q_2 = \mathbb{Z}/2\mathbb{Z}, \qquad R_2 = \{\,1, b\,\}$$

with rules $\qquad by = yb, \qquad bb = y,$

G_3

$$F_3 = \mathbb{Z} = \langle z \rangle\,, \qquad Q_3 = \mathbb{Z}/2\mathbb{Z}, \qquad R_3 = \{\,1, c\,\}$$

with rules $\qquad cz = zc, \qquad cc = zz.$

The isomorphism problem for virtually free groups

G_1

$$F_1 = \mathbb{Z} = \langle x \rangle\,, \qquad Q_1 = \mathbb{Z}/2\mathbb{Z}, \qquad R_1 = \{\, 1, a \,\}$$

with rules $\qquad ax = xa, \qquad aa = 1.$

G_2

$$F_2 = \mathbb{Z} = \langle y \rangle\,, \qquad Q_2 = \mathbb{Z}/2\mathbb{Z}, \qquad R_2 = \{\, 1, b \,\}$$

with rules $\qquad by = yb, \qquad bb = y,$

G_3

$$F_3 = \mathbb{Z} = \langle z \rangle\,, \qquad Q_3 = \mathbb{Z}/2\mathbb{Z}, \qquad R_3 = \{\, 1, c \,\}$$

with rules $\qquad cz = zc, \qquad cc = zz.$

Then $G_1 \cong G_3 \cong \mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ (via $z \mapsto x$, $c \mapsto ax$) and $G_2 \cong \mathbb{Z}$.

The isomorphism problem for virtually free groups

Theorem (Krstić, 1989)

The isomorphism problem for virtually free groups is decidable (input: arbitrary presentations with the promise to be virtually free).

The isomorphism problem for virtually free groups

Theorem (Krstić, 1989)

The isomorphism problem for virtually free groups is decidable (input: arbitrary presentations with the promise to be virtually free).

Theorem (Sénizergues, 1993)

The isomorphism problem with virtually free presentations as input is primitive recursive.

The isomorphism problem for virtually free groups

Theorem (Krstić, 1989)

The isomorphism problem for virtually free groups is decidable (input: arbitrary presentations with the promise to be virtually free).

Theorem (Sénizergues, 1993)

The isomorphism problem with virtually free presentations as input is primitive recursive.

Theorem (Sénizergues, 1996)

The isomorphism problem with context-free grammars as input is primitive recursive.

The isomorphism problem for virtually free groups

Theorem (Krstić, 1989)

The isomorphism problem for virtually free groups is decidable (input: arbitrary presentations with the promise to be virtually free).

Theorem (Sénizergues, 1993)

The isomorphism problem with virtually free presentations as input is primitive recursive.

Theorem (Sénizergues, 1996)

The isomorphism problem with context-free grammars as input is primitive recursive.

Theorem (Sénizergues, W. 2018)

The isomorphism problem

- *with virtually free presentations as input is in* PSPACE,
- *with context-free grammasr as input it is in* $\mathsf{SPACE}(2^{2^{\mathcal{O}(n)}})$.

Definition (Graph of Groups)

A graph of groups $\mathcal{G}$ is a connected graph $Y = (V(Y), E(Y))$ and

1. for each vertex $P \in V(Y)$, a vertex group G_P,
2. for each edge $y \in E(Y)$, an edge group $G_y \leq G_{s(y)}$.
3. for each $y \in E(Y)$, an isomorphism $f_y : G_y \to G_{\overline{y}}$ with $f_y \circ f_{\overline{y}} = \mathrm{Id}$.

Definition (Graph of Groups)

A graph of groups $\mathcal{G}$ is a connected graph $Y = (V(Y), E(Y))$ and

1. for each vertex $P \in V(Y)$, a vertex group G_P,
2. for each edge $y \in E(Y)$, an edge group $G_y \leq G_{s(y)}$.
3. for each $y \in E(Y)$, an isomorphism $f_y : G_y \to G_{\overline{y}}$ with $f_y \circ f_{\overline{y}} = \mathrm{Id}$.

Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

Definition (Graph of Groups)

A graph of groups $\mathcal{G}$ is a connected graph $Y = (V(Y), E(Y))$ and

1. for each vertex $P \in V(Y)$, a vertex group G_P,
2. for each edge $y \in E(Y)$, an edge group $G_y \leq G_{s(y)}$.
3. for each $y \in E(Y)$, an isomorphism $f_y : G_y \to G_{\overline{y}}$ with $f_y \circ f_{\overline{y}} = \mathrm{Id}$.

Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

Let $T \subseteq E(Y)$ be a spanning tree of Y

$$\pi_1(\mathcal{G}, T) = F(E(Y))$$

modulo defining relations

$$\{ x = 1 \quad | \; x \in T \}$$

Definition (Graph of Groups)

A graph of groups $\mathcal{G}$ is a connected graph $Y = (V(Y), E(Y))$ and

1. for each vertex $P \in V(Y)$, a vertex group G_P,
2. for each edge $y \in E(Y)$, an edge group $G_y \leq G_{s(y)}$.
3. for each $y \in E(Y)$, an isomorphism $f_y : G_y \to G_{\overline{y}}$ with $f_y \circ f_{\overline{y}} = \mathrm{Id}$.

Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

Let $T \subseteq E(Y)$ be a spanning tree of Y

$$\pi_1(\mathcal{G}, T) = F(E(Y)) * \mathop{\huge *}_{P \in V(Y)} G_P$$

modulo defining relations

$$\{ x = 1 \quad | \quad x \in T \quad \}$$

Definition (Graph of Groups)

A graph of groups $\mathcal{G}$ is a connected graph $Y = (V(Y), E(Y))$ and

1. for each vertex $P \in V(Y)$, a vertex group G_P,
2. for each edge $y \in E(Y)$, an edge group $G_y \leq G_{s(y)}$.
3. for each $y \in E(Y)$, an isomorphism $f_y : G_y \to G_{\overline{y}}$ with $f_y \circ f_{\overline{y}} = \mathrm{Id}$.

Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

Let $T \subseteq E(Y)$ be a spanning tree of Y

$$\pi_1(\mathcal{G}, T) = F(E(Y)) * \mathop{\ast}_{P \in V(Y)} G_P$$

modulo defining relations

$$\{\, x = 1,\ y a \overline{y} = f_y(a) \ \mid\ x \in T,\ y \in E(Y), a \in G_y \,\}$$

Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

Let $T \subseteq E(Y)$ be a spanning tree of Y

$$\pi_1(\mathcal{G}, T) = F(E(Y)) * \mathop{\ast}_{P \in V(Y)} G_P$$

modulo defining relations

$$\{\, x = 1,\ ya\overline{y} = f_y(a) \ \mid x \in T,\ y \in E(Y), a \in G_y \,\}$$

Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

Let $T \subseteq E(Y)$ be a spanning tree of Y

$$\pi_1(\mathcal{G}, T) = F(E(Y)) * \mathop{\ast}_{P \in V(Y)} G_P$$

modulo defining relations

$$\{\, x = 1,\ ya\overline{y} = f_y(a) \ \mid x \in T,\ y \in E(Y), a \in G_y \,\}$$

Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

Let $T \subseteq E(Y)$ be a spanning tree of Y

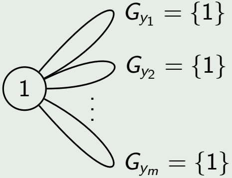
$$\pi_1(\mathcal{G}, T) = F(E(Y)) * \mathop{\ast}_{P \in V(Y)} G_P$$

modulo defining relations

$$\{\, x = 1,\ ya\overline{y} = f_y(a) \ \mid x \in T,\, y \in E(Y), a \in G_y \,\}$$

Example

F_m



Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

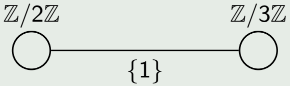
Let $T \subseteq E(Y)$ be a spanning tree of Y

$$\pi_1(\mathcal{G}, T) = F(E(Y)) * \mathop{\ast}_{P \in V(Y)} G_P$$

modulo defining relations

$$\{\, x = 1,\ ya\overline{y} = f_y(a) \ \mid x \in T,\ y \in E(Y), a \in G_y \,\}$$

Example



$$\mathrm{PSL}(2, \mathbb{Z}) \cong \mathbb{Z}/2\mathbb{Z} * \mathbb{Z}/3\mathbb{Z}$$

Definition (Fundamental group)

The fundamental group $\pi_1(\mathcal{G}, T)$ of a graph of groups $\mathcal{G}$ over Y is the fundamental group of Y + elements of the respective vertex groups.

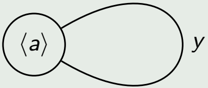
Let $T \subseteq E(Y)$ be a spanning tree of Y

$$\pi_1(\mathcal{G}, T) = F(E(Y)) * \mathop{\ast}_{P \in V(Y)} G_P$$

modulo defining relations

$$\{ x = 1, \, ya\overline{y} = f_y(a) \mid x \in T, \, y \in E(Y), a \in G_y \}$$

Example



$\mathrm{BS}_{p,q} = \langle a, y \mid ya^p y^{-1} = a^q \rangle$ edge groups $G_y = \langle a^p \rangle$ and $G_{\overline{y}} = \langle a^q \rangle$ and isomorphism $a^p \mapsto a^q$

Theorem (Karrass, Pietrowski, Solitar 73)

A f. g. group is virtually free iff it is the fundamental group of a finite graph of groups with finite vertex groups.

Theorem (Karrass, Pietrowski, Solitar 73)

A f. g. group is virtually free iff it is the fundamental group of a finite graph of groups with finite vertex groups.

Theorem (Guirardel, Levitt 07; Clay, Forester 09)

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be reduced finite graph of groups with finite vertex groups. Then $\pi_1(\mathcal{G}_1, T_1) \cong \pi_1(\mathcal{G}_2, T_2)$ iff $\mathcal{G}_1$ can be transformed into $\mathcal{G}_2$ by a sequence of slide moves.

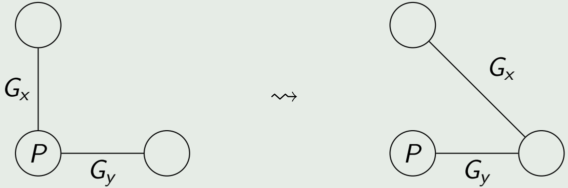
Theorem (Karrass, Pietrowski, Solitar 73)

A f. g. group is virtually free iff it is the fundamental group of a finite graph of groups with finite vertex groups.

Theorem (Guirardel, Levitt 07; Clay, Forester 09)

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be reduced finite graph of groups with finite vertex groups. Then $\pi_1(\mathcal{G}_1, T_1) \cong \pi_1(\mathcal{G}_2, T_2)$ iff $\mathcal{G}_1$ can be transformed into $\mathcal{G}_2$ by a sequence of slide moves.

Example (Slide move)



If there is some $g \in G_P$ such that $g^{-1} G_x^x g \leq G_y^y$.

Theorem (Karrass, Pietrowski, Solitar 73)

A f. g. group is virtually free iff it is the fundamental group of a finite graph of groups with finite vertex groups.

Theorem (Guirardel, Levitt 07; Clay, Forester 09)

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be reduced finite graph of groups with finite vertex groups. Then $\pi_1(\mathcal{G}_1, T_1) \cong \pi_1(\mathcal{G}_2, T_2)$ iff $\mathcal{G}_1$ can be transformed into $\mathcal{G}_2$ by a sequence of slide moves.

Corollary

It can be decided in NSPACE(n) *whether $\pi_1(\mathcal{G}_1, T_1) \cong \pi_1(\mathcal{G}_2, T_2)$ given two graph of groups $\mathcal{G}_1$ and $\mathcal{G}_2$ with finite vertex groups.*

Theorem (Karrass, Pietrowski, Solitar 73)

A f. g. group is virtually free iff it is the fundamental group of a finite graph of groups with finite vertex groups.

Theorem (Guirardel, Levitt 07; Clay, Forester 09)

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be reduced finite graph of groups with finite vertex groups. Then $\pi_1(\mathcal{G}_1, T_1) \cong \pi_1(\mathcal{G}_2, T_2)$ iff $\mathcal{G}_1$ can be transformed into $\mathcal{G}_2$ by a sequence of slide moves.

Corollary

It can be decided in NSPACE(n) whether $\pi_1(\mathcal{G}_1, T_1) \cong \pi_1(\mathcal{G}_2, T_2)$ given two graph of groups $\mathcal{G}_1$ and $\mathcal{G}_2$ with finite vertex groups.

Krstić's proof.

- For both input groups guess a GoG + an isomorphism
- verify that the guesses are correct
- check the two GoGs for isomorphism □

New approach

- Guess a GoG + an isomorphism.
- Check that the guess is correct.
- Check the two GoGs for isomorphism.

New approach

Show that the GoG and the isomorphism are “small”.

- Guess a GoG + an isomorphism.
- Check that the guess is correct.
- Check the two GoGs for isomorphism.

New approach

Show that the GoG and the isomorphism are "small".

- Guess a GoG + an isomorphism.
- Check that the guess is correct.
- Check the two GoGs for isomorphism.

Theorem (Sénizergues, W. 2018)

The following problem is in $\mathrm{NTIME}(2^{2^{\mathcal{O}(n)}})$*:*
Input: a c.f grammar for $\mathrm{WP}(G)$*,*
Compute a GoG $\mathcal{G}$ *with finite vertex groups and* $\pi_1(\mathcal{G}, T) \cong G$

New approach

Show that the GoG and the isomorphism are "small".

- Guess a GoG + an isomorphism.
- Check that the guess is correct.
- Check the two GoGs for isomorphism.

Theorem (Sénizergues, W. 2018)

The following problem is in $\mathsf{NTIME}(2^{2^{\mathcal{O}(n)}})$*:*
Input: a c.f grammar for $\mathrm{WP}(G)$*,*
Compute a GoG $\mathcal{G}$ *with finite vertex groups and* $\pi_1(\mathcal{G}, T) \cong G$

Theorem (Sénizergues, W. 2018)

The following problem is in NP*:*
Input: a group G *given as virtually free presentation,*
Compute a GoG $\mathcal{G}$ *with finite vertex groups and* $\pi_1(\mathcal{G}, T) \cong G$*.*

Small graph of groups

Main Lemma

Let G be given as context-free grammar of size $N \geq 4$ for $\mathrm{WP}(G)$. There is a graph of groups $\mathcal{G}$ over Y and an isomorphism $\varphi : \pi_1(\mathcal{G}, T) \to G$ with

1. $|V(Y)| \leq N^{50 \cdot 2^N}$,
2. $|G_P| \leq N^{50 \cdot 2^N}$ for all $P \in V(Y)$,
3. $|\varphi(a)| \leq 24 \cdot N^{175 \cdot 2^N}$ for every $a \in \Delta =$ generating set of $\pi_1(\mathcal{G}, T)$.

Small graph of groups

Main Lemma

Let G be given as context-free grammar of size $N \geq 4$ for $\mathrm{WP}(G)$. There is a graph of groups $\mathcal{G}$ over Y and an isomorphism $\varphi : \pi_1(\mathcal{G}, T) \to G$ with

1. $|V(Y)| \leq N^{50 \cdot 2^N}$,
2. $|G_P| \leq N^{50 \cdot 2^N}$ for all $P \in V(Y)$,
3. $|\varphi(a)| \leq 24 \cdot N^{175 \cdot 2^N}$ for every $a \in \Delta =$ generating set of $\pi_1(\mathcal{G}, T)$.

$\rightsquigarrow 2^{2^{\mathcal{O}(N)}}$ size

Small graph of groups

Main Lemma

Let G be given as context-free grammar of size $N \geq 4$ for $\mathrm{WP}(G)$. There is a graph of groups $\mathcal{G}$ over Y and an isomorphism $\varphi : \pi_1(\mathcal{G}, T) \to G$ with

1. $|V(Y)| \leq N^{50 \cdot 2^N}$,
2. $|G_P| \leq N^{50 \cdot 2^N}$ for all $P \in V(Y)$,
3. $|\varphi(a)| \leq 24 \cdot N^{175 \cdot 2^N}$ for every $a \in \Delta =$ generating set of $\pi_1(\mathcal{G}, T)$.

$\rightsquigarrow 2^{2^{\mathcal{O}(N)}}$ size

If G is given as virtually free presentation of size $M \geq 4$, then

1. $|V(Y)| \leq M + 1$,
2. $|G_P| \leq M$ for all $P \in V(Y)$,
3. $|\varphi(a)| \leq 12(M+1)^6$ for every $a \in \Delta$.

Small graph of groups

Main Lemma

Let G be given as context-free grammar of size $N \geq 4$ for $\mathrm{WP}(G)$. There is a graph of groups $\mathcal{G}$ over Y and an isomorphism $\varphi : \pi_1(\mathcal{G}, T) \to G$ with

1. $|V(Y)| \leq N^{50 \cdot 2^N}$,
2. $|G_P| \leq N^{50 \cdot 2^N}$ for all $P \in V(Y)$,
3. $|\varphi(a)| \leq 24 \cdot N^{175 \cdot 2^N}$ for every $a \in \Delta =$ generating set of $\pi_1(\mathcal{G}, T)$.

$\rightsquigarrow 2^{2^{\mathcal{O}(N)}}$ size

If G is given as virtually free presentation of size $M \geq 4$, then

1. $|V(Y)| \leq M + 1$,
2. $|G_P| \leq M$ for all $P \in V(Y)$,
3. $|\varphi(a)| \leq 12(M+1)^6$ for every $a \in \Delta$.

$\rightsquigarrow M^{\mathcal{O}(1)}$ size

Guess and Check

Let Σ generate G and $p : \Sigma^* \to G$ the canonical projection.

Guess and Check

Let Σ generate G and $p : \Sigma^* \to G$ the canonical projection.

$\mathrm{WP}(G) = p^{-1}(1) = \{ w \in \Sigma^* \mid w =_G 1 \}$ is context-free.

Guess and Check

Let Σ generate G and $p : \Sigma^* \to G$ the canonical projection.

$\mathrm{WP}(G) = p^{-1}(1) = \{\, w \in \Sigma^* \mid w =_G 1 \,\}$ is context-free.

1. Guess the graph of groups $\mathcal{G}$ and a hom. $\varphi : \Delta^* \to \Sigma^*$, within the bounds of the Main Lemma ($\Delta =$ generators of $\pi_1(\mathcal{G}, T)$)

Guess and Check

Let Σ generate G and $p : \Sigma^* \to G$ the canonical projection.

$\mathrm{WP}(G) = p^{-1}(1) = \{ w \in \Sigma^* \mid w =_G 1 \}$ is context-free.

1. Guess the graph of groups $\mathcal{G}$ and a hom. $\varphi : \Delta^* \to \Sigma^*$, within the bounds of the Main Lemma ($\Delta =$ generators of $\pi_1(\mathcal{G}, T)$)
2. Verify that φ induces a homomorphism $\tilde{\varphi} : \pi_1(\mathcal{G}, T) \to G$:
   - check whether $\varphi(r) =_G 1$ for all Relations $r = 1$ of $\pi_1(\mathcal{G}, T)$

Guess and Check

Let Σ generate G and $p : \Sigma^* \to G$ the canonical projection.

$\mathrm{WP}(G) = p^{-1}(1) = \{\, w \in \Sigma^* \mid w =_G 1 \,\}$ is context-free.

1. Guess the graph of groups $\mathcal{G}$ and a hom. $\varphi : \Delta^* \to \Sigma^*$, within the bounds of the Main Lemma ($\Delta =$ generators of $\pi_1(\mathcal{G}, T)$)
2. Verify that φ induces a homomorphism $\tilde{\varphi} : \pi_1(\mathcal{G}, T) \to G$:
   - check whether $\varphi(r) =_G 1$ for all Relations $r = 1$ of $\pi_1(\mathcal{G}, T)$
3. Verify that $\tilde{\varphi}$ is injective:
   - test whether $$\varphi^{-1}(\underbrace{\mathrm{WP}(G)}_{\text{context-free}}) \cap \underbrace{\{\,\text{normal forms}\,\}}_{\text{regular}} = \{\varepsilon\}$$

Guess and Check

Let Σ generate G and $p : \Sigma^* \to G$ the canonical projection.

$\mathrm{WP}(G) = p^{-1}(1) = \{\, w \in \Sigma^* \mid w =_G 1 \,\}$ is context-free.

1. Guess the graph of groups $\mathcal{G}$ and a hom. $\varphi : \Delta^* \to \Sigma^*$, within the bounds of the Main Lemma ($\Delta =$ generators of $\pi_1(\mathcal{G}, T)$)
2. Verify that φ induces a homomorphism $\tilde{\varphi} : \pi_1(\mathcal{G}, T) \to G$:
   - check whether $\varphi(r) =_G 1$ for all Relations $r = 1$ of $\pi_1(\mathcal{G}, T)$
3. Verify that $\tilde{\varphi}$ is injective:
   - test whether $$\varphi^{-1}(\underbrace{\mathrm{WP}(G)}_{\text{context-free}}) \cap \underbrace{\{\text{normal forms}\}}_{\text{regular}} = \{\varepsilon\}$$
4. Verify that φ is surjective:
   - for all $a \in \Sigma$ test if $a \in \{\, \varphi(g) \mid g \in \Delta \,\}^*$ (rational subset membership)

Guess and Check

Let Σ generate G and $p : \Sigma^* \to G$ the canonical projection.

$\mathrm{WP}(G) = p^{-1}(1) = \{\, w \in \Sigma^* \;\mid\; w =_G 1 \,\}$ is context-free.

1. Guess the graph of groups $\mathcal{G}$ and a hom. $\varphi : \Delta^* \to \Sigma^*$, within the bounds of the Main Lemma ($\Delta =$ generators of $\pi_1(\mathcal{G}, T)$)
2. Verify that φ induces a homomorphism $\tilde{\varphi} : \pi_1(\mathcal{G}, T) \to G$:
   - check whether $\varphi(r) =_G 1$ for all Relations $r = 1$ of $\pi_1(\mathcal{G}, T)$
3. Verify that $\tilde{\varphi}$ is injective:
   - test whether $\varphi^{-1}(\underbrace{\mathrm{WP}(G)}_{\text{context-free}}) \cap \underbrace{\{\,\text{normal forms}\,\}}_{\text{regular}} = \{\varepsilon\}$
4. Verify that φ is surjective:
   - for all $a \in \Sigma$ test if $a \in \{\, \varphi(g) \mid g \in \Delta \,\}^*$ (rational subset membership)

$$\iff \mathrm{WP}(G) \cap a^{-1} \cdot \{\, \varphi(g) \;\mid\; g \in \Delta \,\}^* \neq \emptyset$$

Proof of the Main Lemma

Theorem (Sénizergues 96)

Let G be a context-free group and N be the size of a c.f. grammar in Chomsky normal form for its word problem. Then

- $|H| \leq N^{12 \cdot 2^N + 10}$ *for every finite subgroup* $H \leq G$,
- *every reduced graph of groups for G has at most* $N^{12 \cdot 2^N + 11}$ *edges.*

If G is given as virtually free presentation of size $M \geq 4$, then

- $|G_P| \leq M$ for all $P \in V(Y)$,
- $|V(Y)| \leq M + 1$,

Proof of the Main Lemma

Theorem (Sénizergues 96)

Let G be a context-free group and N be the size of a c.f. grammar in Chomsky normal form for its word problem. Then

- $|H| \leq N^{12 \cdot 2^N + 10}$ *for every finite subgroup* $H \leq G$,
- *every reduced graph of groups for G has at most* $N^{12 \cdot 2^N + 11}$ *edges.*

If G is given as virtually free presentation of size $M \geq 4$, then

- $|G_P| \leq M$ for all $P \in V(Y)$,
- $|V(Y)| \leq M + 1$,

$\rightsquigarrow$ remains to bound $|\varphi(a)|$ for $a \in \Delta$

Proof of the Main Lemma

Theorem (Sénizergues 96)

Let G be a context-free group and N be the size of a c.f. grammar in Chomsky normal form for its word problem. Then

- $|H| \leq N^{12 \cdot 2^N + 10}$ *for every finite subgroup* $H \leq G$,
- *every reduced graph of groups for G has at most $N^{12 \cdot 2^N + 11}$ edges.*

If G is given as virtually free presentation of size $M \geq 4$, then

- $|G_P| \leq M$ for all $P \in V(Y)$,
- $|V(Y)| \leq M + 1$,

$\rightsquigarrow$ remains to bound $|\varphi(a)|$ for $a \in \Delta$

Idea: Follow the proof of the Muller-Schupp theorem by Diekert, W. 2013

Proof of the Main Lemma

Theorem (Sénizergues 96)

Let G be a context-free group and N be the size of a c.f. grammar in Chomsky normal form for its word problem. Then

- $|H| \leq N^{12 \cdot 2^N + 10}$ *for every finite subgroup* $H \leq G$,
- *every reduced graph of groups for G has at most* $N^{12 \cdot 2^N + 11}$ *edges.*

If G is given as virtually free presentation of size $M \geq 4$, then

- $|G_P| \leq M$ for all $P \in V(Y)$,
- $|V(Y)| \leq M + 1$,

$\rightsquigarrow$ remains to bound $|\varphi(a)|$ for $a \in \Delta$

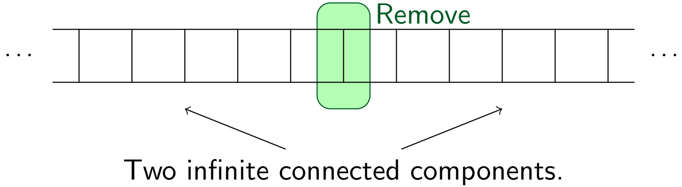
Idea: Follow the proof of the Muller-Schupp theorem by Diekert, W. 2013

$\rightsquigarrow$ Rest of the talk

Muller and Schupp's Proof (1983)

- Every infinite virtually free group has more than one end.

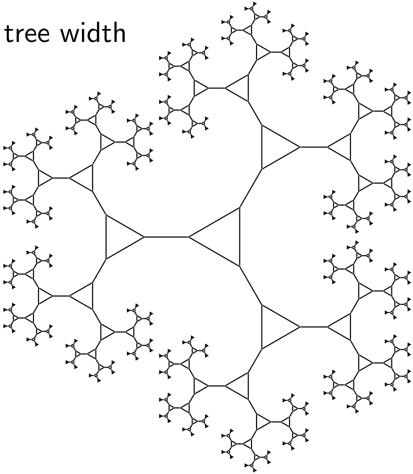
  Example: $\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$



- Stallings' Structure Theorem: every group with more than one end splits as HNN extension or amalgamated product over a finite subgroup.
- G finitely presented $\rightsquigarrow$ G is accessible: this splitting happens only finitely many times (Dunwoody 1985).

Let $\Gamma(G)$ be the Cayley graph of a context-free group G. Then:

- $\Gamma(G)$ is quasi-isometric to a tree
- $\Gamma(G)$ has finite tree width



The Cayley graph of $\mathrm{PSL}(2,\mathbb{Z}) \cong \mathbb{Z}/2\mathbb{Z} * \mathbb{Z}/3\mathbb{Z}$ has finite tree-width.

Virtually free groups are "tree-like"

Let $\Gamma(G)$ be the Cayley graph of a context-free group G. Then:

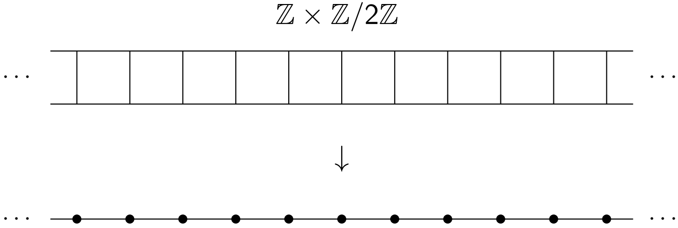
- $\Gamma(G)$ is quasi-isometric to a tree
- $\Gamma(G)$ has finite tree width

$$\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$$

Virtually free groups are "tree-like"

Let $\Gamma(G)$ be the Cayley graph of a context-free group G. Then:

- $\Gamma(G)$ is quasi-isometric to a tree
- $\Gamma(G)$ has finite tree width

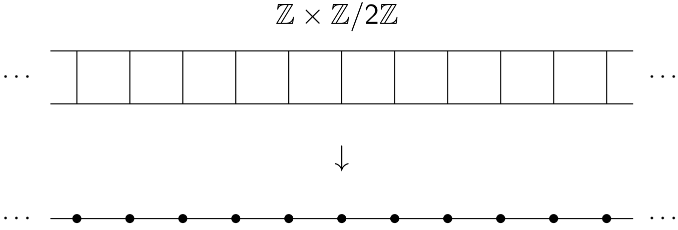


Bass-Serre theory: an action on a tree with finite stabilizers and finitely many orbit gives us the graph of groups

Virtually free groups are "tree-like"

Let $\Gamma(G)$ be the Cayley graph of a context-free group G. Then:

- $\Gamma(G)$ is quasi-isometric to a tree
- $\Gamma(G)$ has finite tree width



Bass-Serre theory: an action on a tree with finite stabilizers and finitely many orbit gives us the graph of groups

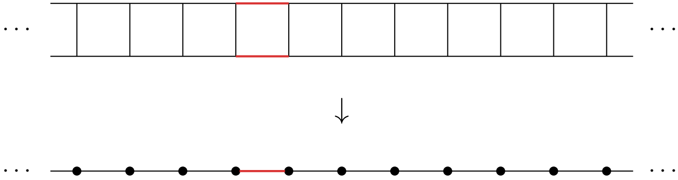
Aim: construct this tree

Virtually free groups are “tree-like”

Let $\Gamma(G)$ be the Cayley graph of a context-free group G. Then:

- $\Gamma(G)$ is quasi-isometric to a tree
- $\Gamma(G)$ has finite tree width

$\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$



Bass-Serre theory: an action on a tree with finite stabilizers and finitely many orbit gives us the graph of groups

Aim: construct this tree

Cuts

Definition

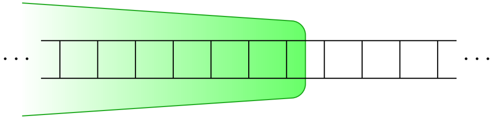
A cut is a subset $C \subseteq V(\Gamma)$ such that

- C and $\overline{C}$ are non-empty and connected,
- the boundary δC (= edges from C to $\overline{C}$) is finite.

Definition

A cut is a subset $C \subseteq V(\Gamma)$ such that

- C and $\overline{C}$ are non-empty and connected,
- the boundary δC ($=$ edges from C to $\overline{C}$) is finite.

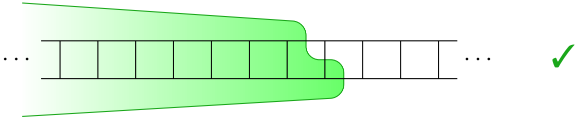
-  ✓

Cuts

Definition

A cut is a subset $C \subseteq V(\Gamma)$ such that

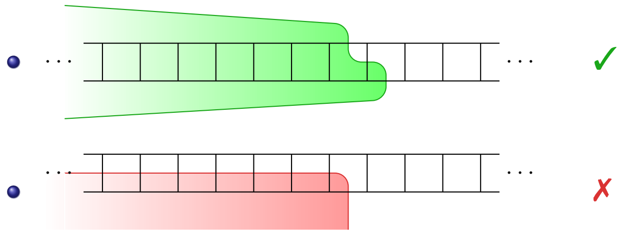
- C and $\overline{C}$ are non-empty and connected,
- the boundary δC (= edges from C to $\overline{C}$) is finite.

- 

Definition

A cut is a subset $C \subseteq V(\Gamma)$ such that

- C and $\overline{C}$ are non-empty and connected,
- the boundary δC (= edges from C to $\overline{C}$) is finite.

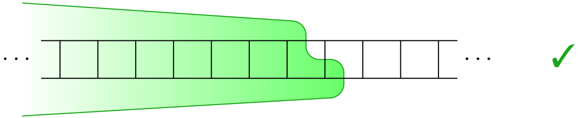


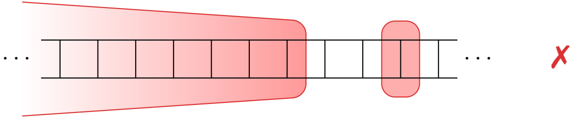
Cuts

Definition

A cut is a subset $C \subseteq V(\Gamma)$ such that

- C and $\overline{C}$ are non-empty and connected,
- the boundary δC ($=$ edges from C to $\overline{C}$) is finite.

- 

- 

Tree sets

Definition

A tree set is a set of cuts $\mathcal{C}$ such that

- $C \in \mathcal{C} \implies \overline{C} \in \mathcal{C}$,
- cuts in $\mathcal{C}$ are pairwise nested:

$$C \subseteq D \text{ or } C \subseteq \overline{D} \text{ or } D \subseteq C \text{ or } D \subseteq \overline{C} \text{ for all } C, D \in \mathcal{C},$$

- the partial order $(\mathcal{C}, \subseteq)$ is discrete:

$$\{\, E \in \mathcal{C} \mid C \subseteq E \subseteq D \,\} \text{ if finite for all } C, D \in \mathcal{C}.$$

Definition

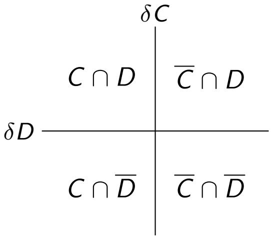
A tree set is a set of cuts $\mathcal{C}$ such that

- $C \in \mathcal{C} \implies \overline{C} \in \mathcal{C}$,
- cuts in $\mathcal{C}$ are pairwise nested:

$$C \subseteq D \text{ or } C \subseteq \overline{D} \text{ or } D \subseteq C \text{ or } D \subseteq \overline{C} \text{ for all } C, D \in \mathcal{C},$$

- the partial order $(\mathcal{C}, \subseteq)$ is discrete:

$$\{\, E \in \mathcal{C} \mid C \subseteq E \subseteq D \,\} \text{ if finite for all } C, D \in \mathcal{C}.$$

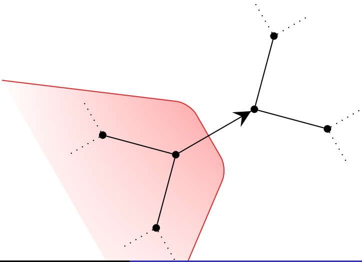


Definition

A tree set is a set of cuts $\mathcal{C}$ such that

- $C \in \mathcal{C} \implies \overline{C} \in \mathcal{C}$,
- cuts in $\mathcal{C}$ are pairwise nested:
  $$C \subseteq D \text{ or } C \subseteq \overline{D} \text{ or } D \subseteq C \text{ or } D \subseteq \overline{C} \text{ for all } C, D \in \mathcal{C},$$
- the partial order $(\mathcal{C}, \subseteq)$ is discrete:
  $$\{\, E \in \mathcal{C} \mid C \subseteq E \subseteq D \,\} \text{ if finite for all } C, D \in \mathcal{C}.$$

tree set = directed edge set of an (undirected) tree



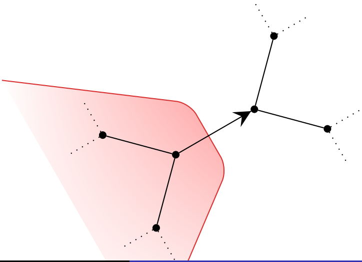
Tree sets

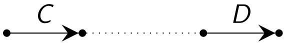
Definition

A tree set is a set of cuts $\mathcal{C}$ such that

- $C \in \mathcal{C} \implies \overline{C} \in \mathcal{C}$,
- cuts in $\mathcal{C}$ are pairwise nested:
$$C \subseteq D \text{ or } C \subseteq \overline{D} \text{ or } D \subseteq C \text{ or } D \subseteq \overline{C} \text{ for all } C, D \in \mathcal{C},$$
- the partial order $(\mathcal{C}, \subseteq)$ is discrete:
$$\{\, E \in \mathcal{C} \mid C \subseteq E \subseteq D \,\} \text{ if finite for all } C, D \in \mathcal{C}.$$

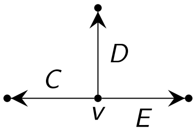
tree set = directed edge set of an (undirected) tree





means $C \subseteq D$

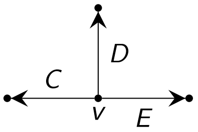
vertices = equivalence classes of edges:



$$v = \{ C, D, E \}$$

Tree sets

vertices = equivalence classes of edges:



$v = \{ C, D, E \}$

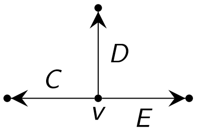
Definition

For $C, D \in \mathcal{C}$ the relation $C \sim D$ is defined as follows:

Either $C = D$,

or $\overline{C} \subsetneqq D$ and there is no $E \in \mathcal{C}$ with $\overline{C} \subsetneqq E \subsetneqq D$.

vertices = equivalence classes of edges:



$v = \{ C, D, E \}$

Definition

For C, $D \in \mathcal{C}$ the relation $C \sim D$ is defined as follows:

Either $C = D$,

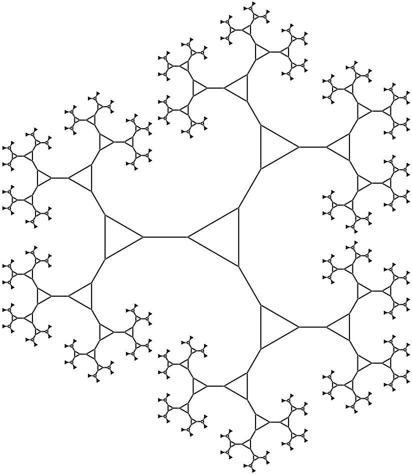
or $\overline{C} \subsetneq D$ and there is no $E \in \mathcal{C}$ with $\overline{C} \subsetneq E \subsetneq D$.

Proposition (Dunwoody, 1979)

The graph $T(\mathcal{C})$ is a tree, where

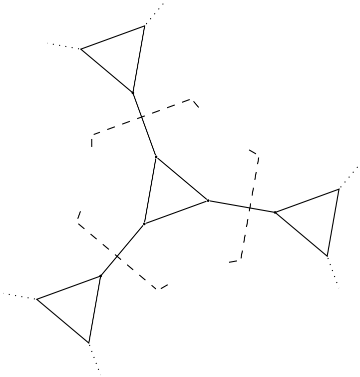
Vertices: $V(T(\mathcal{C})) = \{ [C] \mid C \in \mathcal{C} \}$,

Edges: $E(T(\mathcal{C})) = \{ \{[C], [\overline{C}]\} \mid C \in \mathcal{C} \}$.



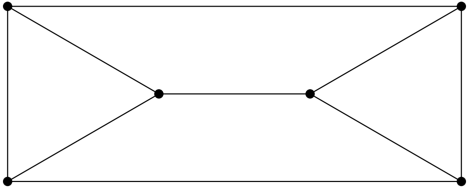
The Cayley graph of $\mathrm{PSL}(2,\mathbb{Z}) \cong \mathbb{Z}/2\mathbb{Z} * \mathbb{Z}/3\mathbb{Z}$.

Vertices in the structure tree

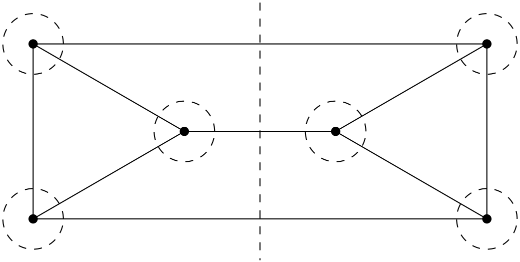


Three cuts in one equivalence class = one vertex in $T(\mathcal{C})$.

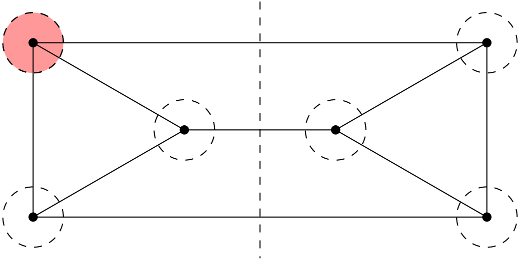
Example



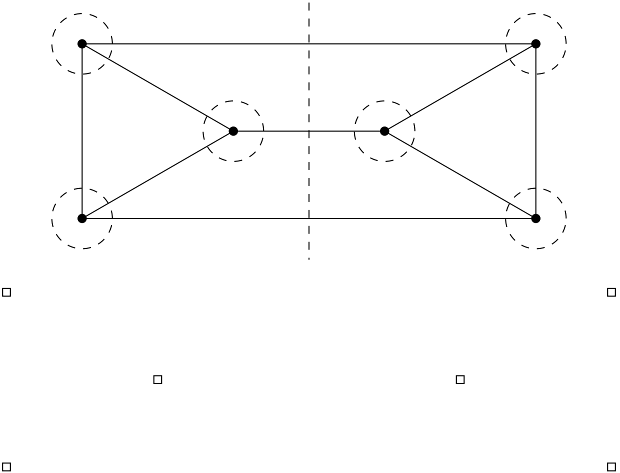
Example



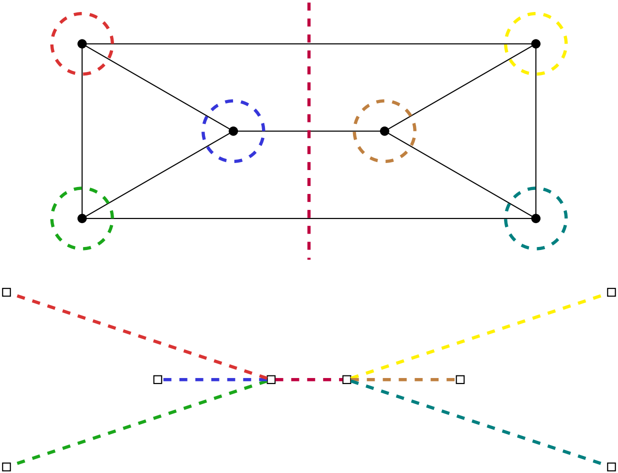
Example



Example



Example



How to find a tree set?

Aims:

- find a structure tree for an arbitrary locally finite, connected graph Γ
- if Γ is tree-like, the structure tree should be locally finite
  $\rightsquigarrow$ every bi-infinite geodesic should be split by some cut

How to find a tree set?

Aims:

- find a structure tree for an arbitrary locally finite, connected graph Γ
- if Γ is tree-like, the structure tree should be locally finite
  $\rightsquigarrow$ every bi-infinite geodesic should be split by some cut

First idea: take all cuts

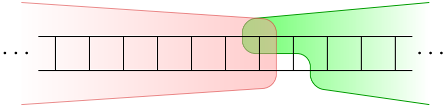
Aims:

- find a structure tree for an arbitrary locally finite, connected graph Γ
- if Γ is tree-like, the structure tree should be locally finite
  $\rightsquigarrow$ every bi-infinite geodesic should be split by some cut

First idea: take all cuts

Problems:

- not nested



How to find a tree set?

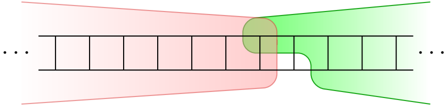
Aims:

- find a structure tree for an arbitrary locally finite, connected graph Γ
- if Γ is tree-like, the structure tree should be locally finite
  $\rightsquigarrow$ every bi-infinite geodesic should be split by some cut

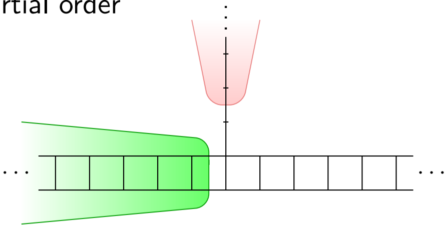
First idea: take all cuts

Problems:

- not nested



- not a discrete partial order



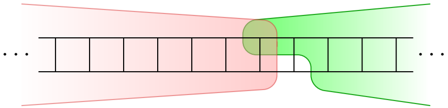
Aims:

- find a structure tree for an arbitrary locally finite, connected graph Γ
- if Γ is tree-like, the structure tree should be locally finite
  $\rightsquigarrow$ every bi-infinite geodesic should be split by some cut

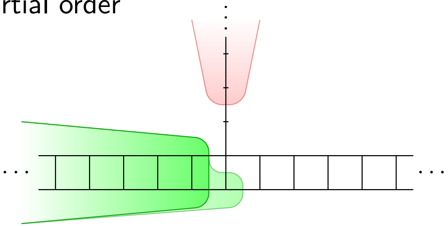
First idea: take all cuts

Problems:

- not nested



- not a discrete partial order



How to find a tree set?

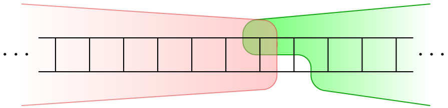
Aims:

- find a structure tree for an arbitrary locally finite, connected graph Γ
- if Γ is tree-like, the structure tree should be locally finite
  $\rightsquigarrow$ every bi-infinite geodesic should be split by some cut

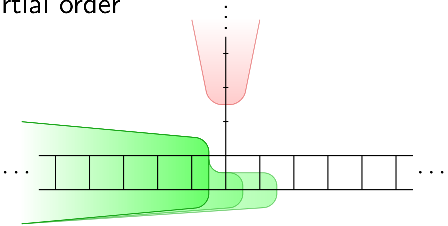
First idea: take all cuts

Problems:

- not nested



- not a discrete partial order



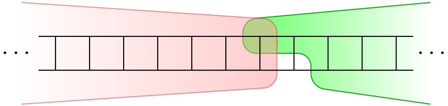
Aims:

- find a structure tree for an arbitrary locally finite, connected graph Γ
- if Γ is tree-like, the structure tree should be locally finite
  $\rightsquigarrow$ every bi-infinite geodesic should be split by some cut

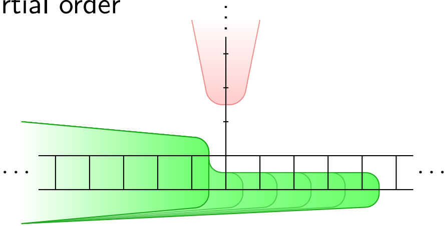
First idea: take all cuts

Problems:

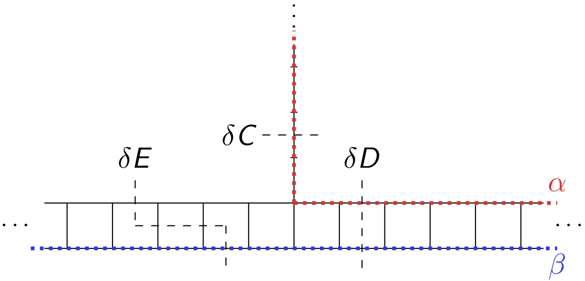
- not nested



- not a discrete partial order



Minimal cuts = cuts which are minimal splitting a bi-infinite geodesic.



- $C \in \mathcal{C}_{\min}(\alpha)$
- $D \in \mathcal{C}(\alpha) \cap \mathcal{C}_{\min}$ but $D \notin \mathcal{C}_{\min}(\alpha)$
- $E \notin \mathcal{C}_{\min}$

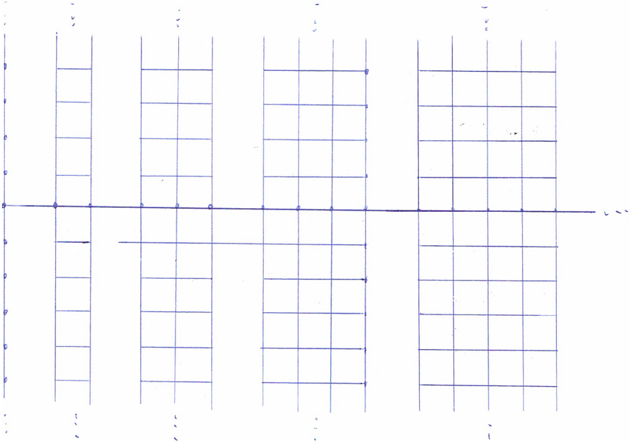
Accessible graphs

A graph Γ is accessible iff $\exists\, K \in \mathbb{N}$ with $|\delta C| \leq K$ for all $C \in \mathcal{C}_{\mathsf{min}}$.

Accessible graphs

A graph Γ is accessible iff $\exists\, K \in \mathbb{N}$ with $|\delta C| \leq K$ for all $C \in \mathcal{C}_{\min}$.

- There are non-accessible graphs:



Accessible graphs

A graph Γ is accessible iff $\exists\, K \in \mathbb{N}$ with $|\delta C| \leq K$ for all $C \in \mathcal{C}_{\mathrm{min}}$.

- There are non-accessible graphs.

Theorem (Thomassen, Woess, 1993)

G is accessible iff its Cayley graph Γ is accessible.

Theorem (Dunwoody, 1993)

There is a non-accessible group.

- $\rightsquigarrow$ There are non-accessible Cayley graphs.

But: every Cayley graph you can draw in a meaningful way is accessible.

- $\rightsquigarrow$ Tree-like Cayley graphs are accessible.

Accessible graphs

Lemma

The partial order $(\mathcal{C}_{\min}, \subseteq)$ is discrete iff Γ is accessible.

Accessible graphs

Lemma

The partial order $(\mathcal{C}_{\min}, \subseteq)$ *is discrete iff* Γ *is accessible.*

Lemma

Fix $K \in \mathbb{N}$ *and an edge* e *of* Γ. *There are only finitely many cuts* C *with* $e \in \delta C$ *and* $|\delta C| \leq K$.

Lemma

The partial order $(\mathcal{C}_{\min}, \subseteq)$ is discrete iff Γ is accessible.

Lemma

Fix $K \in \mathbb{N}$ and an edge e of Γ. There are only finitely many cuts C with $e \in \delta C$ and $|\delta C| \leq K$.

Proof (Thomassen, Woess, 1993).

$\mathcal{C}^K_{\Gamma,e}$ = cuts in Γ with $e \in \delta C$ and $|\delta C| \leq K$.

Accessible graphs

Lemma

The partial order $(\mathcal{C}_{\min}, \subseteq)$ is discrete iff Γ is accessible.

Lemma

Fix $K \in \mathbb{N}$ and an edge e of Γ. There are only finitely many cuts C with $e \in \delta C$ and $|\delta C| \leq K$.

Proof (Thomassen, Woess, 1993).

$\mathcal{C}^K_{\Gamma,e}$ = cuts in Γ with $e \in \delta C$ and $|\delta C| \leq K$.
Induction on K: the case $K = 1$ is trivial.

Lemma

The partial order $(\mathcal{C}_{\min}, \subseteq)$ is discrete iff Γ is accessible.

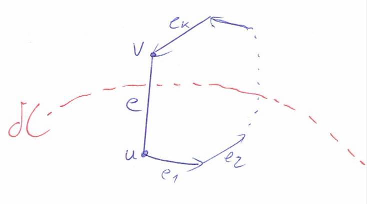
Lemma

Fix $K \in \mathbb{N}$ and an edge e of Γ. There are only finitely many cuts C with $e \in \delta C$ and $|\delta C| \leq K$.

Proof (Thomassen, Woess, 1993).

$\mathcal{C}^K_{\Gamma,e}$ = cuts in Γ with $e \in \delta C$ and $|\delta C| \leq K$.
Induction on K: the case $K = 1$ is trivial.

Let $K > 1$ and $e = \{u, v\}$ and let $e_1, \ldots, e_m$ be a path from u to v.



□

Accessible graphs

Lemma

The partial order $(\mathcal{C}_{\min}, \subseteq)$ is discrete iff Γ is accessible.

Lemma

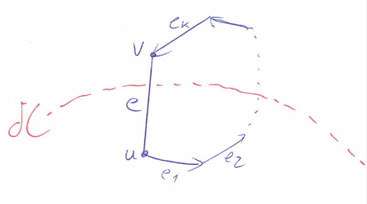
Fix $K \in \mathbb{N}$ and an edge e of Γ. There are only finitely many cuts C with $e \in \delta C$ and $|\delta C| \leq K$.

Proof (Thomassen, Woess, 1993).

$\mathcal{C}^K_{\Gamma,e}$ = cuts in Γ with $e \in \delta C$ and $|\delta C| \leq K$.
Induction on K: the case $K = 1$ is trivial.

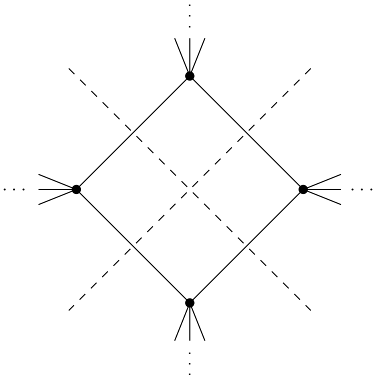
Let $K > 1$ and $e = \{u, v\}$ and let $e_1, \ldots, e_m$ be a path from u to v. Then

$$\mathcal{C}^K_{\Gamma,e} \subseteq \bigcup_{i=1}^{k} \mathcal{C}^{K-1}_{\Gamma - e, e_i}.$$

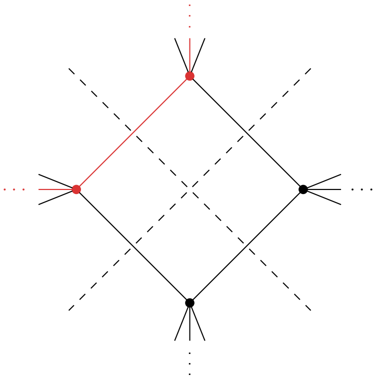


□

Minimal cuts still might not be nested:

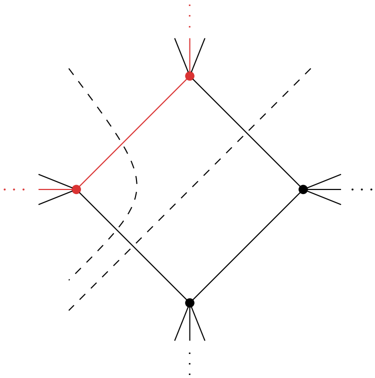


Minimal cuts still might not be nested:



Minimal cuts

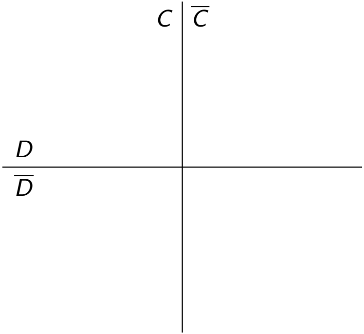
Minimal cuts still might not be nested:



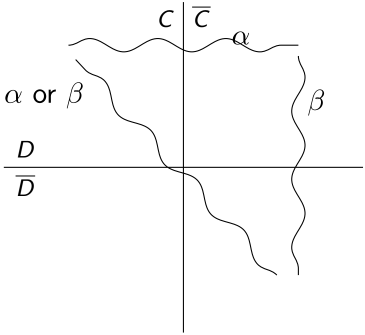
But, then we can switch to a subset.

Minimal Cuts

Let $C, D \in \mathcal{C}_{\min}$ not nested:

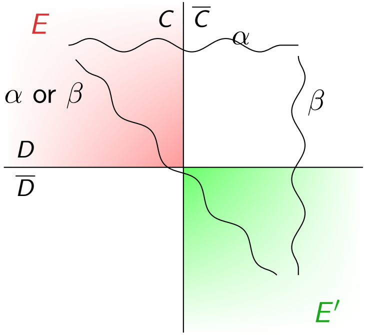


Let $C, D \in \mathcal{C}_{\min}$ not nested:



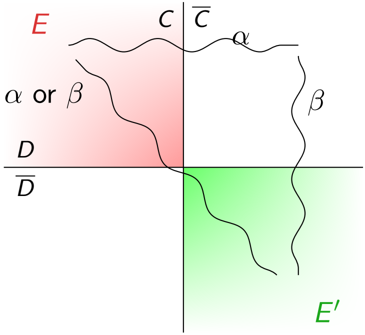
Minimal Cuts

Let $C, D \in \mathcal{C}_{\min}$ not nested:



$|\delta C| \leq |\delta E|$ and $|\delta D| \leq |\delta E'|$

Let $C, D \in \mathcal{C}_{\min}$ not nested:



$$|\delta C| \le |\delta E| \qquad \text{and} \qquad |\delta D| \le |\delta E'|$$

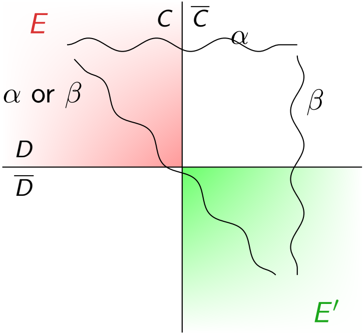
$$\delta E \cup \delta E' \subseteq \delta C \cup \delta D$$

$$\delta E \cap \delta E' \subseteq \delta C \cap \delta D$$

$$|\delta E| + |\delta E'| \le |\delta C| + |\delta D|$$

Minimal Cuts

Let $C, D \in \mathcal{C}_{\min}$ not nested:



$$|\delta C| \leq |\delta E| \qquad \text{and} \qquad |\delta D| \leq |\delta E'|$$

$$\delta E \cup \delta E' \subseteq \delta C \cup \delta D$$

$$\delta E \cap \delta E' \subseteq \delta C \cap \delta D$$

$$|\delta E| + |\delta E'| \leq |\delta C| + |\delta D|$$

$\rightsquigarrow$ take E and E' instead of C and D.

Optimal cuts

A cut C is optimal, if

- $C \in \mathcal{C}_{\min}(\alpha)$ for some bi-infinite geodesic α and
- the number of non-nested cuts is minimal among $\mathcal{C}_{\min}(\alpha)$.

Theorem (Diekert, W. 13)

For a tree-like Cayley graph Γ, the subset $\mathcal{C}_{\mathrm{opt}} \subseteq \mathcal{C}_{\mathsf{min}}$ satisfies:

- *optimal cuts form a tree set,*

Optimal cuts

A cut C is optimal, if

- $C \in \mathcal{C}_{\min}(\alpha)$ for some bi-infinite geodesic α and
- the number of non-nested cuts is minimal among $\mathcal{C}_{\min}(\alpha)$.

Theorem (Diekert, W. 13)

For a tree-like Cayley graph Γ, the subset $\mathcal{C}_{\text{opt}} \subseteq \mathcal{C}_{\mathsf{min}}$ satisfies:

- *optimal cuts form a tree set,*
- *every bi-infinite geodesic is split by an optimal cut,*

Optimal cuts

A cut C is optimal, if

- $C \in \mathcal{C}_{\min}(\alpha)$ for some bi-infinite geodesic α and
- the number of non-nested cuts is minimal among $\mathcal{C}_{\min}(\alpha)$.

Theorem (Diekert, W. 13)

For a tree-like Cayley graph Γ, the subset $\mathcal{C}_{\mathrm{opt}} \subseteq \mathcal{C}_{\mathsf{min}}$ satisfies:

- *optimal cuts form a tree set,*
- *every bi-infinite geodesic is split by an optimal cut,*
- *G acts on $\mathcal{C}_{\mathrm{opt}}$ with finitely many orbits,*

Optimal cuts

A cut C is optimal, if

- $C \in \mathcal{C}_{\min}(\alpha)$ for some bi-infinite geodesic α and
- the number of non-nested cuts is minimal among $\mathcal{C}_{\min}(\alpha)$.

Theorem (Diekert, W. 13)

For a tree-like Cayley graph Γ, the subset $\mathcal{C}_{\text{opt}} \subseteq \mathcal{C}_{\text{min}}$ satisfies:

- *optimal cuts form a tree set,*
- *every bi-infinite geodesic is split by an optimal cut,*
- *G acts on $\mathcal{C}_{\text{opt}}$ with finitely many orbits,*
- *equivalence classes $[C]$ are finite.*

Optimal cuts

A cut C is optimal, if

- $C \in \mathcal{C}_{\min}(\alpha)$ for some bi-infinite geodesic α and
- the number of non-nested cuts is minimal among $\mathcal{C}_{\min}(\alpha)$.

Theorem (Diekert, W. 13)

For a tree-like Cayley graph Γ, the subset $\mathcal{C}_{\mathrm{opt}} \subseteq \mathcal{C}_{\mathsf{min}}$ satisfies:

- *optimal cuts form a tree set,*
- *every bi-infinite geodesic is split by an optimal cut,*
- *G acts on $\mathcal{C}_{\mathrm{opt}}$ with finitely many orbits,*
- *equivalence classes $[C]$ are finite.*

$\rightsquigarrow$ *G acts on the tree $T(\mathcal{C}_{\mathrm{opt}})$ with finitely many orbits and finite vertex stabilizers.*

Optimal cuts

A cut C is optimal, if

- $C \in \mathcal{C}_{\min}(\alpha)$ for some bi-infinite geodesic α and
- the number of non-nested cuts is minimal among $\mathcal{C}_{\min}(\alpha)$.

Theorem (Diekert, W. 13)

For a tree-like Cayley graph Γ, the subset $\mathcal{C}_{\mathrm{opt}} \subseteq \mathcal{C}_{\mathsf{min}}$ satisfies:

- *optimal cuts form a tree set,*
- *every bi-infinite geodesic is split by an optimal cut,*
- *G acts on $\mathcal{C}_{\mathrm{opt}}$ with finitely many orbits,*
- *equivalence classes $[C]$ are finite.*

$\rightsquigarrow$ *G acts on the tree $T(\mathcal{C}_{\mathrm{opt}})$ with finitely many orbits and finite vertex stabilizers.*

$G \backslash T(\mathcal{C}_{\mathrm{opt}})$ is the desired graph of groups (resp. a reduced subset of $\mathcal{C}_{\mathrm{opt}}$).

Back to the isomorphism problem: Roadmap for Proving the Main Lemma

Aim: find "small" isomorphism φ

- $\varphi(g) = g$ for $g \in G_P = \mathrm{Stab}(P)$
- $\varphi(y)$ conjugates elements of G_P into the stabilizer of some nearby $Q \in V(T(\mathcal{C}_{\mathrm{opt}}))$

Back to the isomorphism problem: Roadmap for Proving the Main Lemma

Aim: find "small" isomorphism φ

- $\varphi(g) = g$ for $g \in G_P = \mathrm{Stab}(P)$
- $\varphi(y)$ conjugates elements of G_P into the stabilizer of some nearby $Q \in V(T(\mathcal{C}_{\mathrm{opt}}))$

Show:

- boundaries of minimal cuts are small
- equivalent cuts are not far apart

$\rightsquigarrow$ find representatives for $T(\mathcal{C}_{\mathrm{opt}})$ within $B(2^{2^{\mathcal{O}(N)}})$ (resp. $B(N^{\mathcal{O}(1)})$)

Summary

- Isomorphism test for fundamental groups of GoGs with finite vertex groups is in NSPACE(n).

Summary

- Isomorphism test for fundamental groups of GoGs with finite vertex groups is in NSPACE(n).
- Given a context-free grammar, we can compute the GoG in NTIME($2^{2^{\mathcal{O}(n)}}$).

Summary

- Isomorphism test for fundamental groups of GoGs with finite vertex groups is in NSPACE(n).
- Given a context-free grammar, we can compute the GoG in NTIME($2^{2^{\mathcal{O}(n)}}$).
- Given a virtually free presentation, we can compute the GoG in NP.

Summary

- Isomorphism test for fundamental groups of GoGs with finite vertex groups is in NSPACE(n).
- Given a context-free grammar, we can compute the GoG in NTIME($2^{2^{\mathcal{O}(n)}}$).
- Given a virtually free presentation, we can compute the GoG in NP.
- Bounds via cuts in the structure tree.

Summary

- Isomorphism test for fundamental groups of GoGs with finite vertex groups is in NSPACE(n).
- Given a context-free grammar, we can compute the GoG in NTIME($2^{2^{\mathcal{O}(n)}}$).
- Given a virtually free presentation, we can compute the GoG in NP.
- Bounds via cuts in the structure tree.

Open Questions:

- Precise complexity bounds.

Summary

- Isomorphism test for fundamental groups of GoGs with finite vertex groups is in NSPACE(n).
- Given a context-free grammar, we can compute the GoG in NTIME($2^{2^{\mathcal{O}(n)}}$).
- Given a virtually free presentation, we can compute the GoG in NP.
- Bounds via cuts in the structure tree.

Open Questions:

- Precise complexity bounds.
- What about other groups (e. g. hyperbolic)?

Summary

- Isomorphism test for fundamental groups of GoGs with finite vertex groups is in NSPACE(n).
- Given a context-free grammar, we can compute the GoG in NTIME($2^{2^{\mathcal{O}(n)}}$).
- Given a virtually free presentation, we can compute the GoG in NP.
- Bounds via cuts in the structure tree.

Open Questions:

- Precise complexity bounds.
- What about other groups (e. g. hyperbolic)?
- Algorithm for finding structure trees for finite graphs?

Summary

- Isomorphism test for fundamental groups of GoGs with finite vertex groups is in NSPACE(n).
- Given a context-free grammar, we can compute the GoG in NTIME($2^{2^{\mathcal{O}(n)}}$).
- Given a virtually free presentation, we can compute the GoG in NP.
- Bounds via cuts in the structure tree.

Open Questions:

- Precise complexity bounds.
- What about other groups (e. g. hyperbolic)?
- Algorithm for finding structure trees for finite graphs?

Thank you!