

Subgroups of free groups: finite index extensions

Pascal Weil

LaBRI, CNRS et Université de Bordeaux

FREC, LaBRI, June 2012

Free groups and subgroups

- Representation of subgroups
- Rank, index and conjugates
- Intersection of subgroups
- Malnormal and pure subgroups

Subgroups of subgroups

- The maximal finite index extension of H
- Computing the fi extensions of H
- A fast algorithm to compute H_{fi}

Computing the malnormal closure of H

- An automata-theoretic characterization
- A polynomial algorithm

The free group on A , $F(A)$

- ▶ the “simplest” group built from the symbols in set A : start with strings (*words*) on alphabet $A \cup \bar{A}$, and let $a\bar{a} = \bar{a}a = 1$

The free group on A , $F(A)$

- ▶ the “simplest” group built from the symbols in set A : start with strings (*words*) on alphabet $A \cup \bar{A}$, and let $a\bar{a} = \bar{a}a = 1$
- ▶ *i.e.*, perform reductions: $ua\bar{a}v \rightarrow uv$, $u\bar{a}av \rightarrow uv$. For instance $a^3\bar{a}\bar{b}ab^2\bar{b}\bar{a}a \xrightarrow{*} a^2\bar{b}ab$

The free group on A , $F(A)$

- ▶ the “simplest” group built from the symbols in set A : start with strings (*words*) on alphabet $A \cup \bar{A}$, and let $a\bar{a} = \bar{a}a = 1$
- ▶ *i.e.*, perform reductions: $ua\bar{a}v \rightarrow uv$, $u\bar{a}av \rightarrow uv$. For instance $a^3\bar{a}\bar{b}ab^2\bar{b}\bar{a}a \xrightarrow{*} a^2\bar{b}ab$
- ▶ this rewriting system is confluent, it terminates: it yields a unique representation by means of *reduced* words

The free group on A , $F(A)$

- ▶ the “simplest” group built from the symbols in set A : start with strings (*words*) on alphabet $A \cup \bar{A}$, and let $a\bar{a} = \bar{a}a = 1$
- ▶ *i.e.*, perform reductions: $ua\bar{a}v \rightarrow uv$, $u\bar{a}av \rightarrow uv$. For instance $a^3\bar{a}\bar{b}ab^2\bar{b}\bar{a}a \xrightarrow{*} a^2\bar{b}ab$
- ▶ this rewriting system is confluent, it terminates: it yields a unique representation by means of *reduced* words
- ▶ $F(A)$ and $F(B)$ are isomorphic iff A and B have the same cardinality : the notions of *rank*, *basis* make sense for free groups

The free group on A , $F(A)$

- ▶ the “simplest” group built from the symbols in set A : start with strings (*words*) on alphabet $A \cup \bar{A}$, and let $a\bar{a} = \bar{a}a = 1$
- ▶ *i.e.*, perform reductions: $ua\bar{a}v \rightarrow uv$, $u\bar{a}av \rightarrow uv$. For instance $a^3\bar{a}\bar{b}ab^2\bar{b}\bar{a}a \xrightarrow{*} a^2\bar{b}ab$
- ▶ this rewriting system is confluent, it terminates: it yields a unique representation by means of *reduced* words
- ▶ $F(A)$ and $F(B)$ are isomorphic iff A and B have the same cardinality : the notions of *rank*, *basis* make sense for free groups
- ▶ The subgroups of a free group are free. . .

The free group on A , $F(A)$

- ▶ the “simplest” group built from the symbols in set A : start with strings (*words*) on alphabet $A \cup \bar{A}$, and let $a\bar{a} = \bar{a}a = 1$
- ▶ *i.e.*, perform reductions: $ua\bar{a}v \rightarrow uv$, $u\bar{a}av \rightarrow uv$. For instance $a^3\bar{a}\bar{b}ab^2\bar{b}\bar{a}a \xrightarrow{*} a^2\bar{b}ab$
- ▶ this rewriting system is confluent, it terminates: it yields a unique representation by means of *reduced* words
- ▶ $F(A)$ and $F(B)$ are isomorphic iff A and B have the same cardinality : the notions of *rank*, *basis* make sense for free groups
- ▶ The subgroups of a free group are free. . .
- ▶ . . . but a subgroup may have a larger rank than the group in which it sits!

The free group on A , $F(A)$

- ▶ the “simplest” group built from the symbols in set A : start with strings (*words*) on alphabet $A \cup \bar{A}$, and let $a\bar{a} = \bar{a}a = 1$
- ▶ *i.e.*, perform reductions: $ua\bar{a}v \rightarrow uv$, $u\bar{a}av \rightarrow uv$. For instance $a^3\bar{a}\bar{b}ab^2\bar{b}\bar{a}a \xrightarrow{*} a^2\bar{b}ab$
- ▶ this rewriting system is confluent, it terminates: it yields a unique representation by means of *reduced* words
- ▶ $F(A)$ and $F(B)$ are isomorphic iff A and B have the same cardinality : the notions of *rank*, *basis* make sense for free groups
- ▶ The subgroups of a free group are free. . .
- ▶ . . . but a subgroup may have a larger rank than the group in which it sits!
- ▶ Every subset of $\{a^n b a^{-n} \mid n \geq 0\}$ is a basis of a subgroup of $F(a, b)$.

In lieu of motivation

- ▶ Algorithmic problems on elements (reduced words) and on finitely generated subgroups of free groups: compute the rank, a basis, the index, conjugacy problems, etc.

In lieu of motivation

- ▶ Algorithmic problems on elements (reduced words) and on finitely generated subgroups of free groups: compute the rank, a basis, the index, conjugacy problems, etc.
- ▶ The general idea for subgroups: to represent the finitely generated subgroups of a free group by combinatorial objects (certain kinds of *automata*)

In lieu of motivation

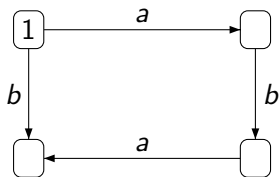
- ▶ Algorithmic problems on elements (reduced words) and on finitely generated subgroups of free groups: compute the rank, a basis, the index, conjugacy problems, etc.
- ▶ The general idea for subgroups: to represent the finitely generated subgroups of a free group by combinatorial objects (certain kinds of *automata*)
- ▶ results in effective (often efficient) computations on subgroups of free groups

Representation of subgroups

▶ $H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$

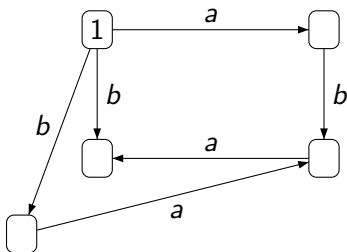
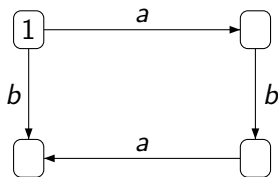
Representation of subgroups

► $H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$



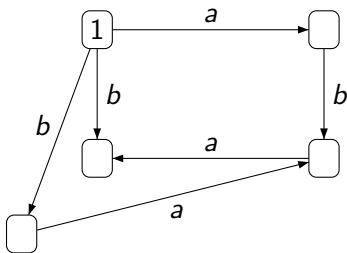
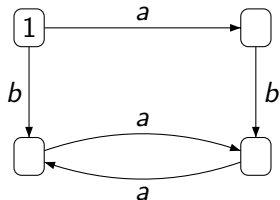
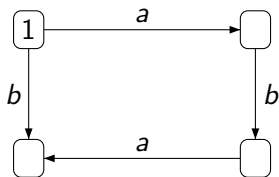
Representation of subgroups

► $H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$



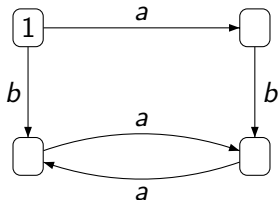
Representation of subgroups

► $H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$



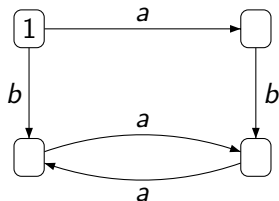
Representation of subgroups

- ▶ $H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$
- ▶ reduced automaton: finite, connected, deterministic and co-deterministic, with a distinguished state 1; every vertex $v \neq 1$ has valency at least 2



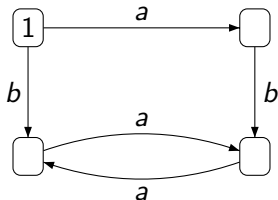
Representation of subgroups

- ▶ $H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$
- ▶ reduced automaton: finite, connected, deterministic and co-deterministic, with a distinguished state 1; every vertex $v \neq 1$ has valency at least 2
- ▶ $H =$ all reduced words that read from 1 to 1



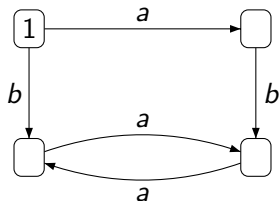
Representation of subgroups

- ▶ $H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$
- ▶ this automaton is denoted by $\Gamma(H)$, or $\Gamma_A(H)$. It characterizes H , not the generating set

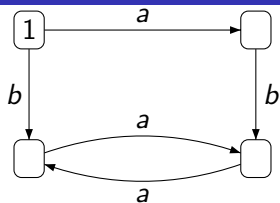


Representation of subgroups

- ▶ $H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$
- ▶ this automaton is denoted by $\Gamma(H)$, or $\Gamma_A(H)$. It characterizes H , not the generating set
- ▶ $\Gamma(H)$ is computable in $O(n \log^* n)$

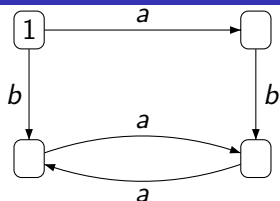


Basic invariants of subgroups



$$H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$$

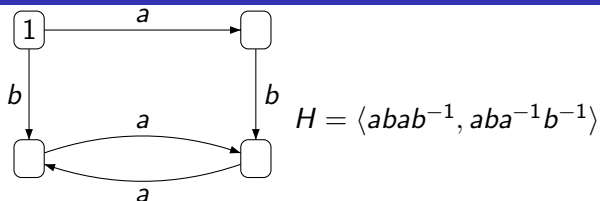
Basic invariants of subgroups



$$H = \langle abab^{-1}, aba^{-1}b^{-1}, aba^3b^{-1} \rangle$$

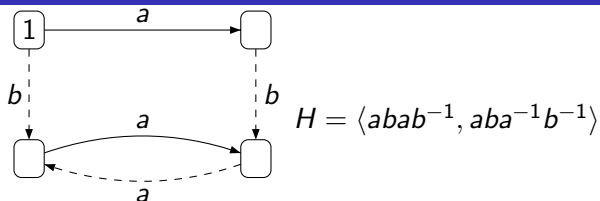
- ▶ aba^3b^{-1} is a product of the other generators

Basic invariants of subgroups



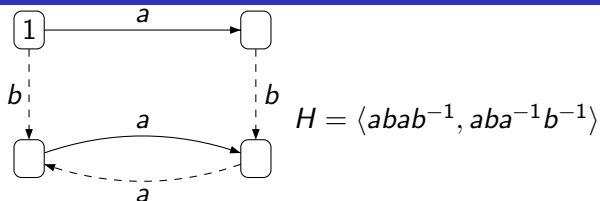
- ▶ aba^3b^{-1} is a product of the other generators
- ▶ an efficient solution of the (generalized) membership problem

Basic invariants of subgroups



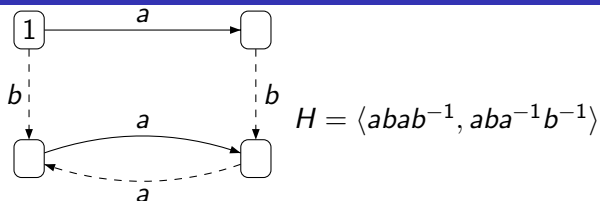
- ▶ aba^3b^{-1} is a product of the other generators
- ▶ an efficient solution of the (generalized) membership problem
- ▶ the rank is the number of “independent” loops, which can be read on $\Gamma(H)$: choose a spanning tree T ; then each A -labeled edge not in T yields a basis element

Basic invariants of subgroups



- ▶ aba^3b^{-1} is a product of the other generators
- ▶ an efficient solution of the (generalized) membership problem
- ▶ the rank is the number of “independent” loops, which can be read on $\Gamma(H)$: choose a spanning tree T ; then each A -labeled edge not in T yields a basis element
- ▶ H has rank 2, and for the dotted spanning tree, we get basis $abab^{-1}, ba^2b^{-1}$

Basic invariants of subgroups

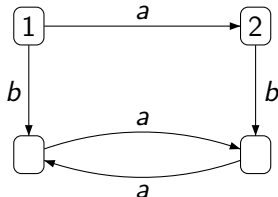


- ▶ aba^3b^{-1} is a product of the other generators
- ▶ an efficient solution of the (generalized) membership problem
- ▶ the rank is the number of “independent” loops, which can be read on $\Gamma(H)$: choose a spanning tree T ; then each A -labeled edge not in T yields a basis element
- ▶ H has rank 2, and for the dotted spanning tree, we get basis $abab^{-1}, ba^2b^{-1}$
- ▶ $\text{rank}(H) = |E(H)| - |V(H)| + 1$.

Finite-index subgroups

Each vertex is an H -coset:

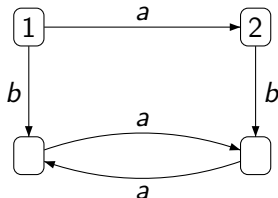
vertex 2 is Ha , or also $Hbab^{-1}$



Finite-index subgroups

Each vertex is an H -coset:

vertex 2 is Ha , or also $Hbab^{-1}$

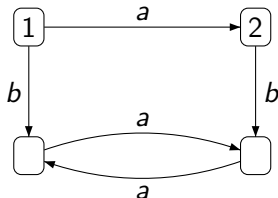


Recall: $F(A)$ is partitioned by the H -cosets Hg ($g \in F(A)$), the index of H is the number of these cosets.

Finite-index subgroups

Each vertex is an H -coset:

vertex 2 is Ha , or also $Hbab^{-1}$



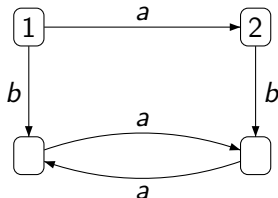
- ▶ In general, not all H -cosets occur as vertices of $\Gamma(H)$. But if $\Gamma(H)$ is a permutation graph, they do, and H has finite index, equal to the number of vertices $|V(H)|$. That's an iff.

Recall: $F(A)$ is partitioned by the H -cosets Hg ($g \in F(A)$), the index of H is the number of these cosets.

Finite-index subgroups

Each vertex is an H -coset:

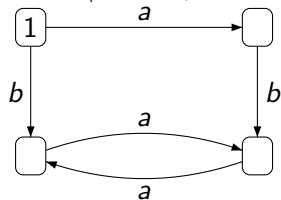
vertex 2 is Ha , or also $Hbab^{-1}$



- ▶ In general, not all H -cosets occur as vertices of $\Gamma(H)$. But if $\Gamma(H)$ is a permutation graph, they do, and H has finite index, equal to the number of vertices $|V(H)|$. That's an iff.
- ▶ this yields a proof of the Nielsen-Schreier formula for finite index subgroups: if H has finite index $i(H)$, then $\text{rank}(H) - 1 = i(H)(\text{rank}(F(A)) - 1)$

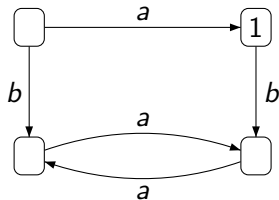
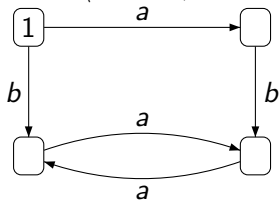
Computing the conjugates of H

► $H = \langle abab^{-1}, aba^{-1}b^{-1} \rangle$



Computing the conjugates of H

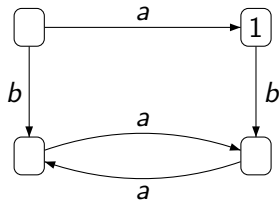
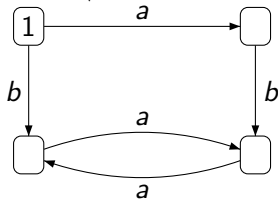
- $H = \langle abab^{-1}, aba^{-1}b^{-1} \rangle$



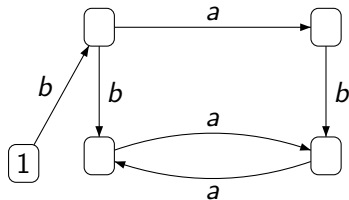
$a^{-1}Ha$

Computing the conjugates of H

- $H = \langle abab^{-1}, aba^{-1}b^{-1} \rangle$



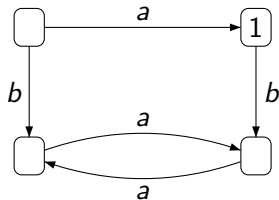
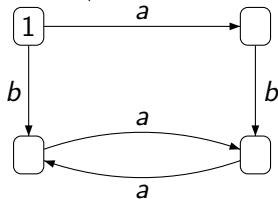
$a^{-1}Ha$



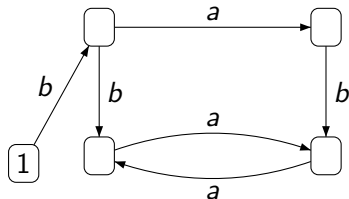
bHb^{-1}

Computing the conjugates of H

- ▶ $H = \langle abab^{-1}, aba^{-1}b^{-1} \rangle$



$a^{-1}Ha$

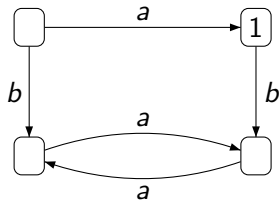
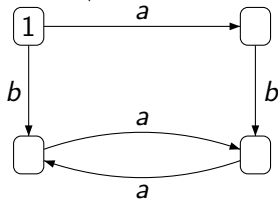


bHb^{-1}

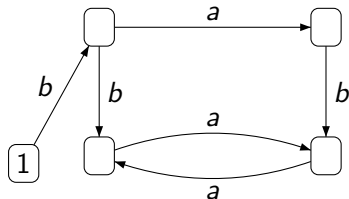
- ▶ an algorithm to solve the conjugacy problem

Computing the conjugates of H

- ▶ $H = \langle abab^{-1}, aba^{-1}b^{-1} \rangle$



$a^{-1}Ha$



bHb^{-1}

- ▶ the conjugacy class of H is characterized by the cyclic core of $\Gamma(H)$

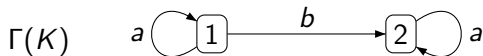
Computing the intersection of two subgroups

$$K = \langle a, bab^{-1} \rangle$$

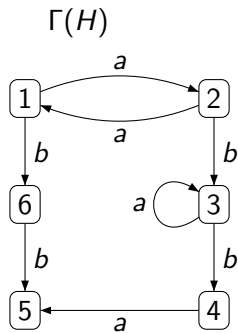
$$H = \langle a^2, abab^{-1}a^{-1}, ab^2ab^{-2} \rangle$$

$H \cap K = ?$ It contains a^2 ; what else?

Computing the intersection of two subgroups

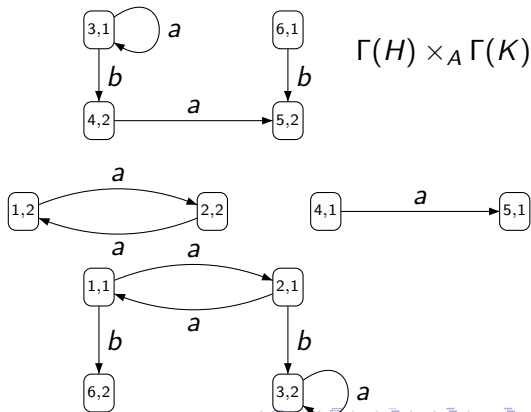
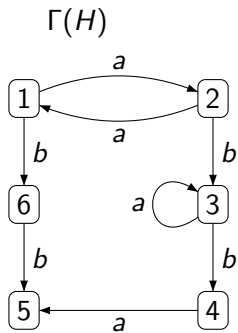
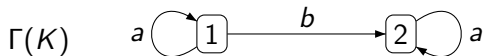


$$K = \langle a, bab^{-1} \rangle$$

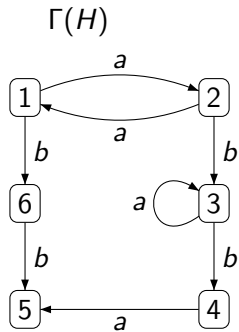
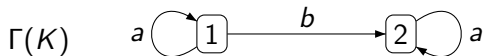


$$H = \langle a^2, abab^{-1}a^{-1}, ab^2ab^{-2} \rangle$$

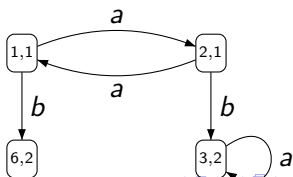
Computing the intersection of two subgroups



Computing the intersection of two subgroups



$$H \cap K = \langle a^2, abab^{-1}a^{-1} \rangle$$



Malnormal subgroups

- ▶ H is *malnormal* if $x \notin H$ implies $H \cap x^{-1}Hx = 1$

Malnormal subgroups

- ▶ H is *malnormal* if $x \notin H$ implies $H \cap x^{-1}Hx = 1$
- ▶ Equivalently: no word $u \neq 1$ may label a loop at two different vertices of $\Gamma(H)$

Malnormal subgroups

- ▶ H is *malnormal* if $x \notin H$ implies $H \cap x^{-1}Hx = 1$
- ▶ Equivalently: no word $u \neq 1$ may label a loop at two different vertices of $\Gamma(H)$
- ▶ To decide this property, consider the square $\Gamma(H) \times_A \Gamma(H)$: H is malnormal if and only if every non-diagonal connected component of this graph is a tree

Pure subgroups

- ▶ H is *pure* if $x^n \in H$ and $n \neq 0$ implies $x \in H$ (also called: *radical-free* or *isolated*)

Pure subgroups

- ▶ H is *pure* if $x^n \in H$ and $n \neq 0$ implies $x \in H$ (also called: *radical-free* or *isolated*)
- ▶ Every malnormal subgroup is pure (if $x^n \in H$ and $n \neq 0$, then $x^n \in H^x$)

Pure subgroups

- ▶ H is *pure* if $x^n \in H$ and $n \neq 0$ implies $x \in H$ (also called: *radical-free* or *isolated*)
- ▶ Every malnormal subgroup is pure (if $x^n \in H$ and $n \neq 0$, then $x^n \in H^x$)
- ▶ Equivalent property: for every word $u \neq 1$, $n \neq 0$, if u^n labels a loop at a vertex p of $\Gamma(H)$, then u labels a loop at that vertex

Pure subgroups

- ▶ H is *pure* if $x^n \in H$ and $n \neq 0$ implies $x \in H$ (also called: *radical-free* or *isolated*)
- ▶ Every malnormal subgroup is pure (if $x^n \in H$ and $n \neq 0$, then $x^n \in H^x$)
- ▶ Equivalent property: for every word $u \neq 1$, $n \neq 0$, if u^n labels a loop at a vertex p of $\Gamma(H)$, then u labels a loop at that vertex
- ▶ In other words, the *automaton* $\Gamma(H)$ is counter-free, its language is star-free

Pure subgroups

- ▶ H is *pure* if $x^n \in H$ and $n \neq 0$ implies $x \in H$ (also called: *radical-free* or *isolated*)
- ▶ Every malnormal subgroup is pure (if $x^n \in H$ and $n \neq 0$, then $x^n \in H^x$)
- ▶ Equivalent property: for every word $u \neq 1$, $n \neq 0$, if u^n labels a loop at a vertex p of $\Gamma(H)$, then u labels a loop at that vertex
- ▶ In other words, the *automaton* $\Gamma(H)$ is counter-free, its language is star-free
- ▶ To decide this property, verify whether the transition monoid of the automaton $\Gamma(H)$ is aperiodic (contains no non-trivial subgroup)

Pure subgroups

- ▶ H is *pure* if $x^n \in H$ and $n \neq 0$ implies $x \in H$ (also called: *radical-free* or *isolated*)
- ▶ Every malnormal subgroup is pure (if $x^n \in H$ and $n \neq 0$, then $x^n \in H^x$)
- ▶ Equivalent property: for every word $u \neq 1$, $n \neq 0$, if u^n labels a loop at a vertex p of $\Gamma(H)$, then u labels a loop at that vertex
- ▶ In other words, the *automaton* $\Gamma(H)$ is counter-free, its language is star-free
- ▶ To decide this property, verify whether the transition monoid of the automaton $\Gamma(H)$ is aperiodic (contains no non-trivial subgroup)
- ▶ Beware: this is a PSPACE-complete problem

Subgroups of subgroups

- ▶ $H \leq_{fg} K \leq_{fg} F$ if and only if there exists $\Gamma(H) \rightarrow \Gamma(K)$, a labeled graph morphism mapping 1 to 1

Subgroups of subgroups

- ▶ $H \leq_{fg} K \leq_{fg} F$ if and only if there exists $\Gamma(H) \rightarrow \Gamma(K)$, a labeled graph morphism mapping 1 to 1
- ▶ Why? because a reduced word is in H iff it labels a loop at 1 in $\Gamma(H)$

Subgroups of subgroups

- ▶ $H \leq_{fg} K \leq_{fg} F$ if and only if there exists $\Gamma(H) \rightarrow \Gamma(K)$, a labeled graph morphism mapping 1 to 1
- ▶ Why? because a reduced word is in H iff it labels a loop at 1 in $\Gamma(H)$
- ▶ If $H \leq_{fg} K$, then this graph morphism is unique, written φ_H^K

Subgroups of subgroups

- ▶ $H \leq_{fg} K \leq_{fg} F$ if and only if there exists $\Gamma(H) \rightarrow \Gamma(K)$, a labeled graph morphism mapping 1 to 1
- ▶ Why? because a reduced word is in H iff it labels a loop at 1 in $\Gamma(H)$
- ▶ If $H \leq_{fg} K$, then this graph morphism is unique, written φ_H^K
- ▶ If φ_H^K is one-to-one (on vertices), i.e. $\Gamma(H)$ is a subgraph of $\Gamma(K)$, then $H \leq_{ff} K$

Finite index extensions and covers

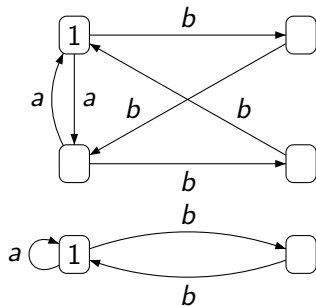
- ▶ Let $H \leq_{fg} K \leq_{fg} F$. First: the *cyclically reduced* case (no vertex of degree 1)

Finite index extensions and covers

- ▶ Let $H \leq_{fg} K \leq_{fg} F$. First: the *cyclically reduced* case (no vertex of degree 1)
- ▶ $H \leq_{fi} K$ if and only if $\Gamma(H) \rightarrow \Gamma(K)$ is a *cover*: an onto morphism such that, for every edge $k \xrightarrow{a} k'$ in $\Gamma(K)$ and every pre-image h of k , there is an edge $h \xrightarrow{a} h'$ for some h' in $\Gamma(H)$

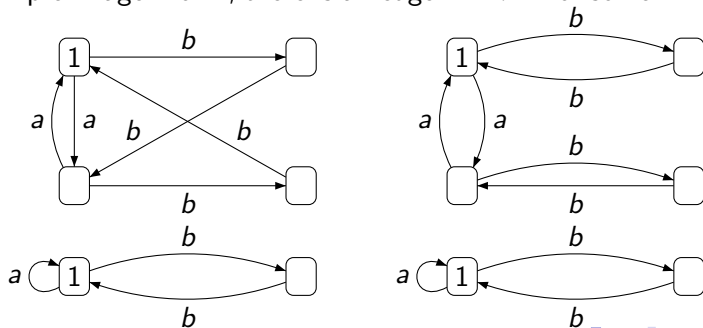
Finite index extensions and covers

- ▶ Let $H \leq_{fg} K \leq_{fg} F$. First: the *cyclically reduced* case (no vertex of degree 1)
- ▶ $H \leq_{fi} K$ if and only if $\Gamma(H) \rightarrow \Gamma(K)$ is a *cover*: an onto morphism such that, for every edge $k \xrightarrow{a} k'$ in $\Gamma(K)$ and every pre-image h of k , there is an edge $h \xrightarrow{a} h'$ for some h' in $\Gamma(H)$



Finite index extensions and covers

- ▶ Let $H \leq_{fg} K \leq_{fg} F$. First: the *cyclically reduced* case (no vertex of degree 1)
- ▶ $H \leq_{fi} K$ if and only if $\Gamma(H) \rightarrow \Gamma(K)$ is a *cover*: an onto morphism such that, for every edge $k \xrightarrow{a} k'$ in $\Gamma(K)$ and every pre-image h of k , there is an edge $h \xrightarrow{a} h'$ for some h' in $\Gamma(H)$



Finite index extensions and covers

- ▶ Let $H \leq_{fg} K \leq_{fg} F$. First: the *cyclically reduced* case (no vertex of degree 1)

Finite index extensions and covers

- ▶ Let $H \leq_{fg} K \leq_{fg} F$. First: the *cyclically reduced* case (no vertex of degree 1)
- ▶ Second: *the general case*. $\Gamma(H)$ consists of a tail, labeled $\text{tail}(H)$, and a cyclically reduced part $\text{cc}(\Gamma(H))$

Finite index extensions and covers

- ▶ Let $H \leq_{fg} K \leq_{fg} F$. First: the *cyclically reduced* case (no vertex of degree 1)
- ▶ Second: *the general case*. $\Gamma(H)$ consists of a tail, labeled $\text{tail}(H)$, and a cyclically reduced part $\text{cc}(\Gamma(H))$
- ▶ $\text{tail}(H)$ is the maximum common prefix of all the reduced words in H

Finite index extensions and covers

- ▶ Let $H \leq_{fg} K \leq_{fg} F$. First: the *cyclically reduced* case (no vertex of degree 1)
- ▶ Second: *the general case*. $\Gamma(H)$ consists of a tail, labeled $\text{tail}(H)$, and a cyclically reduced part $\text{cc}(\Gamma(H))$
- ▶ $\text{tail}(H)$ is the maximum common prefix of all the reduced words in H
- ▶ $H \leq_{fi} K$ if and only if $\text{tail}(H) = \text{tail}(K)$ and φ_H^K is a cover $\text{cc}(\Gamma(H)) \rightarrow \text{cc}(\Gamma(K))$ (technical verification)

Finitely many finite index extensions

- ▶ Since $H \leq_{fi} K$ implies the existence of a cover $\Gamma(H) \rightarrow \Gamma(K)$, H has finitely many finite index extensions

Finitely many finite index extensions

- ▶ Since $H \leq_{fi} K$ implies the existence of a cover $\Gamma(H) \rightarrow \Gamma(K)$,
 H has finitely many finite index extensions
- ▶ which are all computable

Finitely many finite index extensions

- ▶ Since $H \leq_{fi} K$ implies the existence of a cover $\Gamma(H) \rightarrow \Gamma(K)$,
 H has finitely many finite index extensions
- ▶ which are all computable

Corollary

H has a maximal fi extension, noted H_{fi} , which is computable

Finitely many finite index extensions

- ▶ Since $H \leq_{fi} K$ implies the existence of a cover $\Gamma(H) \rightarrow \Gamma(K)$, H has finitely many finite index extensions
- ▶ which are all computable

Corollary

H has a maximal fi extension, noted H_{fi} , which is computable

- ▶ One ingredient missing

Greenberg's theorem

Theorem

If $H \leq_{fi} K, L$, then $H \leq_{fi} \langle K, L \rangle$

Greenberg's theorem

Theorem

If $H \leq_{fi} K, L$, then $H \leq_{fi} \langle K, L \rangle$

- ▶ To compute $\Gamma(\langle K, L \rangle)$, take the union of $\Gamma(K)$ and $\Gamma(L)$, identify origins, and fold

Greenberg's theorem

Theorem

If $H \leq_{fi} K, L$, then $H \leq_{fi} \langle K, L \rangle$

- ▶ To compute $\Gamma(\langle K, L \rangle)$, take the union of $\Gamma(K)$ and $\Gamma(L)$, identify origins, and fold
- ▶ Suppose that $\Gamma(H) \rightarrow \Gamma(K)$ and $\Gamma(H) \rightarrow \Gamma(L)$ are covers

Greenberg's theorem

Theorem

If $H \leq_{\text{fi}} K, L$, then $H \leq_{\text{fi}} \langle K, L \rangle$

- ▶ To compute $\Gamma(\langle K, L \rangle)$, take the union of $\Gamma(K)$ and $\Gamma(L)$, identify origins, and fold
- ▶ Suppose that $\Gamma(H) \rightarrow \Gamma(K)$ and $\Gamma(H) \rightarrow \Gamma(L)$ are covers
- ▶ Then for each $k \in \Gamma(K)$, if $1 \xrightarrow{u} k$, then there exists $\ell \in \Gamma(L)$ such that $1 \xrightarrow{u} \ell$

Greenberg's theorem

Theorem

If $H \leq_{fi} K, L$, then $H \leq_{fi} \langle K, L \rangle$

- ▶ To compute $\Gamma(\langle K, L \rangle)$, take the union of $\Gamma(K)$ and $\Gamma(L)$, identify origins, and fold
- ▶ Suppose that $\Gamma(H) \rightarrow \Gamma(K)$ and $\Gamma(H) \rightarrow \Gamma(L)$ are covers
- ▶ Then for each $k \in \Gamma(K)$, if $1 \xrightarrow{u} k$, then there exists $\ell \in \Gamma(L)$ such that $1 \xrightarrow{u} \ell$
- ▶ Folding from $\Gamma(K) \cup \Gamma(L)$ to $\Gamma(\langle K, L \rangle)$: take the transitive closure of $k \sim \ell$ if there exists u such that $1 \xrightarrow{u} k$ and $1 \xrightarrow{u} \ell$, that is, if (k, ℓ) is in the connected component of $(1, 1)$ in $\Gamma(K \cap L)$

Greenberg's theorem

Theorem

If $H \leq_{fi} K, L$, then $H \leq_{fi} \langle K, L \rangle$

- ▶ To compute $\Gamma(\langle K, L \rangle)$, take the union of $\Gamma(K)$ and $\Gamma(L)$, identify origins, and fold
- ▶ Suppose that $\Gamma(H) \rightarrow \Gamma(K)$ and $\Gamma(H) \rightarrow \Gamma(L)$ are covers
- ▶ Then for each $k \in \Gamma(K)$, if $1 \xrightarrow{u} k$, then there exists $\ell \in \Gamma(L)$ such that $1 \xrightarrow{u} \ell$
- ▶ Folding from $\Gamma(K) \cup \Gamma(L)$ to $\Gamma(\langle K, L \rangle)$: take the transitive closure of $k \sim \ell$ if there exists u such that $1 \xrightarrow{u} k$ and $1 \xrightarrow{u} \ell$, that is, if (k, ℓ) is in the connected component of $(1, 1)$ in $\Gamma(K \cap L)$
- ▶ which is easily seen to define a cover from $\Gamma(K)$ and $\Gamma(L)$

An algebraic characterization of H_{fi}

- ▶ Define the *commensurator* of H as the set of all g such that $H \cap H^g \leq_{\text{fi}} H, H^g$

An algebraic characterization of H_{fi}

- ▶ Define the *commensurator* of H as the set of all g such that $H \cap H^g \leq_{fi} H, H^g$
- ▶ the commensurator is a subgroup of F , and a fi extension of H : elementary

An algebraic characterization of H_{fi}

- ▶ Define the *commensurator* of H as the set of all g such that $H \cap H^g \leq_{fi} H, H^g$
- ▶ the commensurator is a subgroup of F , and a fi extension of H : elementary
- ▶ if $H \leq_{fi} K$ and $g \in K$, then $H^g \leq_{fi} K$ and hence $H \cap H^g \leq_{fi} K$. Therefore $H \cap H^g \leq_{fi} H, H^g$: g is in the commensurator

An algebraic characterization of H_{fi}

- ▶ Define the *commensurator* of H as the set of all g such that $H \cap H^g \leq_{fi} H, H^g$
- ▶ the commensurator is a subgroup of F , and a fi extension of H : elementary
- ▶ if $H \leq_{fi} K$ and $g \in K$, then $H^g \leq_{fi} K$ and hence $H \cap H^g \leq_{fi} K$. Therefore $H \cap H^g \leq_{fi} H, H^g$: g is in the commensurator

Theorem

H_{fi} is the commensurator of H

Now compute the finite index extensions!

- ▶ Starting from $\Gamma(H)$, which state identifications yield a finite index extension?

Now compute the finite index extensions!

- ▶ Starting from $\Gamma(H)$, which state identifications yield a finite index extension?
- ▶ Recall: $\Gamma(H)$ has a tail, and a cyclic core $cc(\Gamma(H))$

Now compute the finite index extensions!

- ▶ Starting from $\Gamma(H)$, which state identifications yield a finite index extension?
- ▶ Recall: $\Gamma(H)$ has a tail, and a cyclic core $cc(\Gamma(H))$
- ▶ $\tilde{L}_p(H)$ = the language accepted by $cc(\Gamma(H))$ with initial state p , all states final
 $L_p(H)$ = set of reduced words in $\tilde{L}_p(H)$

Now compute the finite index extensions!

- ▶ Starting from $\Gamma(H)$, which state identifications yield a finite index extension?
- ▶ Recall: $\Gamma(H)$ has a tail, and a cyclic core $cc(\Gamma(H))$
- ▶ $\tilde{L}_p(H)$ = the language accepted by $cc(\Gamma(H))$ with initial state p , all states final
 $L_p(H)$ = set of reduced words in $\tilde{L}_p(H)$
- ▶ if $p \in tail(\Gamma(H))$ and $q \in cc(\Gamma(H))$, then $\Gamma(H)/(p = q)$ is not a finite index extension
- ▶ because the tail is not preserved

Now compute the finite index extensions!

- ▶ Starting from $\Gamma(H)$, which state identifications yield a finite index extension?
- ▶ Recall: $\Gamma(H)$ has a tail, and a cyclic core $cc(\Gamma(H))$
- ▶ $\tilde{L}_p(H)$ = the language accepted by $cc(\Gamma(H))$ with initial state p , all states final
 $L_p(H)$ = set of reduced words in $\tilde{L}_p(H)$
- ▶ if $p, q \in cc(\Gamma(H))$ and $L_p(H) \neq L_q(H)$, then $\Gamma(H)/(p = q)$ is not a finite index extension

Now compute the finite index extensions!

- ▶ Starting from $\Gamma(H)$, which state identifications yield a finite index extension?
- ▶ Recall: $\Gamma(H)$ has a tail, and a cyclic core $cc(\Gamma(H))$
- ▶ $\tilde{L}_p(H)$ = the language accepted by $cc(\Gamma(H))$ with initial state p , all states final
 $L_p(H)$ = set of reduced words in $\tilde{L}_p(H)$

Theorem

$\Gamma(H)/(p = q)$ is a finite index extension if and only if
 $p, q \in cc(\Gamma(H))$ and $L_p(H) = L_q(H)$ (or $\tilde{L}_p(H) = \tilde{L}_q(H)$)

Counting the finite index extensions of H

- ▶ Say $p \sim q$ is $p, q \in \text{cc}(\Gamma(H))$ and $L_p(H) = L_q(H)$

Counting the finite index extensions of H

- ▶ Say $p \sim q$ is $p, q \in \text{cc}(\Gamma(H))$ and $L_p(H) = L_q(H)$
- ▶ \sim is a union of connected components of $\text{cc}(\Gamma(H)) \times_A \text{cc}(\Gamma(H))$

Counting the finite index extensions of H

- ▶ Say $p \sim q$ is $p, q \in \text{cc}(\Gamma(H))$ and $L_p(H) = L_q(H)$
- ▶ \sim is a union of connected components of $\text{cc}(\Gamma(H)) \times_A \text{cc}(\Gamma(H))$

if $p \xrightarrow{u} r$, then $L_r(H) = \{\text{red}(\bar{u}x) \mid x \in L_p(H)\}$

if $p \sim q$ and $(p, q) \xrightarrow{u} (p', q')$, then $p' \sim q'$

Counting the finite index extensions of H

- ▶ Say $p \sim q$ is $p, q \in \text{cc}(\Gamma(H))$ and $L_p(H) = L_q(H)$
- ▶ \sim is a union of connected components of $\text{cc}(\Gamma(H)) \times_A \text{cc}(\Gamma(H))$
- ▶ and $p \sim q$ if and only if the component projections from to connected component of (p, q) are covers of $\text{cc}(\Gamma(H))$ and $\text{cc}(\Gamma(K))$

Counting the finite index extensions of H

- ▶ Say $p \sim q$ is $p, q \in \text{cc}(\Gamma(H))$ and $L_p(H) = L_q(H)$
- ▶ \sim is a union of connected components of $\text{cc}(\Gamma(H)) \times_A \text{cc}(\Gamma(H))$
- ▶ and $p \sim q$ if and only if the component projections from to connected component of (p, q) are covers of $\text{cc}(\Gamma(H))$ and $\text{cc}(\Gamma(K))$
- ▶ Every \sim -class contains some $(1, r)$, so if $\text{cc}(\Gamma(H))$ has n vertices, there are at most n classes, and each is non trivial (so the corresponding quotient has at most $n/2$ elements)

Counting the finite index extensions of H

- ▶ Say $p \sim q$ is $p, q \in \text{cc}(\Gamma(H))$ and $L_p(H) = L_q(H)$
- ▶ \sim is a union of connected components of $\text{cc}(\Gamma(H)) \times_A \text{cc}(\Gamma(H))$
- ▶ and $p \sim q$ if and only if the component projections from to connected component of (p, q) are covers of $\text{cc}(\Gamma(H))$ and $\text{cc}(\Gamma(K))$
- ▶ Every \sim -class contains some $(1, r)$, so if $\text{cc}(\Gamma(H))$ has n vertices, there are at most n classes, and each is non trivial (so the corresponding quotient has at most $n/2$ elements)
- ▶ So if there are at most $f(n)$ finite index extensions, then $f(n) \leq nf(n/2)$, $f(n) \leq n^{\frac{1}{2}(1+\log_2 n)} = \sqrt{n 2^{\log_2^2 n}}$

Computing all finite extensions of H

- ▶ Let $H \leq_{fi} K \leq F$ and $\varphi: \Gamma(H) \rightarrow \Gamma(K)$. Then $p \in cc(\Gamma(H))$ iff $\varphi(p) \in cc(\Gamma(K))$, and in that case $L_p(H) = L_{\varphi(p)}(K)$

Computing all finite extensions of H

- ▶ Let $H \leq_{fi} K \leq F$ and $\varphi: \Gamma(H) \rightarrow \Gamma(K)$. Then $p \in cc(\Gamma(H))$ iff $\varphi(p) \in cc(\Gamma(K))$, and in that case $L_p(H) = L_{\varphi(p)}(K)$
- ▶ So $H \leq_{fi} K$ iff $\Gamma(K)$ is obtained by performing some identifications $p = q$ such that $p, q \in cc(\Gamma(H))$ and $L_p(H) = L_q(H)$, and then reducing the graph

Computing all finite extensions of H

- ▶ Let $H \leq_{fi} K \leq F$ and $\varphi: \Gamma(H) \rightarrow \Gamma(K)$. Then $p \in cc(\Gamma(H))$ iff $\varphi(p) \in cc(\Gamma(K))$, and in that case $L_p(H) = L_{\varphi(p)}(K)$
- ▶ So $H \leq_{fi} K$ iff $\Gamma(K)$ is obtained by performing some identifications $p = q$ such that $p, q \in cc(\Gamma(H))$ and $L_p(H) = L_q(H)$, and then reducing the graph
- ▶ $\Gamma(H_{fi})$ is obtained by performing all identifications $p = q$ such that $p, q \in cc(\Gamma(H))$ and $L_p(H) = L_q(H)$. No reduction is necessary

A fast algorithm to compute H_{fi}

- ▶ $H \leq_{fi} K$ iff $\Gamma(K)$ is obtained by performing some identifications $p = q$ such that $p, q \in cc(\Gamma(H))$ and $L_p(H) = L_q(H)$ (or $\tilde{L}_p(H) = \tilde{L}_q(H)$), and then reducing the graph
- ▶ $\Gamma(H_{fi})$ is obtained by performing all identifications $p = q$ such that $p, q \in cc(\Gamma(H))$ and $L_p(H) = L_q(H)$ (or $\tilde{L}_p(H) = \tilde{L}_q(H)$). No reduction is necessary

A fast algorithm to compute H_{fi}

- ▶ $H \leq_{fi} K$ iff $\Gamma(K)$ is obtained by performing some identifications $p = q$ such that $p, q \in cc(\Gamma(H))$ and $L_p(H) = L_q(H)$ (or $\tilde{L}_p(H) = \tilde{L}_q(H)$), and then reducing the graph
- ▶ $\Gamma(H_{fi})$ is obtained by performing all identifications $p = q$ such that $p, q \in cc(\Gamma(H))$ and $L_p(H) = L_q(H)$ (or $\tilde{L}_p(H) = \tilde{L}_q(H)$). No reduction is necessary
- ▶ To find all these pairs (p, q) : minimize the automaton $cc(\Gamma(H))$ with all states final! This can be done in $\mathcal{O}(n \log n)$

Free groups and subgroups

Representation of subgroups

Rank, index and conjugates

Intersection of subgroups

Malnormal and pure subgroups

Subgroups of subgroups

The maximal finite index extension of H

Computing the fi extensions of H

A fast algorithm to compute H_{fi}

Computing the malnormal closure of H

An automata-theoretic characterization

A polynomial algorithm

Malnormal subgroups

Definition

H is *malnormal* if for each $g \notin H$, $H \cap H^g = 1$

Malnormal subgroups

Definition

H is *malnormal* if for each $g \notin H$, $H \cap H^g = 1$

- ▶ H is malnormal iff every non-diagonal connected component of $\Gamma(H) \times_A \Gamma(H)$ is a tree. . .

Malnormal subgroups

Definition

H is *malnormal* if for each $g \notin H$, $H \cap H^g = 1$

- ▶ H is malnormal iff every non-diagonal connected component of $\Gamma(H) \times_A \Gamma(H)$ is a tree. . .
- ▶ . . . iff $L_p(H) \cap L_q(H)$ is finite for each $p \neq q \in \text{cc}(\Gamma(H))$

Malnormal subgroups

Definition

H is *malnormal* if for each $g \notin H$, $H \cap H^g = 1$

- ▶ H is malnormal iff every non-diagonal connected component of $\Gamma(H) \times_A \Gamma(H)$ is a tree. . .
- ▶ . . . iff $L_p(H) \cap L_q(H)$ is finite for each $p \neq q \in \text{cc}(\Gamma(H))$
- ▶ malnormality is decidable in time $\mathcal{O}(n^2 \log n)$

Malnormal subgroups

Definition

H is *malnormal* if for each $g \notin H$, $H \cap H^g = 1$

- ▶ H is malnormal iff every non-diagonal connected component of $\Gamma(H) \times_A \Gamma(H)$ is a tree. . .
- ▶ . . . iff $L_p(H) \cap L_q(H)$ is finite for each $p \neq q \in \text{cc}(\Gamma(H))$
- ▶ malnormality is decidable in time $\mathcal{O}(n^2 \log n)$
- ▶ A malnormal subgroup has no proper finite index extensions

Computing the malnormal closure of H

Definition

The malnormal closure H_{mal} of H is the least malnormal subgroup containing H

Computing the malnormal closure of H

Definition

The malnormal closure H_{mal} of H is the least malnormal subgroup containing H

- ▶ The algebraic closure of H is malnormal, so H_{mal} exists, is an algebraic extension

Computing the malnormal closure of H

Definition

The malnormal closure H_{mal} of H is the least malnormal subgroup containing H

- ▶ The algebraic closure of H is malnormal, so H_{mal} exists, is an algebraic extension
- ▶ Recall that K is malnormal iff $L_p(K) \cap L_q(K)$ is finite for each $p \neq q \in \text{cc}(\Gamma(K))$

Computing the malnormal closure of H

Definition

The malnormal closure H_{mal} of H is the least malnormal subgroup containing H

- ▶ The algebraic closure of H is malnormal, so H_{mal} exists, is an algebraic extension
- ▶ Recall that K is malnormal iff $L_p(K) \cap L_q(K)$ is finite for each $p \neq q \in \text{cc}(\Gamma(K))$
- ▶ $\Gamma(H_{\text{mal}})$ is computed by repeatedly applying the following procedure: *identify all pairs $p, q \in \text{cc}(\Gamma(H))$ such that $L_p(H) \cap L_q(H)$ is infinite, and reduce.*

Computing the malnormal closure of H

Definition

The malnormal closure H_{mal} of H is the least malnormal subgroup containing H

- ▶ The algebraic closure of H is malnormal, so H_{mal} exists, is an algebraic extension
- ▶ Recall that K is malnormal iff $L_p(K) \cap L_q(K)$ is finite for each $p \neq q \in \text{cc}(\Gamma(K))$
- ▶ $\Gamma(H_{\text{mal}})$ is computed by repeatedly applying the following procedure: *identify all pairs $p, q \in \text{cc}(\Gamma(H))$ such that $L_p(H) \cap L_q(H)$ is infinite, and reduce.*
- ▶ H_{mal} is computable in time $\mathcal{O}(n^3 \log n)$

Thank you for your attention!