# Whitehead minimization and computation of algebraic closures in polynomial time

**Enric Ventura**

Departament de Matemàtica Aplicada III

Universitat Politècnica de Catalunya

Omsk International Workshop

August 19th, 2009.

# Outline

# Outline

# Definitions and notation

- $A = \{a_1, \ldots, a_n\}$ is a finite alphabet ($n$ letters).
- $A^{\pm 1} = A \cup A^{-1} = \{a_1, a_1^{-1}, \ldots, a_n, a_n^{-1}\}$.
- Usually, $A = \{a, b, c\}$.
- $(A^{\pm 1})^*$ the free monoid on $A^{\pm 1}$ (words on $A^{\pm 1}$).
- $F_A = (A^{\pm 1})^* / \sim$ is the free group on $A$ (words on $A^{\pm 1}$ modulo reduction).
- Every $w \in A^*$ has a unique reduced form,
- 1 denotes the empty word, and $|\cdot|$ the (shortest) length in $F_A$:
  $|1| = 0, \quad |aba^{-1}| = |abbb^{-1}a^{-1}| = 3, \quad |uv| \leqslant |u| + |v|$.

# Definitions and notation

- $A = \{a_1, \ldots, a_n\}$ is a finite alphabet ($n$ letters).
- $A^{\pm 1} = A \cup A^{-1} = \{a_1, a_1^{-1}, \ldots, a_n, a_n^{-1}\}$.
- Usually, $A = \{a, b, c\}$.
- $(A^{\pm 1})^*$ the free monoid on $A^{\pm 1}$ (words on $A^{\pm 1}$).
- $F_A = (A^{\pm 1})^* / \sim$ is the free group on $A$ (words on $A^{\pm 1}$ modulo reduction).
- Every $w \in A^*$ has a unique reduced form,
- 1 denotes the empty word, and $|\cdot|$ the (shortest) length in $F_A$:
  $|1| = 0, \quad |aba^{-1}| = |abbb^{-1}a^{-1}| = 3, \quad |uv| \leqslant |u| + |v|$.

# Definitions and notation

- $A = \{a_1, \ldots, a_n\}$ is a finite alphabet ($n$ letters).
- $A^{\pm 1} = A \cup A^{-1} = \{a_1, a_1^{-1}, \ldots, a_n, a_n^{-1}\}$.
- Usually, $A = \{a, b, c\}$.
- $(A^{\pm 1})^*$ the free monoid on $A^{\pm 1}$ (words on $A^{\pm 1}$).
- $F_A = (A^{\pm 1})^* / \sim$ is the free group on $A$ (words on $A^{\pm 1}$ modulo reduction).
- Every $w \in A^*$ has a unique reduced form,
- 1 denotes the empty word, and $|\cdot|$ the (shortest) length in $F_A$:
  $|1| = 0, \quad |aba^{-1}| = |abbb^{-1}a^{-1}| = 3, \quad |uv| \leqslant |u| + |v|$.

# Definitions and notation

- $A = \{a_1, \ldots, a_n\}$ is a finite alphabet ($n$ letters).
- $A^{\pm 1} = A \cup A^{-1} = \{a_1, a_1^{-1}, \ldots, a_n, a_n^{-1}\}$.
- Usually, $A = \{a, b, c\}$.
- $(A^{\pm 1})^*$ the free monoid on $A^{\pm 1}$ (words on $A^{\pm 1}$).
- $F_A = (A^{\pm 1})^* / \sim$ is the free group on $A$ (words on $A^{\pm 1}$ modulo reduction).
- Every $w \in A^*$ has a unique reduced form,
- 1 denotes the empty word, and $|\cdot|$ the (shortest) length in $F_A$:
  $|1| = 0, \quad |aba^{-1}| = |abbb^{-1}a^{-1}| = 3, \quad |uv| \leqslant |u| + |v|$.

# Definitions and notation

- $A = \{a_1, \ldots, a_n\}$ is a finite alphabet ($n$ letters).
- $A^{\pm 1} = A \cup A^{-1} = \{a_1, a_1^{-1}, \ldots, a_n, a_n^{-1}\}$.
- Usually, $A = \{a, b, c\}$.
- $(A^{\pm 1})^*$ the free monoid on $A^{\pm 1}$ (words on $A^{\pm 1}$).
- $F_A = (A^{\pm 1})^* / \sim$ is the free group on $A$ (words on $A^{\pm 1}$ modulo reduction).
- Every $w \in A^*$ has a unique reduced form,
- 1 denotes the empty word, and $|\cdot|$ the (shortest) length in $F_A$:
  $|1| = 0, \quad |aba^{-1}| = |abbb^{-1}a^{-1}| = 3, \quad |uv| \leqslant |u| + |v|$.

# Definitions and notation

- $A = \{a_1, \ldots, a_n\}$ is a finite alphabet ($n$ letters).
- $A^{\pm 1} = A \cup A^{-1} = \{a_1, a_1^{-1}, \ldots, a_n, a_n^{-1}\}$.
- Usually, $A = \{a, b, c\}$.
- $(A^{\pm 1})^*$ the free monoid on $A^{\pm 1}$ (words on $A^{\pm 1}$).
- $F_A = (A^{\pm 1})^* / \sim$ is the free group on $A$ (words on $A^{\pm 1}$ modulo reduction).
- Every $w \in A^*$ has a unique reduced form,
- 1 denotes the empty word, and $|\cdot|$ the (shortest) length in $F_A$:
  $|1| = 0, \quad |aba^{-1}| = |abbb^{-1}a^{-1}| = 3, \quad |uv| \leqslant |u| + |v|$.

# Definitions and notation

- $A = \{a_1, \ldots, a_n\}$ is a finite alphabet ($n$ letters).
- $A^{\pm 1} = A \cup A^{-1} = \{a_1, a_1^{-1}, \ldots, a_n, a_n^{-1}\}$.
- Usually, $A = \{a, b, c\}$.
- $(A^{\pm 1})^*$ the free monoid on $A^{\pm 1}$ (words on $A^{\pm 1}$).
- $F_A = (A^{\pm 1})^* / \sim$ is the free group on $A$ (words on $A^{\pm 1}$ modulo reduction).
- Every $w \in A^*$ has a unique reduced form,
- 1 denotes the empty word, and $|\cdot|$ the (shortest) length in $F_A$:
  $|1| = 0, \quad |aba^{-1}| = |abbb^{-1}a^{-1}| = 3, \quad |uv| \leqslant |u| + |v|$.

# Motivation

- In basic linear algebra:

$$U \leqslant V \leqslant K^n \quad \Rightarrow \quad V = U \oplus L.$$

- In $\mathbb{Z}^n$, the analog is almost true:

$$U \leqslant V \leqslant \mathbb{Z}^n \quad \Rightarrow \quad \exists\ U \leq_{fi} U' \leqslant V \text{ s.t. } V = U' \oplus L.$$

- In $F(A)$, the analog is ...

far from true because $H \leqslant K \not\Rightarrow r(H) \leqslant r(K)$ ...

# Motivation

- In basic linear algebra:

$$U \leqslant V \leqslant K^n \quad \Rightarrow \quad V = U \oplus L.$$

- In $\mathbb{Z}^n$, the analog is almost true:

$$U \leqslant V \leqslant \mathbb{Z}^n \quad \Rightarrow \quad \exists\ U \leq_{fi} U' \leqslant V \text{ s.t. } V = U' \oplus L.$$

- In $F(A)$, the analog is ...

far from true because $H \leqslant K \not\Rightarrow r(H) \leqslant r(K)$ ...

# Motivation

- In basic linear algebra:

$$U \leqslant V \leqslant K^n \quad \Rightarrow \quad V = U \oplus L.$$

- In $\mathbb{Z}^n$, the analog is almost true:

$$U \leqslant V \leqslant \mathbb{Z}^n \quad \Rightarrow \quad \exists \ U \leq_{fi} U' \leqslant V \text{ s.t. } V = U' \oplus L.$$

- In $F(A)$, the analog is ...

far from true because $H \leqslant K \not\Rightarrow r(H) \leqslant r(K)$ ...

# Motivation

- In basic linear algebra:

$$U \leqslant V \leqslant K^n \quad \Rightarrow \quad V = U \oplus L.$$

- In $\mathbb{Z}^n$, the analog is almost true:

$$U \leqslant V \leqslant \mathbb{Z}^n \quad \Rightarrow \quad \exists \ U \leq_{fi} U' \leqslant V \text{ s.t. } V = U' \oplus L.$$

- In $F(A)$, the analog is ...

almost true again, ... in the sense of Takahasi.

# Algebraic and transcendental elements

Mimicking field theory...

## Definition

*Let $H \leqslant F(A)$ and $w \in F(A)$. We say that $w$ is*

- *algebraic over $H$ if $\exists\, 1 \neq e_H(x) \in H * \langle x \rangle$ such that $e_H(w) = 1$;*
- *transcendental over $H$ otherwise.*

## Observation

*$w$ is transcendental over $H \iff \langle H, w \rangle \simeq H * \langle w \rangle$*

*$\iff H$ is contained in a proper f.f. of $\langle H, w \rangle$.*

## Problem

*$w_1, w_2$ algebraic over $H \not\Rightarrow w_1 w_2$ algebraic over $H$.*

$H = \langle a, \bar{b}ab, \bar{c}ac \rangle \leqslant \langle a, b, c \rangle$, and $w_1 = b$, $w_2 = \bar{c}$

# Algebraic and transcendental elements

Mimicking field theory...

## Definition

*Let $H \leqslant F(A)$ and $w \in F(A)$. We say that $w$ is*

- *algebraic over $H$ if $\exists\, 1 \neq e_H(x) \in H * \langle x \rangle$ such that $e_H(w) = 1$;*
- *transcendental over $H$ otherwise.*

## Observation

*$w$ is transcendental over $H \iff \langle H, w \rangle \simeq H * \langle w \rangle$*
*$\iff H$ is contained in a proper f.f. of $\langle H, w \rangle$.*

## Problem

*$w_1, w_2$ algebraic over $H \not\Rightarrow w_1 w_2$ algebraic over $H$.*

$H = \langle a, \overline{b}ab, \overline{c}ac \rangle \leqslant \langle a, b, c \rangle$, and $w_1 = b$, $w_2 = \overline{c}$

# Algebraic and transcendental elements

Mimicking field theory...

## Definition

*Let $H \leqslant F(A)$ and $w \in F(A)$. We say that $w$ is*

- *algebraic over $H$ if $\exists\, 1 \neq e_H(x) \in H * \langle x \rangle$ such that $e_H(w) = 1$;*
- *transcendental over $H$ otherwise.*

## Observation

*$w$ is transcendental over $H \iff \langle H, w \rangle \simeq H * \langle w \rangle$*
*$\iff H$ is contained in a proper f.f. of $\langle H, w \rangle$.*

## Problem

*$w_1, w_2$ algebraic over $H \not\Rightarrow w_1 w_2$ algebraic over $H$.*

$H = \langle a, \overline{b}ab, \overline{c}ac \rangle \leqslant \langle a, b, c \rangle$, and $w_1 = b$, $w_2 = \overline{c}$

# Algebraic and transcendental elements

Mimicking field theory...

## Definition

*Let $H \leqslant F(A)$ and $w \in F(A)$. We say that $w$ is*

- *algebraic over $H$ if $\exists\, 1 \neq e_H(x) \in H * \langle x \rangle$ such that $e_H(w) = 1$;*
- *transcendental over $H$ otherwise.*

## Observation

*$w$ is transcendental over $H$ $\iff \langle H, w \rangle \simeq H * \langle w \rangle$*
*$\iff H$ is contained in a proper f.f. of $\langle H, w \rangle$.*

## Problem

*$w_1, w_2$ algebraic over $H \not\Rightarrow w_1 w_2$ algebraic over $H$.*

$H = \langle a, \overline{b}ab, \overline{c}ac \rangle \leqslant \langle a, b, c \rangle$, and $w_1 = b$, $w_2 = \overline{c}$

# Algebraic and transcendental elements

Mimicking field theory...

## Definition

*Let $H \leqslant F(A)$ and $w \in F(A)$. We say that w is*

- *algebraic over H if $\exists\, 1 \neq e_H(x) \in H * \langle x \rangle$ such that $e_H(w) = 1$;*
- *transcendental over H otherwise.*

## Observation

*w is transcendental over H* $\iff \langle H, w \rangle \simeq H * \langle w \rangle$

$\iff$ *H is contained in a proper f.f. of* $\langle H, w \rangle$.

## Problem

$w_1, w_2$ *algebraic over H* $\not\Rightarrow$ $w_1 w_2$ *algebraic over H.*

$H = \langle a, \overline{b}ab, \overline{c}ac \rangle \leqslant \langle a, b, c \rangle$, and $w_1 = b$, $w_2 = \overline{c}$

# Algebraic and transcendental elements

Mimicking field theory...

## Definition

*Let $H \leqslant F(A)$ and $w \in F(A)$. We say that w is*

- *algebraic over H if $\exists\, 1 \neq e_H(x) \in H * \langle x \rangle$ such that $e_H(w) = 1$;*
- *transcendental over H otherwise.*

## Observation

*w is transcendental over H $\iff \langle H, w \rangle \simeq H * \langle w \rangle$*
*$\iff$ H is contained in a proper f.f. of $\langle H, w \rangle$.*

## Problem

*$w_1, w_2$ algebraic over H $\not\Rightarrow$ $w_1 w_2$ algebraic over H.*

$H = \langle a, \bar{b}ab, \bar{c}ac \rangle \leqslant \langle a, b, c \rangle$, and $w_1 = b$, $w_2 = \bar{c}$

## Algebraic and free extensions

A relative notion works better...

**Definition**

*Let $H \leqslant K \leqslant F(A)$ and $w \in K$. We say that $w$ is*

- *$K$-algebraic over $H$ if $\forall$ free factorization $K = K_1 * K_2$ with $H \leqslant K_1$, we have $w \in K_1$;*
- *$K$-transcendental over $H$ otherwise.*

**Observation**

*$w$ is algebraic over $H$ if and only if it is $\langle H, w\rangle$-algebraic over $H$.*

**Observation**

*If $w_1$ and $w_2$ are $K$-algebraic over $H$, then so is $w_1 w_2$.*

# Algebraic and free extensions

A relative notion works better...

**Definition**

*Let $H \leqslant K \leqslant F(A)$ and $w \in K$. We say that $w$ is*

- *$K$-algebraic over $H$ if $\forall$ free factorization $K = K_1 * K_2$ with $H \leqslant K_1$, we have $w \in K_1$;*
- *$K$-transcendental over $H$ otherwise.*

**Observation**

*$w$ is algebraic over $H$ if and only if it is $\langle H, w\rangle$-algebraic over $H$.*

**Observation**

*If $w_1$ and $w_2$ are $K$-algebraic over $H$, then so is $w_1 w_2$.*

# Algebraic and free extensions

A relative notion works better...

## Definition

*Let $H \leqslant K \leqslant F(A)$ and $w \in K$. We say that $w$ is*

- *$K$-algebraic over $H$ if $\forall$ free factorization $K = K_1 * K_2$ with $H \leqslant K_1$, we have $w \in K_1$;*
- *$K$-transcendental over $H$ otherwise.*

## Observation

*$w$ is algebraic over $H$ if and only if it is $\langle H, w\rangle$-algebraic over $H$.*

## Observation

*If $w_1$ and $w_2$ are $K$-algebraic over $H$, then so is $w_1 w_2$.*

# Algebraic and free extensions

A relative notion works better...

## Definition

*Let $H \leqslant K \leqslant F(A)$ and $w \in K$. We say that w is*

- *$K$-algebraic over $H$ if $\forall$ free factorization $K = K_1 * K_2$ with $H \leqslant K_1$, we have $w \in K_1$;*
- *$K$-transcendental over $H$ otherwise.*

## Observation

*$w$ is algebraic over $H$ if and only if it is $\langle H, w \rangle$-algebraic over $H$.*

## Observation

*If $w_1$ and $w_2$ are $K$-algebraic over $H$, then so is $w_1 w_2$.*

# Algebraic and free extensions

A relative notion works better...

## Definition

*Let $H \leqslant K \leqslant F(A)$ and $w \in K$. We say that w is*

- *$K$-algebraic over $H$ if $\forall$ free factorization $K = K_1 * K_2$ with $H \leqslant K_1$, we have $w \in K_1$;*
- *$K$-transcendental over $H$ otherwise.*

## Observation

*$w$ is algebraic over $H$ if and only if it is $\langle H, w\rangle$-algebraic over $H$.*

## Observation

*If $w_1$ and $w_2$ are $K$-algebraic over $H$, then so is $w_1 w_2$.*

# Algebraic and free extensions

## Definition

*Let $H \leqslant K \leqslant F(A)$.*

*We say that $H \leqslant K$ is an algebraic extension, denoted $H \leq_{alg} K$,*

$\iff$ *every $w \in K$ is $K$-algebraic over $H$,*

$\iff$ *$H$ is not contained in any proper free factor of $K$,*

$\iff$ *$H \leqslant K_1 \leqslant K_1 * K_2 = K$ implies $K_2 = 1$.*

*We say that $H \leqslant K$ is a free extension, denoted $H \leq_{ff} K$,*

$\iff$ *every $w \in K$ is $K$-transcendental over $H$,*

$\iff$ *$H \leqslant H * L = K$ for some $L$.*

# Algebraic and free extensions

## Definition

*Let* $H \leqslant K \leqslant F(A)$.
*We say that* $H \leqslant K$ *is an algebraic extension, denoted* $H \leq_{alg} K$,
$\iff$ *every* $w \in K$ *is* $K$*-algebraic over* $H$,
$\iff$ $H$ *is not contained in any proper free factor of* $K$,
$\iff$ $H \leqslant K_1 \leqslant K_1 * K_2 = K$ *implies* $K_2 = 1$.

*We say that* $H \leqslant K$ *is a free extension, denoted* $H \leq_{ff} K$,
$\iff$ *every* $w \in K$ *is* $K$*-transcendental over* $H$,
$\iff$ $H \leqslant H * L = K$ *for some* $L$.

# Algebraic and free extensions

## Definition

*Let* $H \leqslant K \leqslant F(A)$.
*We say that* $H \leqslant K$ *is an* *algebraic extension*, *denoted* $H \leq_{alg} K$,
$\Longleftrightarrow$ *every* $w \in K$ *is* $K$*-algebraic over* $H$,
$\Longleftrightarrow$ $H$ *is not contained in any proper free factor of* $K$,
$\Longleftrightarrow$ $H \leqslant K_1 \leqslant K_1 * K_2 = K$ *implies* $K_2 = 1$.

*We say that* $H \leqslant K$ *is a* *free extension*, *denoted* $H \leq_{ff} K$,
$\Longleftrightarrow$ *every* $w \in K$ *is* $K$*-transcendental over* $H$,
$\Longleftrightarrow$ $H \leqslant H * L = K$ *for some* $L$.

# Algebraic and free extensions

## Definition

*Let $H \leqslant K \leqslant F(A)$.*

*We say that $H \leqslant K$ is an algebraic extension, denoted $H \leq_{alg} K$,*

$\iff$ *every $w \in K$ is $K$-algebraic over $H$,*

$\iff$ *$H$ is not contained in any proper free factor of $K$,*

$\iff$ *$H \leqslant K_1 \leqslant K_1 * K_2 = K$ implies $K_2 = 1$.*

*We say that $H \leqslant K$ is a free extension, denoted $H \leq_{ff} K$,*

$\iff$ *every $w \in K$ is $K$-transcendental over $H$,*

$\iff$ *$H \leqslant H * L = K$ for some $L$.*

# Algebraic and free extensions

## Definition

*Let* $H \leqslant K \leqslant F(A)$.
*We say that* $H \leqslant K$ *is an algebraic extension, denoted* $H \leq_{alg} K$,

$\iff$ *every* $w \in K$ *is* $K$*-algebraic over* $H$,

$\iff$ $H$ *is not contained in any proper free factor of* $K$,

$\iff$ $H \leqslant K_1 \leqslant K_1 * K_2 = K$ *implies* $K_2 = 1$.

*We say that* $H \leqslant K$ *is a free extension, denoted* $H \leq_{ff} K$,

$\iff$ *every* $w \in K$ *is* $K$*-transcendental over* $H$,

$\iff$ $H \leqslant H * L = K$ *for some* $L$.

# Algebraic and free extensions

## Example

- $\langle a \rangle \leqslant_{ff} \langle a, b \rangle \leqslant_{ff} \langle a, b, c \rangle$, and $\langle x^r \rangle \leqslant_{alg} \langle x \rangle$, $\forall x \in F_A \; \forall \; 0 \neq r \in \mathbb{Z}$.
- if $r(H) \geqslant 2$ and $r(K) \leqslant 2$ then $H \leqslant_{alg} K$.
- $H \leqslant_{alg} K \leqslant_{alg} L$ implies $H \leqslant_{alg} L$.
- $H \leqslant_{ff} K \leqslant_{ff} L$ implies $H \leqslant_{ff} L$.
- $H \leqslant_{alg} L$ and $H \leqslant K \leqslant L$ imply $K \leqslant_{alg} L$ but not necessarily $H \leqslant_{alg} K$.
- $H \leqslant_{ff} L$ and $H \leqslant K \leqslant L$ imply $H \leqslant_{ff} K$ but not necessarily $K \leqslant_{ff} L$.

How many algebraic extensions does a given $H$ have in $F(A)$ ?

Can we compute them all ?

# Algebraic and free extensions

## Example

- $\langle a\rangle \leqslant_{ff} \langle a, b\rangle \leqslant_{ff} \langle a, b, c\rangle$, *and* $\langle x^r\rangle \leqslant_{alg} \langle x\rangle$, $\forall x \in F_A\ \forall\ 0 \neq r \in \mathbb{Z}$.
- *if* $r(H) \geqslant 2$ *and* $r(K) \leqslant 2$ *then* $H \leqslant_{alg} K$.
- $H \leqslant_{alg} K \leqslant_{alg} L$ *implies* $H \leqslant_{alg} L$.
- $H \leqslant_{ff} K \leqslant_{ff} L$ *implies* $H \leqslant_{ff} L$.
- $H \leqslant_{alg} L$ *and* $H \leqslant K \leqslant L$ *imply* $K \leqslant_{alg} L$ *but not necessarily* $H \leqslant_{alg} K$.
- $H \leqslant_{ff} L$ *and* $H \leqslant K \leqslant L$ *imply* $H \leqslant_{ff} K$ *but not necessarily* $K \leqslant_{ff} L$.

How many algebraic extensions does a given $H$ have in $F(A)$ ?

Can we compute them all ?

# Algebraic and free extensions

## Example

- $\langle a\rangle \leqslant_{ff} \langle a, b\rangle \leqslant_{ff} \langle a, b, c\rangle$, and $\langle x^r\rangle \leqslant_{alg} \langle x\rangle$, $\forall x \in F_A\ \forall\ 0 \neq r \in \mathbb{Z}$.
- if $r(H) \geqslant 2$ and $r(K) \leqslant 2$ then $H \leqslant_{alg} K$.
- $H \leqslant_{alg} K \leqslant_{alg} L$ implies $H \leqslant_{alg} L$.
- $H \leqslant_{ff} K \leqslant_{ff} L$ implies $H \leqslant_{ff} L$.
- $H \leqslant_{alg} L$ and $H \leqslant K \leqslant L$ imply $K \leqslant_{alg} L$ but not necessarily $H \leqslant_{alg} K$.
- $H \leqslant_{ff} L$ and $H \leqslant K \leqslant L$ imply $H \leqslant_{ff} K$ but not necessarily $K \leqslant_{ff} L$.

How many algebraic extensions does a given $H$ have in $F(A)$ ?

Can we compute them all ?

# Algebraic and free extensions

## Example

- $\langle a\rangle \leqslant_{ff} \langle a, b\rangle \leqslant_{ff} \langle a, b, c\rangle$, *and* $\langle x^r\rangle \leqslant_{alg} \langle x\rangle$, $\forall x \in F_A \ \forall\ 0 \neq r \in \mathbb{Z}$.
- *if* $r(H) \geqslant 2$ *and* $r(K) \leqslant 2$ *then* $H \leqslant_{alg} K$.
- $H \leqslant_{alg} K \leqslant_{alg} L$ *implies* $H \leqslant_{alg} L$.
- $H \leqslant_{ff} K \leqslant_{ff} L$ *implies* $H \leqslant_{ff} L$.
- $H \leqslant_{alg} L$ *and* $H \leqslant K \leqslant L$ *imply* $K \leqslant_{alg} L$ *but not necessarily* $H \leqslant_{alg} K$.
- $H \leqslant_{ff} L$ *and* $H \leqslant K \leqslant L$ *imply* $H \leqslant_{ff} K$ *but not necessarily* $K \leqslant_{ff} L$.

How many algebraic extensions does a given $H$ have in $F(A)$ ?

Can we compute them all ?

# Algebraic and free extensions

## Example

- $\langle a \rangle \leqslant_{ff} \langle a, b \rangle \leqslant_{ff} \langle a, b, c \rangle$, and $\langle x^r \rangle \leqslant_{alg} \langle x \rangle$, $\forall x \in F_A \; \forall \; 0 \neq r \in \mathbb{Z}$.
- if $r(H) \geqslant 2$ and $r(K) \leqslant 2$ then $H \leqslant_{alg} K$.
- $H \leqslant_{alg} K \leqslant_{alg} L$ implies $H \leqslant_{alg} L$.
- $H \leqslant_{ff} K \leqslant_{ff} L$ implies $H \leqslant_{ff} L$.
- $H \leqslant_{alg} L$ and $H \leqslant K \leqslant L$ imply $K \leqslant_{alg} L$ but not necessarily $H \leqslant_{alg} K$.
- $H \leqslant_{ff} L$ and $H \leqslant K \leqslant L$ imply $H \leqslant_{ff} K$ but not necessarily $K \leqslant_{ff} L$.

How many algebraic extensions does a given $H$ have in $F(A)$ ?

Can we compute them all ?

# Algebraic and free extensions

## Example

- $\langle a \rangle \leqslant_{ff} \langle a, b \rangle \leqslant_{ff} \langle a, b, c \rangle$, *and* $\langle x^r \rangle \leqslant_{alg} \langle x \rangle$, $\forall x \in F_A\ \forall\ 0 \neq r \in \mathbb{Z}$.
- *if* $r(H) \geqslant 2$ *and* $r(K) \leqslant 2$ *then* $H \leqslant_{alg} K$.
- $H \leqslant_{alg} K \leqslant_{alg} L$ *implies* $H \leqslant_{alg} L$.
- $H \leqslant_{ff} K \leqslant_{ff} L$ *implies* $H \leqslant_{ff} L$.
- $H \leqslant_{alg} L$ *and* $H \leqslant K \leqslant L$ *imply* $K \leqslant_{alg} L$ *but not necessarily* $H \leqslant_{alg} K$.
- $H \leqslant_{ff} L$ *and* $H \leqslant K \leqslant L$ *imply* $H \leqslant_{ff} K$ *but not necessarily* $K \leqslant_{ff} L$.

How many algebraic extensions does a given $H$ have in $F(A)$ ?

Can we compute them all ?

# Algebraic and free extensions

## Example

- $\langle a\rangle \leqslant_{ff} \langle a, b\rangle \leqslant_{ff} \langle a, b, c\rangle$, and $\langle x^r\rangle \leqslant_{alg} \langle x\rangle$, $\forall x \in F_A\ \forall\ 0 \neq r \in \mathbb{Z}$.
- if $r(H) \geqslant 2$ and $r(K) \leqslant 2$ then $H \leqslant_{alg} K$.
- $H \leqslant_{alg} K \leqslant_{alg} L$ implies $H \leqslant_{alg} L$.
- $H \leqslant_{ff} K \leqslant_{ff} L$ implies $H \leqslant_{ff} L$.
- $H \leqslant_{alg} L$ and $H \leqslant K \leqslant L$ imply $K \leqslant_{alg} L$ but not necessarily $H \leqslant_{alg} K$.
- $H \leqslant_{ff} L$ and $H \leqslant K \leqslant L$ imply $H \leqslant_{ff} K$ but not necessarily $K \leqslant_{ff} L$.

How many algebraic extensions does a given $H$ have in $F(A)$ ?

Can we compute them all ?

# Algebraic and free extensions

## Example

- $\langle a\rangle \leqslant_{ff} \langle a, b\rangle \leqslant_{ff} \langle a, b, c\rangle$, and $\langle x^r\rangle \leqslant_{alg} \langle x\rangle$, $\forall x \in F_A \ \forall\ 0 \neq r \in \mathbb{Z}$.
- if $r(H) \geqslant 2$ and $r(K) \leqslant 2$ then $H \leqslant_{alg} K$.
- $H \leqslant_{alg} K \leqslant_{alg} L$ implies $H \leqslant_{alg} L$.
- $H \leqslant_{ff} K \leqslant_{ff} L$ implies $H \leqslant_{ff} L$.
- $H \leqslant_{alg} L$ and $H \leqslant K \leqslant L$ imply $K \leqslant_{alg} L$ but not necessarily $H \leqslant_{alg} K$.
- $H \leqslant_{ff} L$ and $H \leqslant K \leqslant L$ imply $H \leqslant_{ff} K$ but not necessarily $K \leqslant_{ff} L$.

How many algebraic extensions does a given $H$ have in $F(A)$ ?

Can we compute them all ?

# Takahasi's Theorem

**Theorem (Takahasi, 1951)**

*For every $H \leqslant_{fg} F_A$, the set of algebraic extensions, denoted $\mathcal{AE}(H)$, is finite.*

- Original proof by Takahasi was combinatorial and technical,
- Modern proof, using Stallings automata, is much simpler, and due independently to Ventura (1997), Margolis-Sapir-Weil (2001) and Kapovich-Miasnikov (2002).
- Additionally, $\mathcal{AE}(H)$ is computable.

# Takahasi's Theorem

**Theorem (Takahasi, 1951)**

*For every $H \leqslant_{fg} F_A$, the set of algebraic extensions, denoted $\mathcal{AE}(H)$, is finite.*

- Original proof by Takahasi was combinatorial and technical,
- Modern proof, using Stallings automata, is much simpler, and due independently to Ventura (1997), Margolis-Sapir-Weil (2001) and Kapovich-Miasnikov (2002).
- Additionally, $\mathcal{AE}(H)$ is computable.

# Takahasi's Theorem

**Theorem (Takahasi, 1951)**

*For every $H \leqslant_{fg} F_A$, the set of algebraic extensions, denoted $\mathcal{AE}(H)$, is finite.*

- Original proof by Takahasi was combinatorial and technical,
- Modern proof, using Stallings automata, is much simpler, and due independently to Ventura (1997), Margolis-Sapir-Weil (2001) and Kapovich-Miasnikov (2002).
- Additionally, $\mathcal{AE}(H)$ is computable.

# Takahasi's Theorem

**Theorem (Takahasi, 1951)**

*For every $H \leqslant_{fg} F_A$, the set of algebraic extensions, denoted $\mathcal{AE}(H)$, is finite.*

- Original proof by Takahasi was combinatorial and technical,
- Modern proof, using Stallings automata, is much simpler, and due independently to Ventura (1997), Margolis-Sapir-Weil (2001) and Kapovich-Miasnikov (2002).
- Additionally, $\mathcal{AE}(H)$ is computable.

# Outline

# Stallings automata

## Definition

*A* Stallings automaton *is a finite* $A$*-labeled oriented graph with a distinguished vertex,* $(X, v)$*, such that:*

1- $X$ *is connected,*

2- no *vertex of degree 1 except possibly* $v$ *(*$X$ *is a* core-graph*),*

3- no *two edges with the same label go out of (or in to) the same vertex.*

NO :




YES :

# Stallings automata

## Definition

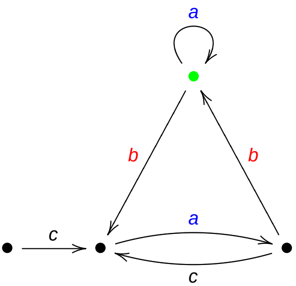
*A* ***Stallings automaton*** *is a finite $A$-labeled oriented graph with a distinguished vertex, $(X, v)$, such that:*

1- *$X$ is connected,*

2- *no vertex of degree 1 except possibly $v$ ($X$ is a* ***core-graph****),*

3- *no two edges with the same label go out of (or in to) the same vertex.*

*NO* :




*YES* :

# Stallings automata

## Definition

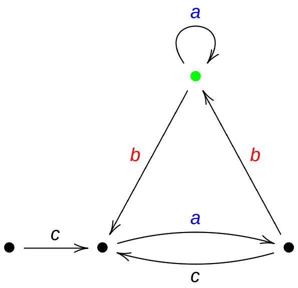
*A Stallings automaton is a finite A-labeled oriented graph with a distinguished vertex, $(X, v)$, such that:*

1- *$X$ is connected,*

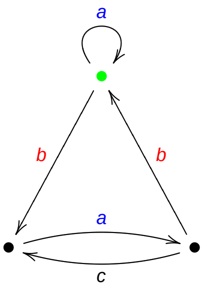
2- *no vertex of degree 1 except possibly $v$ ($X$ is a core-graph),*

3- *no two edges with the same label go out of (or in to) the same vertex.*

NO :




YES :



# Stallings automata

In the influent paper

J. R. Stallings, Topology of finite graphs, Inventiones Math. 71 (1983), 551-565,

Stallings (building on previous works) gave a bijection between finitely generated subgroups of $F_A$ and Stallings automata:

$$\{\text{f.g. subgroups of } F_A\} \quad \longleftrightarrow \quad \{\text{Stallings automata}\},$$

which is crucial for the modern understanding of the lattice of subgroups of $F_A$.

# Stallings automata

In the influent paper

J. R. Stallings, Topology of finite graphs, Inventiones Math. 71 (1983), 551-565,

Stallings (building on previous works) gave a bijection between finitely generated subgroups of $F_A$ and Stallings automata:

$$\{\text{f.g. subgroups of } F_A\} \quad \longleftrightarrow \quad \{\text{Stallings automata}\},$$

which is crucial for the modern understanding of the lattice of subgroups of $F_A$.

# Stallings automata

In the influent paper

J. R. Stallings, Topology of finite graphs, Inventiones Math. 71 (1983), 551-565,

Stallings (building on previous works) gave a bijection between finitely generated subgroups of $F_A$ and Stallings automata:

$$\{\text{f.g. subgroups of } F_A\} \quad \longleftrightarrow \quad \{\text{Stallings automata}\},$$

which is crucial for the modern understanding of the lattice of subgroups of $F_A$.

# Reading the subgroup from the automata

## Definition

*To any given (Stallings) automaton* $(X, v)$*, we associate its* *fundamental group*:

$$\pi(X, v) = \{ \textit{ labels of closed paths at } v \} \leqslant F_A,$$

*clearly, a subgroup of* $F_A$.

$X =$




$$\pi(X, \bullet) = \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$$

$$\pi(X, \bullet) \not\ni bc^{-1}bcaa$$

Membership problem in $\pi(X, \bullet)$ is solvable.

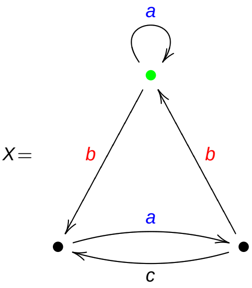
# Reading the subgroup from the automata

## Definition

*To any given (Stallings) automaton* $(X, v)$*, we associate its* ***fundamental group****:*

$$\pi(X, v) = \{\textit{ labels of closed paths at } v\} \leqslant F_A,$$

*clearly, a subgroup of* $F_A$*.*




$\pi(X, \bullet) \;=\; \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$

$\pi(X, \bullet) \;\not\ni\; bc^{-1}bcaa$

Membership problem in $\pi(X, \bullet)$ is solvable.

# Reading the subgroup from the automata

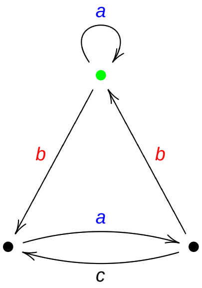
## Definition

*To any given (Stallings) automaton* $(X, v)$*, we associate its fundamental group:*

$$\pi(X, v) = \{ \text{labels of closed paths at } v \} \leqslant F_A,$$

*clearly, a subgroup of* $F_A$*.*

$X =$




$\pi(X, \bullet) = \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$

$\pi(X, \bullet) \not\ni bc^{-1}bcaa$

Membership problem in $\pi(X, \bullet)$ is solvable.

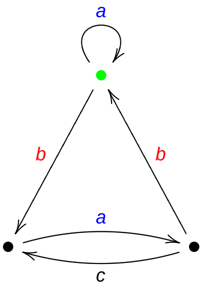
# Reading the subgroup from the automata

## Definition

*To any given (Stallings) automaton $(X, v)$, we associate its fundamental group:*

$$\pi(X, v) = \{\ \textit{labels of closed paths at } v\} \leqslant F_A,$$

*clearly, a subgroup of $F_A$.*

$X =$ 


$$\pi(X, \bullet) \quad = \quad \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$$

$$\pi(X, \bullet) \quad \not\ni \quad bc^{-1}bcaa$$

Membership problem in $\pi(X, \bullet)$ is solvable.

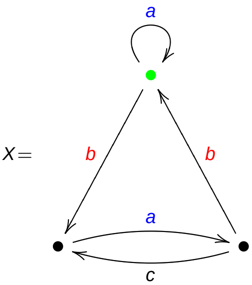
# Reading the subgroup from the automata

## Definition

*To any given (Stallings) automaton* $(X, v)$*, we associate its fundamental group:*

$$\pi(X, v) = \{ \text{ labels of closed paths at } v \} \leqslant F_A,$$

*clearly, a subgroup of* $F_A$*.*




$\pi(X, \bullet) = \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$

$\pi(X, \bullet) \not\ni bc^{-1}bcaa$

Membership problem in $\pi(X, \bullet)$ is solvable.

# Reading the subgroup from the automata

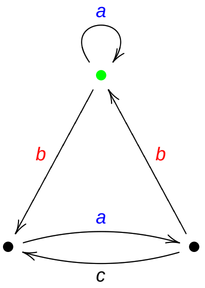
## Definition

*To any given (Stallings) automaton $(X, v)$, we associate its fundamental group:*

$$\pi(X, v) = \{\ \textit{labels of closed paths at } v\} \leqslant F_A,$$

*clearly, a subgroup of $F_A$.*

$X =$




$$\pi(X, \bullet) = \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$$

$$\pi(X, \bullet) \not\ni bc^{-1}bcaa$$

Membership problem in $\pi(X, \bullet)$ is solvable.

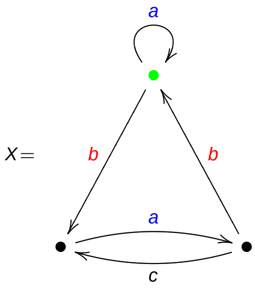
# Reading the subgroup from the automata

## Definition

*To any given (Stallings) automaton $(X, v)$, we associate its fundamental group:*

$$\pi(X, v) = \{\text{ labels of closed paths at } v\} \leqslant F_A,$$

*clearly, a subgroup of $F_A$.*




$$\pi(X, \bullet) = \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$$

$$\pi(X, \bullet) \not\ni bc^{-1}bcaa$$

Membership problem in $\pi(X, \bullet)$ is solvable.

# Reading the subgroup from the automata

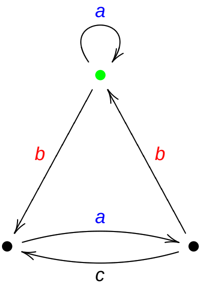
## Definition

*To any given (Stallings) automaton $(X, v)$, we associate its fundamental group:*

$$\pi(X, v) = \{ \text{labels of closed paths at } v \} \leqslant F_A,$$

*clearly, a subgroup of $F_A$.*

$X =$




$$\pi(X, \bullet) = \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$$

$$\pi(X, \bullet) \not\ni bc^{-1}bcaa$$

Membership problem in $\pi(X, \bullet)$ is solvable.

# Reading the subgroup from the automata

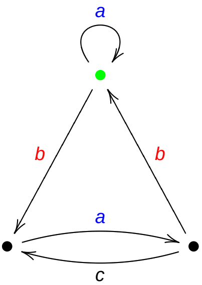
## Definition

*To any given (Stallings) automaton $(X, v)$, we associate its fundamental group:*

$$\pi(X, v) = \{\ \textit{labels of closed paths at } v\} \leqslant F_A,$$

*clearly, a subgroup of $F_A$.*

$X =$




$$\pi(X, \bullet) = \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$$

$$\pi(X, \bullet) \not\ni bc^{-1}bcaa$$

Membership problem in $\pi(X, \bullet)$ is solvable.

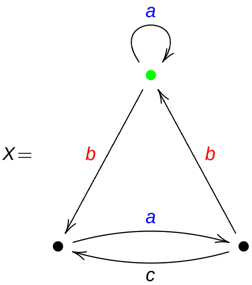
# Reading the subgroup from the automata

## Definition

*To any given (Stallings) automaton* $(X, v)$, *we associate its* *fundamental group*:

$$\pi(X, v) = \{\text{ labels of closed paths at } v\} \leqslant F_A,$$

*clearly, a subgroup of* $F_A$.




$$\pi(X, \bullet) = \{1, a, a^{-1}, bab, bc^{-1}b, babab^{-1}cb^{-1}, \ldots\}$$

$$\pi(X, \bullet) \not\ni bc^{-1}bcaa$$

Membership problem in $\pi(X, \bullet)$ is solvable.

# A basis for $\pi(X, v)$

**Proposition**

*For every Stallings automaton $(X, v)$, the group $\pi(X, v)$ is free of rank $rk(\pi(X, v)) = 1 - |VX| + |EX|$.*

**Proof:**

- Take a maximal tree $T$ in $X$.
- Write $T[p, q]$ for the geodesic (i.e. the unique reduced path) in $T$ from $p$ to $q$.
- For every $e \in EX - ET$, $x_e = label(T[v, \iota e] \cdot e \cdot T[\tau e, v])$ belongs to $\pi(X, v)$.
- Not difficult to see that $\{x_e \mid e \in EX - ET\}$ is a basis for $\pi(X, v)$.
- And, $|EX - ET| = |EX| - |ET|$
  $= |EX| - (|VT| - 1) = 1 - |VX| + |EX|$. $\square$

# A basis for $\pi(X,v)$

## Proposition

*For every Stallings automaton $(X,v)$, the group $\pi(X,v)$ is free of rank $rk(\pi(X,v)) = 1 - |VX| + |EX|$.*

**Proof:**

- Take a maximal tree $T$ in $X$.
- Write $T[p,q]$ for the geodesic (i.e. the unique reduced path) in $T$ from $p$ to $q$.
- For every $e \in EX - ET$, $x_e = label(T[v,\iota e] \cdot e \cdot T[\tau e, v])$ belongs to $\pi(X,v)$.
- Not difficult to see that $\{x_e \mid e \in EX - ET\}$ is a basis for $\pi(X,v)$.
- And, $$\begin{aligned} |EX - ET| &= |EX| - |ET| \\ &= |EX| - (|VT| - 1) = 1 - |VX| + |EX|. \ \square \end{aligned}$$

# A basis for $\pi(X, v)$

## Proposition

*For every Stallings automaton $(X, v)$, the group $\pi(X, v)$ is free of rank $rk(\pi(X, v)) = 1 - |VX| + |EX|$.*

**Proof:**

- Take a maximal tree $T$ in $X$.
- Write $T[p, q]$ for the geodesic (i.e. the unique reduced path) in $T$ from $p$ to $q$.
- For every $e \in EX - ET$, $x_e = label(T[v, \iota e] \cdot e \cdot T[\tau e, v])$ belongs to $\pi(X, v)$.
- Not difficult to see that $\{x_e \mid e \in EX - ET\}$ is a basis for $\pi(X, v)$.
- And, $$\begin{aligned} |EX - ET| &= |EX| - |ET| \\ &= |EX| - (|VT| - 1) = 1 - |VX| + |EX|. \;\square \end{aligned}$$

# A basis for $\pi(X, v)$

## Proposition

*For every Stallings automaton $(X, v)$, the group $\pi(X, v)$ is free of rank $rk(\pi(X, v)) = 1 - |VX| + |EX|$.*

**Proof:**

- Take a maximal tree $T$ in $X$.
- Write $T[p, q]$ for the geodesic (i.e. the unique reduced path) in $T$ from $p$ to $q$.
- For every $e \in EX - ET$, $x_e = label(T[v, \iota e] \cdot e \cdot T[\tau e, v])$ belongs to $\pi(X, v)$.
- Not difficult to see that $\{x_e \mid e \in EX - ET\}$ is a basis for $\pi(X, v)$.
- And, $$\begin{aligned} |EX - ET| &= |EX| - |ET| \\ &= |EX| - (|VT| - 1) = 1 - |VX| + |EX|. \ \square \end{aligned}$$

# A basis for $\pi(X, v)$

## Proposition

*For every Stallings automaton $(X, v)$, the group $\pi(X, v)$ is free of rank $rk(\pi(X, v)) = 1 - |VX| + |EX|$.*

**Proof:**

- Take a maximal tree $T$ in $X$.
- Write $T[p, q]$ for the geodesic (i.e. the unique reduced path) in $T$ from $p$ to $q$.
- For every $e \in EX - ET$, $x_e = label(T[v, \iota e] \cdot e \cdot T[\tau e, v])$ belongs to $\pi(X, v)$.
- Not difficult to see that $\{x_e \mid e \in EX - ET\}$ is a basis for $\pi(X, v)$.
- And, $$\begin{aligned} |EX - ET| &= |EX| - |ET| \\ &= |EX| - (|VT| - 1) = 1 - |VX| + |EX|. \ \square \end{aligned}$$

# A basis for $\pi(X, v)$

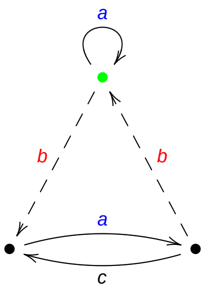
## Proposition

*For every Stallings automaton $(X, v)$, the group $\pi(X, v)$ is free of rank $rk(\pi(X, v)) = 1 - |VX| + |EX|$.*

**Proof:**

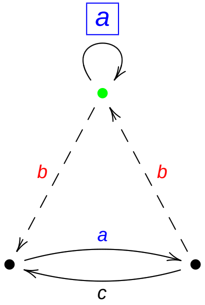
- Take a maximal tree $T$ in $X$.
- Write $T[p, q]$ for the geodesic (i.e. the unique reduced path) in $T$ from $p$ to $q$.
- For every $e \in EX - ET$, $x_e = label(T[v, \iota e] \cdot e \cdot T[\tau e, v])$ belongs to $\pi(X, v)$.
- Not difficult to see that $\{x_e \mid e \in EX - ET\}$ is a basis for $\pi(X, v)$.
- And, $$\begin{aligned} |EX - ET| &= |EX| - |ET| \\ &= |EX| - (|VT| - 1) = 1 - |VX| + |EX|. \ \square \end{aligned}$$

# Example



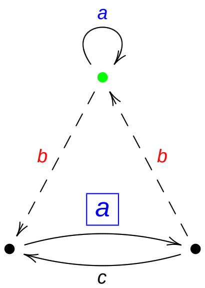

$H = \langle \quad \rangle$

# Example



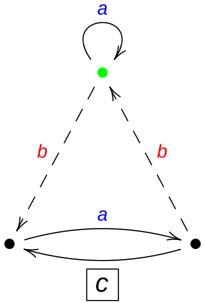

$H = \langle {\color{blue}a}, \quad \rangle$

# Example



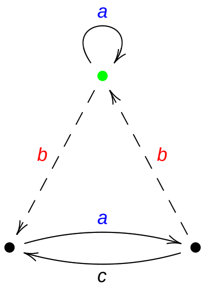

$H = \langle a,\ bab, \quad \rangle$

# Example



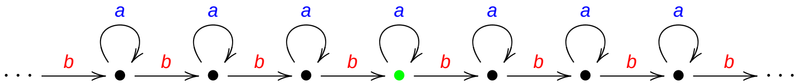

$H = \langle a,\ bab,\ b^{-1}cb^{-1} \rangle$

# Example




$H = \langle a,\ bab,\ b^{-1}cb^{-1} \rangle$
$rk(H) = 1 - 3 + 5 = 3.$

# Example-2




$$F_{\aleph_0} \simeq H = \langle \ldots, b^{-2}ab^{2}, b^{-1}ab, a, bab^{-1}, b^{2}ab^{-2}, \ldots \rangle \leqslant F_2.$$

# Constructing the automata from the subgroup

In any automaton containing the following situation, for $x \in A^{\pm 1}$,

$$\begin{tikzcd} \bullet \arrow[r, "x"] \arrow[dr, "x"'] & u \\ & v \end{tikzcd}$$

we can fold and identify vertices $u$ and $v$ to obtain

$$\bullet \xrightarrow{x} u = v \ .$$

This operation, $(X, v) \rightsquigarrow (X', v)$, is called a Stallings folding.

# Constructing the automata from the subgroup

In any automaton containing the following situation, for $x \in A^{\pm 1}$,

$$\begin{tikzcd} \bullet \arrow[r, "x"] \arrow[dr, "x"'] & u \\ & v \end{tikzcd}$$

we can fold and identify vertices $u$ and $v$ to obtain

$$\bullet \xrightarrow{\quad x \quad} u = v \ .$$

This operation, $(X, v) \rightsquigarrow (X', v)$, is called a Stallings folding.

# Constructing the automata from the subgroup

In any automaton containing the following situation, for $x \in A^{\pm 1}$,

$$\begin{tikzcd} \bullet \arrow[r, "x"] \arrow[dr, "x"'] & u \\ & v \end{tikzcd}$$

we can fold and identify vertices $u$ and $v$ to obtain

$$\bullet \xrightarrow{\quad x \quad} u = v \ .$$

This operation, $(X, v) \rightsquigarrow (X', v)$, is called a Stallings folding.

# Constructing the automata from the subgroup

**Lemma (Stallings)**

*If $(X, v) \rightsquigarrow (X', v')$ is a Stallings folding then $\pi(X, v) = \pi(X', v')$.*

*Given a f.g. subgroup $H = \langle w_1, \ldots w_m \rangle \leqslant F_A$ (we assume $w_i$ are reduced words), do the following:*

1. *Draw the flower automaton,*
2. *Perform successive foldings until obtaining a Stallings automaton, denoted $\Gamma(H)$.*

# Constructing the automata from the subgroup

## Lemma (Stallings)

*If $(X, v) \rightsquigarrow (X', v')$ is a Stallings folding then $\pi(X, v) = \pi(X', v')$.*

*Given a f.g. subgroup $H = \langle w_1, \ldots w_m \rangle \leqslant F_A$ (we assume $w_i$ are reduced words), do the following:*

1- *Draw the flower automaton,*

2- *Perform successive foldings until obtaining a Stallings automaton, denoted $\Gamma(H)$.*

# Constructing the automata from the subgroup

## Lemma (Stallings)

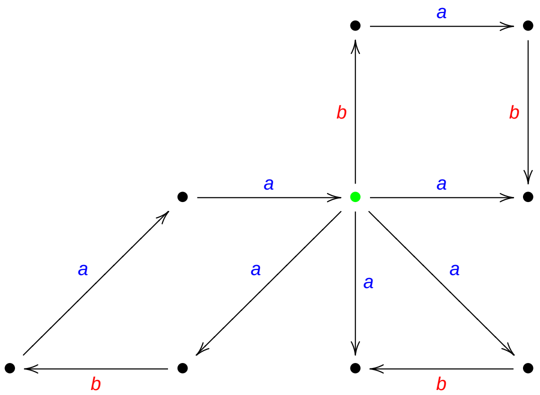
*If* $(X, v) \rightsquigarrow (X', v')$ *is a Stallings folding then* $\pi(X, v) = \pi(X', v')$.

*Given a f.g. subgroup* $H = \langle w_1, \ldots w_m \rangle \leqslant F_A$ *(we assume* $w_i$ *are reduced words), do the following:*

1- *Draw the flower automaton,*

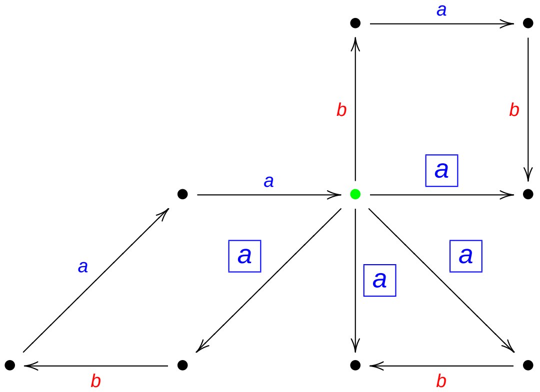
2- *Perform successive foldings until obtaining a Stallings automaton, denoted* $\Gamma(H)$.

# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$



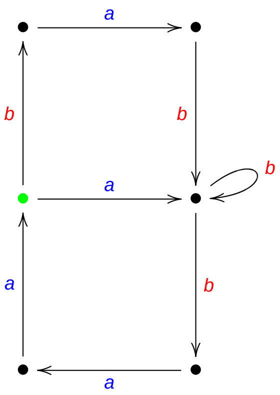

*Flower(H)*

# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$



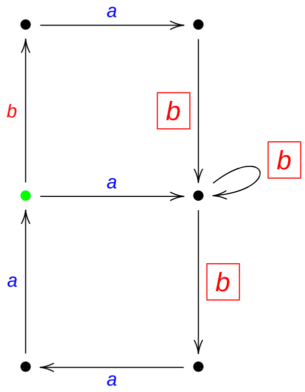

*Flower(H)*

# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$



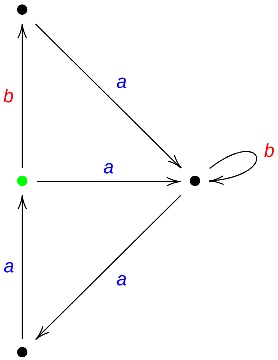

Folding $\#1$

# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$



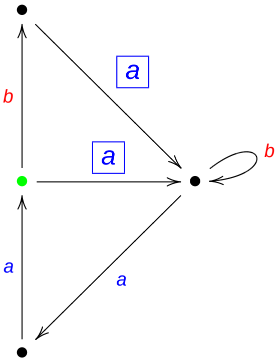

Folding $\#1$.

# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$



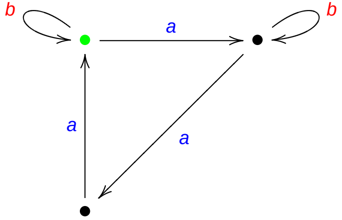

Folding $\#2$.

# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$




Folding $\#2$.

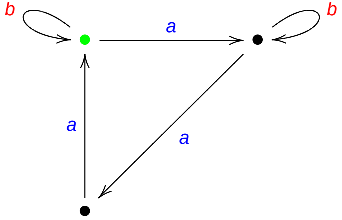
# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$




Folding $\#3$. $\Gamma(H)$

By Stallings Lemma, $\pi(\Gamma(H), \bullet) = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$

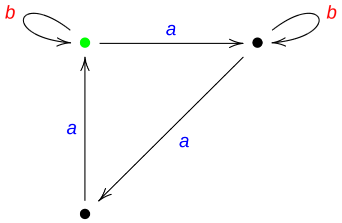
# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$




Folding #3. $\Gamma(H)$

By Stallings Lemma, $\pi(\Gamma(H), \bullet) = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$

# Example: $H = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$




Folding $\#3$. $\Gamma(H)$

By Stallings Lemma, $\pi(\Gamma(H), \bullet) = \langle baba^{-1}, aba^{-1}, aba^2 \rangle$
$= \langle b, aba^{-1}, a^3 \rangle$

# Local confluence

It can be shown that

## Proposition

*The automaton $\Gamma(H)$ does not depend on the sequence of foldings*

## Proposition

*The automaton $\Gamma(H)$ does not depend on the generators of $H$.*

## Theorem

*The following is a bijection:*

$$\begin{array}{rcl} \{\text{f.g. subgroups of } F_A\} & \longleftrightarrow & \{\text{Stallings automata}\} \\ H & \to & \Gamma(H) \\ \pi(X,v) & \leftarrow & (X,v) \end{array}$$

# Local confluence

It can be shown that

**Proposition**

*The automaton $\Gamma(H)$ does not depend on the sequence of foldings*

**Proposition**

*The automaton $\Gamma(H)$ does not depend on the generators of $H$.*

**Theorem**

*The following is a bijection:*

$$\begin{array}{rcl} \{\text{f.g. subgroups of } F_A\} & \longleftrightarrow & \{\text{Stallings automata}\} \\ H & \to & \Gamma(H) \\ \pi(X, v) & \leftarrow & (X, v) \end{array}$$

# Local confluence

It can be shown that

**Proposition**

*The automaton $\Gamma(H)$ does not depend on the sequence of foldings*

**Proposition**

*The automaton $\Gamma(H)$ does not depend on the generators of $H$.*

**Theorem**

*The following is a bijection:*

$$\begin{array}{rcl} \{\textit{f.g. subgroups of } F_A\} & \longleftrightarrow & \{\textit{Stallings automata}\} \\ H & \longrightarrow & \Gamma(H) \\ \pi(X, v) & \longleftarrow & (X, v) \end{array}$$

**Corollary (Nielsen-Schreier)**

*Every subgroup of $F_A$ is free.*

- Finite automata work for the finitely generated case, but everything extends easily to the general case (using infinite graphs).
- The original proof (1920's) is combinatorial and much more technical.

# Nielsen-Schreier Theorem

**Corollary (Nielsen-Schreier)**

*Every subgroup of $F_A$ is free.*

- Finite automata work for the finitely generated case, but everything extends easily to the general case (using infinite graphs).

- The original proof (1920's) is combinatorial and much more technical.

# Nielsen-Schreier Theorem

## Corollary (Nielsen-Schreier)

*Every subgroup of $F_A$ is free.*

- Finite automata work for the finitely generated case, but everything extends easily to the general case (using infinite graphs).
- The original proof (1920's) is combinatorial and much more technical.

# Outline

# Takahasi's theorem

## Definition

*Let $H \leqslant K \leqslant F(A)$. Then, $H \leqslant K$ is algebraic if and only if $H$ is not contained in any proper free factor of $K$.*

## Theorem (Takahasi, 1951)

*For every $H \leqslant_{fg} F_A$, the set of algebraic extensions, $\mathcal{AE}(H)$, is finite.*

**Proof** (Ventura; Margolis-Sapir-Weil; Kapovich-Miasnikov):

- Consider $\tilde{\Gamma}(H)$, the result of attaching all possible (infinite) "hairs" to $\Gamma(H)$ (i.e. the covering of the bouquet corresponding to $H$).
- Given $H \leqslant K$ (both f.g.), we can obtain $\tilde{\Gamma}(K)$ from $\tilde{\Gamma}(H)$ by performing the appropriate identifications of vertices (plus subsequent foldings).

# Takahasi's theorem

## Definition

*Let $H \leqslant K \leqslant F(A)$. Then, $H \leqslant K$ is algebraic if and only if $H$ is not contained in any proper free factor of $K$.*

## Theorem (Takahasi, 1951)

*For every $H \leqslant_{fg} F_A$, the set of algebraic extensions, $\mathcal{AE}(H)$, is finite.*

**Proof** (Ventura; Margolis-Sapir-Weil; Kapovich-Miasnikov):

- Consider $\tilde{\Gamma}(H)$, the result of attaching all possible (infinite) "hairs" to $\Gamma(H)$ (i.e. the covering of the bouquet corresponding to $H$).
- Given $H \leqslant K$ (both f.g.), we can obtain $\tilde{\Gamma}(K)$ from $\tilde{\Gamma}(H)$ by performing the appropriate identifications of vertices (plus subsequent foldings).

## Definition

*Let $H \leqslant K \leqslant F(A)$. Then, $H \leqslant K$ is algebraic if and only if $H$ is not contained in any proper free factor of $K$.*

## Theorem (Takahasi, 1951)

*For every $H \leqslant_{fg} F_A$, the set of algebraic extensions, $\mathcal{AE}(H)$, is finite.*

**Proof** (Ventura; Margolis-Sapir-Weil; Kapovich-Miasnikov):

- Consider $\tilde{\Gamma}(H)$, the result of attaching all possible (infinite) "hairs" to $\Gamma(H)$ (i.e. the covering of the bouquet corresponding to $H$).
- Given $H \leqslant K$ (both f.g.), we can obtain $\tilde{\Gamma}(K)$ from $\tilde{\Gamma}(H)$ by performing the appropriate identifications of vertices (plus subsequent foldings).

# Takahasi's theorem

## Definition

*Let $H \leqslant K \leqslant F(A)$. Then, $H \leqslant K$ is algebraic if and only if $H$ is not contained in any proper free factor of $K$.*

## Theorem (Takahasi, 1951)

*For every $H \leqslant_{fg} F_A$, the set of algebraic extensions, $\mathcal{AE}(H)$, is finite.*

**Proof** (Ventura; Margolis-Sapir-Weil; Kapovich-Miasnikov):

- Consider $\tilde{\Gamma}(H)$, the result of attaching all possible (infinite) "hairs" to $\Gamma(H)$ (i.e. the covering of the bouquet corresponding to $H$).
- Given $H \leqslant K$ (both f.g.), we can obtain $\tilde{\Gamma}(K)$ from $\tilde{\Gamma}(H)$ by performing the appropriate identifications of vertices (plus subsequent foldings).

# Takahasi's theorem

- Hence, if $H \leqslant K$ (both f.g.) then $\Gamma(K)$ contains as a subgraph either $\Gamma(H)$ or some quotient of it (i.e. $\Gamma(H)$ after some identifications of vertices, $\Gamma(H)/\sim$).
- The overgroups of $H$: $\mathcal{O}(H) = \{\pi(\Gamma(H)/\sim, \bullet) \mid \sim \text{ is a partition of } V\Gamma(H)\}$.
- Hence, for every $H \leqslant K$, there exists $L \in \mathcal{O}(H)$ such that $H \leqslant L \leqslant_{ff} K$.
- Thus, $\mathcal{AE}(H) \subseteq \mathcal{O}(H)$ and so, it is finite. $\square$

# Takahasi's theorem

- Hence, if $H \leqslant K$ (both f.g.) then $\Gamma(K)$ contains as a subgraph either $\Gamma(H)$ or some quotient of it (i.e. $\Gamma(H)$ after some identifications of vertices, $\Gamma(H)/\sim$).
- The overgroups of $H$:
$\mathcal{O}(H) = \{\pi(\Gamma(H)/\sim, \bullet) \mid \sim \text{ is a partition of } V\Gamma(H)\}$.
- Hence, for every $H \leqslant K$, there exists $L \in \mathcal{O}(H)$ such that $H \leqslant L \leqslant_{ff} K$.
- Thus, $\mathcal{AE}(H) \subseteq \mathcal{O}(H)$ and so, it is finite. $\square$

# Takahasi's theorem

- Hence, if $H \leqslant K$ (both f.g.) then $\Gamma(K)$ contains as a subgraph either $\Gamma(H)$ or some quotient of it (i.e. $\Gamma(H)$ after some identifications of vertices, $\Gamma(H)/\sim$).
- The overgroups of $H$:
  $\mathcal{O}(H) = \{\pi(\Gamma(H)/\sim, \bullet) \mid \sim \text{ is a partition of } V\Gamma(H)\}$.
- Hence, for every $H \leqslant K$, there exists $L \in \mathcal{O}(H)$ such that $H \leqslant L \leqslant_{ff} K$.
- Thus, $\mathcal{AE}(H) \subseteq \mathcal{O}(H)$ and so, it is finite. $\square$

# Takahasi's theorem

- Hence, if $H \leqslant K$ (both f.g.) then $\Gamma(K)$ contains as a subgraph either $\Gamma(H)$ or some quotient of it (i.e. $\Gamma(H)$ after some identifications of vertices, $\Gamma(H)/\sim$).
- The overgroups of $H$:
  $\mathcal{O}(H) = \{\pi(\Gamma(H)/\sim, \bullet) \mid \sim \text{ is a partition of } V\Gamma(H)\}$.
- Hence, for every $H \leqslant K$, there exists $L \in \mathcal{O}(H)$ such that $H \leqslant L \leqslant_{ff} K$.
- Thus, $\mathcal{AE}(H) \subseteq \mathcal{O}(H)$ and so, it is finite. $\square$

# Computing $\mathcal{AE}(H)$

## Corollary

*$\mathcal{AE}(H)$ is computable.*

**Proof:**

- Compute $\Gamma(H)$,
- Compute $\Gamma(H)/\sim$ for all partitions $\sim$ of $V\Gamma(H)$,
- Compute $\mathcal{O}(H)$,
- Clean $\mathcal{O}(H)$ by detecting all pairs $K_1, K_2 \in \mathcal{O}(H)$ such that $K_1 \leqslant_{ff} K_2$ and deleting $K_2$.
- The resulting set is $\mathcal{AE}(H)$. $\square$

But ...

$\longrightarrow$ there are exponentially many partitions $\sim$

$\longrightarrow$ the cleaning process needs exponential time (... by the moment).

# Computing $\mathcal{AE}(H)$

**Corollary**

*$\mathcal{AE}(H)$ is computable.*

**Proof:**

- Compute $\Gamma(H)$,
- Compute $\Gamma(H)/\sim$ for all partitions $\sim$ of $V\Gamma(H)$,
- Compute $\mathcal{O}(H)$,
- Clean $\mathcal{O}(H)$ by detecting all pairs $K_1, K_2 \in \mathcal{O}(H)$ such that $K_1 \leqslant_{ff} K_2$ and deleting $K_2$.
- The resulting set is $\mathcal{AE}(H)$. $\square$

But ...

$\longrightarrow$ there are exponentially many partitions $\sim$

$\longrightarrow$ the cleaning process needs exponential time (... by the moment).

# Computing $\mathcal{AE}(H)$

## Corollary

*$\mathcal{AE}(H)$ is computable.*

**Proof:**

- Compute $\Gamma(H)$,
- Compute $\Gamma(H)/\sim$ for all partitions $\sim$ of $V\Gamma(H)$,
- Compute $\mathcal{O}(H)$,
- Clean $\mathcal{O}(H)$ by detecting all pairs $K_1, K_2 \in \mathcal{O}(H)$ such that $K_1 \leqslant_{ff} K_2$ and deleting $K_2$.
- The resulting set is $\mathcal{AE}(H)$. $\square$

But ...

$\longrightarrow$ there are exponentially many partitions $\sim$

$\longrightarrow$ the cleaning process needs exponential time (... by the moment).

# Computing $\mathcal{AE}(H)$

## Corollary

*$\mathcal{AE}(H)$ is computable.*

**Proof:**

- Compute $\Gamma(H)$,
- Compute $\Gamma(H)/\sim$ for all partitions $\sim$ of $V\Gamma(H)$,
- Compute $\mathcal{O}(H)$,
- Clean $\mathcal{O}(H)$ by detecting all pairs $K_1, K_2 \in \mathcal{O}(H)$ such that $K_1 \leqslant_{ff} K_2$ and deleting $K_2$.
- The resulting set is $\mathcal{AE}(H)$. $\square$

But ...

$\longrightarrow$ there are exponentially many partitions $\sim$

$\longrightarrow$ the cleaning process needs exponential time (... by the moment).

# Computing $\mathcal{AE}(H)$

## Corollary

*$\mathcal{AE}(H)$ is computable.*

**Proof:**

- Compute $\Gamma(H)$,
- Compute $\Gamma(H)/\sim$ for all partitions $\sim$ of $V\Gamma(H)$,
- Compute $\mathcal{O}(H)$,
- Clean $\mathcal{O}(H)$ by detecting all pairs $K_1, K_2 \in \mathcal{O}(H)$ such that $K_1 \leqslant_{ff} K_2$ and deleting $K_2$.
- The resulting set is $\mathcal{AE}(H)$. $\square$

But ...

$\longrightarrow$ there are exponentially many partitions $\sim$

$\longrightarrow$ the cleaning process needs exponential time (... by the moment).

## Corollary

*$\mathcal{AE}(H)$ is computable.*

**Proof:**

- Compute $\Gamma(H)$,
- Compute $\Gamma(H)/\sim$ for all partitions $\sim$ of $V\Gamma(H)$,
- Compute $\mathcal{O}(H)$,
- Clean $\mathcal{O}(H)$ by detecting all pairs $K_1, K_2 \in \mathcal{O}(H)$ such that $K_1 \leqslant_{ff} K_2$ and deleting $K_2$.
- The resulting set is $\mathcal{AE}(H)$. $\square$

But ...

$\longrightarrow$ there are exponentially many partitions $\sim$

$\longrightarrow$ the cleaning process needs exponential time (... by the moment).

# Computing $\mathcal{AE}(H)$

## Corollary

*$\mathcal{AE}(H)$ is computable.*

**Proof:**

- Compute $\Gamma(H)$,
- Compute $\Gamma(H)/\sim$ for all partitions $\sim$ of $V\Gamma(H)$,
- Compute $\mathcal{O}(H)$,
- Clean $\mathcal{O}(H)$ by detecting all pairs $K_1, K_2 \in \mathcal{O}(H)$ such that $K_1 \leqslant_{ff} K_2$ and deleting $K_2$.
- The resulting set is $\mathcal{AE}(H)$. $\square$

But ...

$\longrightarrow$ there are exponentially many partitions $\sim$

$\longrightarrow$ the cleaning process needs exponential time (... by the moment).

## Corollary

*$\mathcal{AE}(H)$ is computable.*

**Proof:**

- Compute $\Gamma(H)$,
- Compute $\Gamma(H)/\sim$ for all partitions $\sim$ of $V\Gamma(H)$,
- Compute $\mathcal{O}(H)$,
- Clean $\mathcal{O}(H)$ by detecting all pairs $K_1, K_2 \in \mathcal{O}(H)$ such that $K_1 \leqslant_{ff} K_2$ and deleting $K_2$.
- The resulting set is $\mathcal{AE}(H)$. $\square$

But ...

$\longrightarrow$ there are exponentially many partitions $\sim$

$\longrightarrow$ the cleaning process needs exponential time (... by the moment).

# Outline

# The algebraic closure

**Observation**

*If $H \leqslant_{alg} K_1$ and $H \leqslant_{alg} K_2$ then $H \leqslant_{alg} \langle K_1 \cup K_2 \rangle$.*

**Corollary**

*For every $H \leqslant K \leqslant F_A$ (all f.g.), $\mathcal{AE}_K(H)$ has a unique maximal element, called the $K$-algebraic closure of $H$, and denoted $Cl_K(H)$.*

**Corollary**

*Every extension $H \leqslant K$ of f.g. subgroups of $F_A$ splits, in a unique way, in an algebraic part and a free part, $H \leqslant_{alg} Cl_K(H) \leqslant_{ff} K$.*

Compare with Hall's property.

# The algebraic closure

## Observation

*If $H \leqslant_{alg} K_1$ and $H \leqslant_{alg} K_2$ then $H \leqslant_{alg} \langle K_1 \cup K_2 \rangle$.*

## Corollary

*For every $H \leqslant K \leqslant F_A$ (all f.g.), $\mathcal{AE}_K(H)$ has a unique maximal element, called the $K$-algebraic closure of $H$, and denoted $Cl_K(H)$.*

## Corollary

*Every extension $H \leqslant K$ of f.g. subgroups of $F_A$ splits, in a unique way, in an algebraic part and a free part, $H \leqslant_{alg} Cl_K(H) \leqslant_{ff} K$.*

Compare with Hall's property.

# The algebraic closure

## Observation

*If $H \leqslant_{alg} K_1$ and $H \leqslant_{alg} K_2$ then $H \leqslant_{alg} \langle K_1 \cup K_2 \rangle$.*

## Corollary

*For every $H \leqslant K \leqslant F_A$ (all f.g.), $\mathcal{AE}_K(H)$ has a unique maximal element, called the $K$-algebraic closure of $H$, and denoted $Cl_K(H)$.*

## Corollary

*Every extension $H \leqslant K$ of f.g. subgroups of $F_A$ splits, in a unique way, in an algebraic part and a free part, $H \leqslant_{alg} Cl_K(H) \leqslant_{ff} K$.*

Compare with Hall's property.

# The algebraic closure

## Observation

*If $H \leqslant_{alg} K_1$ and $H \leqslant_{alg} K_2$ then $H \leqslant_{alg} \langle K_1 \cup K_2 \rangle$.*

## Corollary

*For every $H \leqslant K \leqslant F_A$ (all f.g.), $\mathcal{AE}_K(H)$ has a unique maximal element, called the $K$-algebraic closure of $H$, and denoted $Cl_K(H)$.*

## Corollary

*Every extension $H \leqslant K$ of f.g. subgroups of $F_A$ splits, in a unique way, in an algebraic part and a free part, $H \leqslant_{alg} Cl_K(H) \leqslant_{ff} K$.*

Compare with Hall's property.

## Computing algebraic closures

In the rest of the talk we'll sketch the proof of:

**Theorem (V. 2009)**

*Given $H \leqslant K \leqslant F_A$ (all f.g.) one can compute (a basis for) $Cl_K(H)$ in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

Main ingredients in the proof:

1) Construct directly $Cl_K(H)$ without having to compute all of $\mathcal{O}(H)$.
2) Use

**Theorem (Roig-V.-Weil, 2007)**

*Given $H, K \leqslant F_A$ (all f.g.), one can decide whether $H \leqslant_{ff} K$ or not, in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

- Whitehead 1930's (classical and exponential),
- Silva-Weil 2006 (graphical algorithm, faster but still exponential),
- Roig-V.-Weil 2007 (variation of Whitehead algorithm in polynomial time).

# Computing algebraic closures

In the rest of the talk we'll sketch the proof of:

**Theorem (V. 2009)**

*Given $H \leqslant K \leqslant F_A$ (all f.g.) one can compute (a basis for) $Cl_K(H)$ in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

Main ingredients in the proof:

1) Construct directly $Cl_K(H)$ without having to compute all of $\mathcal{O}(H)$.
2) Use

**Theorem (Roig-V.-Weil, 2007)**

*Given $H, K \leqslant F_A$ (all f.g.), one can decide whether $H \leqslant_{ff} K$ or not, in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

- Whitehead 1930's (classical and exponential),
- Silva-Weil 2006 (graphical algorithm, faster but still exponential),
- Roig-V.-Weil 2007 (variation of Whitehead algorithm in polynomial time).

# Computing algebraic closures

In the rest of the talk we'll sketch the proof of:

**Theorem (V. 2009)**

*Given $H \leqslant K \leqslant F_A$ (all f.g.) one can compute (a basis for) $Cl_K(H)$ in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

Main ingredients in the proof:

1) Construct directly $Cl_K(H)$ without having to compute all of $\mathcal{O}(H)$.
2) Use

**Theorem (Roig-V.-Weil, 2007)**

*Given $H, K \leqslant F_A$ (all f.g.), one can decide whether $H \leqslant_{ff} K$ or not, in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

- Whitehead 1930's (classical and exponential),
- Silva-Weil 2006 (graphical algorithm, faster but still exponential),
- Roig-V.-Weil 2007 (variation of Whitehead algorithm in polynomial time).

# Computing algebraic closures

In the rest of the talk we'll sketch the proof of:

**Theorem (V. 2009)**

*Given $H \leqslant K \leqslant F_A$ (all f.g.) one can compute (a basis for) $Cl_K(H)$ in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

Main ingredients in the proof:

1) Construct directly $Cl_K(H)$ without having to compute all of $\mathcal{O}(H)$.
2) Use

**Theorem (Roig-V.-Weil, 2007)**

*Given $H, K \leqslant F_A$ (all f.g.), one can decide whether $H \leqslant_{ff} K$ or not, in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

- Whitehead 1930's (classical and exponential),
- Silva-Weil 2006 (graphical algorithm, faster but still exponential),
- Roig-V.-Weil 2007 (variation of Whitehead algorithm in polynomial time).

# Computing algebraic closures

In the rest of the talk we'll sketch the proof of:

**Theorem (V. 2009)**

*Given $H \leqslant K \leqslant F_A$ (all f.g.) one can compute (a basis for) $Cl_K(H)$ in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

Main ingredients in the proof:

1) Construct directly $Cl_K(H)$ without having to compute all of $\mathcal{O}(H)$.
2) Use

**Theorem (Roig-V.-Weil, 2007)**

*Given $H, K \leqslant F_A$ (all f.g.), one can decide whether $H \leqslant_{ff} K$ or not, in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

- Whitehead 1930's (classical and exponential),
- Silva-Weil 2006 (graphical algorithm, faster but still exponential),
- Roig-V.-Weil 2007 (variation of Whitehead algorithm in polynomial time).

# Computing algebraic closures

In the rest of the talk we'll sketch the proof of:

**Theorem (V. 2009)**

*Given $H \leqslant K \leqslant F_A$ (all f.g.) one can compute (a basis for) $Cl_K(H)$ in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

Main ingredients in the proof:

1) Construct directly $Cl_K(H)$ without having to compute all of $\mathcal{O}(H)$.
2) Use

**Theorem (Roig-V.-Weil, 2007)**

*Given $H, K \leqslant F_A$ (all f.g.), one can decide whether $H \leqslant_{ff} K$ or not, in polynomial time w.r.t. the sum of lengths of given generators for $H$ and $K$.*

- Whitehead 1930's (classical and exponential),
- Silva-Weil 2006 (graphical algorithm, faster but still exponential),
- Roig-V.-Weil 2007 (variation of Whitehead algorithm in polynomial time).

# Outline

# Whitehead problem:

## Whitehead Problem

*For a group $G$, find an algorithm s.t. given $u, v \in G$ decides whether there exists $\varphi \in Aut(G)$ such that $\varphi(u) = v$.*

## Theorem (Whitehead)

*Whitehead problem is solvable in $F(A)$.*

**“Proof":**

**First part:** reduce $\|u\|$ and $\|v\|$ as much as possible by applying autos:

$$u \to u_1 \to u_2 \to \cdots \to u',$$

$$v \to v_1 \to v_2 \to \cdots \to v'.$$

**Second part:** analyze who is image of who by some auto, in the (finite!) sphere of given radius $n$, $S_n = \{w \in F_k \mid \|w\| = n\}$. $\square$

# Whitehead problem:

## Whitehead Problem

*For a group $G$, find an algorithm s.t. given $u, v \in G$ decides whether there exists $\varphi \in Aut(G)$ such that $\varphi(u) = v$.*

## Theorem (Whitehead)

*Whitehead problem is solvable in $F(A)$.*

**"Proof":**
**First part:** reduce $\|u\|$ and $\|v\|$ as much as possible by applying autos:

$$u \to u_1 \to u_2 \to \cdots \to u',$$

$$v \to v_1 \to v_2 \to \cdots \to v'.$$

**Second part:** analyze who is image of who by some auto, in the (finite!) sphere of given radius $n$, $S_n = \{w \in F_k \mid \|w\| = n\}$. $\square$

# Whitehead problem:

## Whitehead Problem

*For a group $G$, find an algorithm s.t. given $u, v \in G$ decides whether there exists $\varphi \in Aut(G)$ such that $\varphi(u) = v$.*

## Theorem (Whitehead)

*Whitehead problem is solvable in $F(A)$.*

**“Proof":**
**First part:** reduce $\|u\|$ and $\|v\|$ as much as possible by applying autos:

$$u \to u_1 \to u_2 \to \cdots \to u',$$

$$v \to v_1 \to v_2 \to \cdots \to v'.$$

**Second part:** analyze who is image of who by some auto, in the (finite!) sphere of given radius $n$, $S_n = \{w \in F_k \mid \|w\| = n\}$. $\square$

# Whitehead problem:

## Whitehead Problem

*For a group G, find an algorithm s.t. given $u, v \in G$ decides whether there exists $\varphi \in Aut(G)$ such that $\varphi(u) = v$.*

## Theorem (Whitehead)

*Whitehead problem is solvable in $F(A)$.*

**“Proof":**
**First part:** reduce $\|u\|$ and $\|v\|$ as much as possible by applying autos:

$$u \to u_1 \to u_2 \to \cdots \to u',$$

$$v \to v_1 \to v_2 \to \cdots \to v'.$$

**Second part:** analyze who is image of who by some auto, in the (finite!) sphere of given radius $n$, $S_n = \{w \in F_k \mid \|w\| = n\}$. $\Box$

# Whitehead minimization problem

Let us concentrate in the first part:

## Whitehead Minimization Problem (WMP)

*Given $u \in F(A)$, find $\varphi \in Aut(F(A))$ such that $\|\varphi(u)\|$ is minimal.*

## Lemma (Whitehead)

*Let $u \in F(A)$. If $\exists\, \varphi \in Aut(F(A))$ such that $\|\varphi(u)\| < \|u\|$ then $\exists$ a "Whitehead automorphism" $\alpha$ such that $\|\varphi(u)\| < \|u\|$.*

## Definition

*Whitehead automorphisms are those of the form:*

$$\begin{array}{rcll} F(A) & \to & F(A) & \\ a_i & \mapsto & a_i & \quad \text{(the multiplier)} \\ a_i \neq a_j & \mapsto & a_i^{\epsilon_j} a_j\, a_i^{\delta_j} & \end{array}$$

*where $\epsilon_j = 0, -1$ and $\delta_j = 0, 1$ (there are $\sim k \cdot 4^k$ many, where $k = |A|$).*

# Whitehead minimization problem

Let us concentrate in the first part:

## Whitehead Minimization Problem (WMP)

*Given $u \in F(A)$, find $\varphi \in Aut(F(A))$ such that $\|\varphi(u)\|$ is minimal.*

## Lemma (Whitehead)

*Let $u \in F(A)$. If $\exists\, \varphi \in Aut(F(A))$ such that $\|\varphi(u)\| < \|u\|$ then $\exists$ a “Whitehead automorphism" $\alpha$ such that $\|\varphi(u)\| < \|u\|$.*

## Definition

*Whitehead automorphisms are those of the form:*

$$\begin{array}{rcll} F(A) & \to & F(A) & \\ a_i & \mapsto & a_i & \text{(the multiplier)} \\ a_i \neq a_j & \mapsto & a_i^{\epsilon_j} a_j a_i^{\delta_j} & \end{array}$$

*where $\epsilon_j = 0, -1$ and $\delta_j = 0, 1$ (there are $\sim k \cdot 4^k$ many, where $k = |A|$).*

# Whitehead minimization problem

Let us concentrate in the first part:

## Whitehead Minimization Problem (WMP)

*Given $u \in F(A)$, find $\varphi \in Aut(F(A))$ such that $\|\varphi(u)\|$ is minimal.*

## Lemma (Whitehead)

*Let $u \in F(A)$. If $\exists\, \varphi \in Aut(F(A))$ such that $\|\varphi(u)\| < \|u\|$ then $\exists$ a "Whitehead automorphism" $\alpha$ such that $\|\varphi(u)\| < \|u\|$.*

## Definition

*Whitehead automorphisms are those of the form:*

$$\begin{array}{rcll} F(A) & \to & F(A) & \\ a_i & \mapsto & a_i & \qquad \text{(the multiplier)} \\ a_i \neq a_j & \mapsto & a_i^{\epsilon_j} a_j\, a_i^{\delta_j} & \end{array}$$

*where $\epsilon_j = 0, -1$ and $\delta_j = 0, 1$ (there are $\sim k \cdot 4^k$ many, where $k = |A|$).*

# Classical Whitehead's algorithm (first part)

Classical whitehead algorithm is

- *Keep applying whitehead automorphisms to given u until finding one that decreases its cyclic length.*
- *Repeat until all whiteheads are non-decreasing.*

This is polynomial on $\|u\|$, but exponential on the ambient rank, $k$.

There are several recent results (theoric, heuristic, probabilistic) suggesting that Whitehead algorithm is faster in practice.

# Classical Whitehead's algorithm (first part)

Classical whitehead algorithm is

- *Keep applying whitehead automorphisms to given u until finding one that decreases its cyclic length.*
- *Repeat until all whiteheads are non-decreasing.*

This is polynomial on $||u||$, but exponential on the ambient rank, $k$.

There are several recent results (theoric, heuristic, probabilistic) suggesting that Whitehead algorithm is faster in practice.

# Classical Whitehead's algorithm (first part)

Classical whitehead algorithm is

- *Keep applying whitehead automorphisms to given u until finding one that decreases its cyclic length.*
- *Repeat until all whiteheads are non-decreasing.*

This is polynomial on $\|u\|$, but exponential on the ambient rank, $k$.

There are several recent results (theoric, heuristic, probabilistic) suggesting that Whitehead algorithm is faster in practice.

# Classical Whitehead's algorithm (first part)

Classical whitehead algorithm is

- *Keep applying whitehead automorphisms to given u until finding one that decreases its cyclic length.*
- *Repeat until all whiteheads are non-decreasing.*

This is polynomial on $\|u\|$, but exponential on the ambient rank, $k$.

There are several recent results (theoric, heuristic, probabilistic) suggesting that Whitehead algorithm is faster in practice.

# Improvement

**Theorem (Roig, V., Weil, 2007)**

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**main idea:** given $u \in F_k$, we find in polynomial time one of the whiteheads that decreases $\|u\|$ the most possible.

**Key point:** How does a given Whitehead automorphism $\alpha$ affect the length of a given word $u$ ?

**Three ingredients:**

1) Codify $u$ as its Whitehead's graph (classic in Group Theory),
2) Codify $\alpha$ as a cut in this graph ($\approx$ classic in Group Theory),
3) Use max-flow min-cut algorithm (classic in Computer Science),
4) ... put together and mix ( new! ).

# Improvement

**Theorem (Roig, V., Weil, 2007)**

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**main idea:** given $u \in F_k$, we find in polynomial time one of the whiteheads that decreases $\|u\|$ the most possible.

**Key point:** How does a given Whitehead automorphism $\alpha$ affect the length of a given word $u$ ?

**Three ingredients:**

1) Codify $u$ as its Whitehead's graph (classic in Group Theory),
2) Codify $\alpha$ as a cut in this graph ($\approx$ classic in Group Theory),
3) Use max-flow min-cut algorithm (classic in Computer Science),
4) ... put together and mix ( new! ).

# Improvement

**Theorem (Roig, V., Weil, 2007)**

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**main idea:** given $u \in F_k$, we find in polynomial time one of the whiteheads that decreases $\|u\|$ the most possible.

**Key point:** How does a given Whitehead automorphism $\alpha$ affect the length of a given word $u$ ?

**Three ingredients:**

1) Codify $u$ as its Whitehead's graph (classic in Group Theory),
2) Codify $\alpha$ as a cut in this graph ($\approx$ classic in Group Theory),
3) Use max-flow min-cut algorithm (classic in Computer Science),
4) ... put together and mix ( new! ).

# Improvement

**Theorem (Roig, V., Weil, 2007)**

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**main idea:** given $u \in F_k$, we find in polynomial time one of the whiteheads that decreases $\|u\|$ the most possible.

**Key point:** How does a given Whitehead automorphism $\alpha$ affect the length of a given word $u$ ?

**Three ingredients:**

1) Codify $u$ as its Whitehead's graph (classic in Group Theory),
2) Codify $\alpha$ as a cut in this graph ($\approx$ classic in Group Theory),
3) Use max-flow min-cut algorithm (classic in Computer Science),
4) ... put together and mix ( new! ).

# Improvement

## Theorem (Roig, V., Weil, 2007)

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**main idea:** given $u \in F_k$, we find in polynomial time one of the whiteheads that decreases $\|u\|$ the most possible.

**Key point:** How does a given Whitehead automorphism $\alpha$ affect the length of a given word $u$ ?

**Three ingredients:**

1) Codify $u$ as its Whitehead's graph (classic in Group Theory),
2) Codify $\alpha$ as a cut in this graph ($\approx$ classic in Group Theory),
3) Use max-flow min-cut algorithm (classic in Computer Science),
4) ... put together and mix ( new! ).

# Improvement

**Theorem (Roig, V., Weil, 2007)**

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**main idea:** given $u \in F_k$, we find in polynomial time one of the whiteheads that decreases $\|u\|$ the most possible.

**Key point:** How does a given Whitehead automorphism $\alpha$ affect the length of a given word $u$ ?

**Three ingredients:**

1) Codify $u$ as its Whitehead's graph (classic in Group Theory),
2) Codify $\alpha$ as a cut in this graph ($\approx$ classic in Group Theory),
3) Use max-flow min-cut algorithm (classic in Computer Science),
4) ... put together and mix ( new! ).

# Improvement

**Theorem (Roig, V., Weil, 2007)**

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**main idea:** given $u \in F_k$, we find in polynomial time one of the whiteheads that decreases $\|u\|$ the most possible.

**Key point:** How does a given Whitehead automorphism $\alpha$ affect the length of a given word $u$ ?

**Three ingredients:**

1) Codify $u$ as its Whitehead's graph (classic in Group Theory),
2) Codify $\alpha$ as a cut in this graph ($\approx$ classic in Group Theory),
3) Use max-flow min-cut algorithm (classic in Computer Science),
4) ... put together and mix ( new! ).

# Whitehead's graph

First ingredient: Whitehead's graph of a word.

## Definition

*Given $u \in F_k$ (cyclically reduced), its (unoriented) Whitehead graph, denoted $Wh(u)$, is:*

- *vertices: $A^{\pm 1}$,*
- *edges: for every pair of (cycl.) consecutive letters $u = \cdots xy \cdots$ put an edge between $x$ and $y^{-1}$.*

## Example

$u = aba^{-1}c^{-1}bbabc^{-1}$,

# Whitehead's graph

First ingredient: Whitehead's graph of a word.

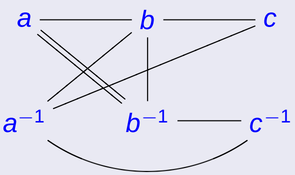
## Definition

*Given $u \in F_k$ (cyclically reduced), its (unoriented) Whitehead graph, denoted $Wh(u)$, is:*

- *vertices: $A^{\pm 1}$,*
- *edges: for every pair of (cycl.) consecutive letters $u = \cdots xy \cdots$ put an edge between $x$ and $y^{-1}$.*

## Example

$$u = aba^{-1}c^{-1}bbabc^{-1},$$



# Cut in a graph

Second ingredient: Cut in a graph.

## Definition

*Given a Whitehead's automorphism $\alpha$, we represent it as the $(a, a^{-1})$-cut*

$$(T = \{a\} \cup \{\textit{letters that go multiplied on the right by } a\}, a)$$

*of the set $A^{\pm 1}$.*

## Example

$$\begin{array}{rcl} \langle a, b, c \rangle = F_3 & \to & F_3 \\ a & \mapsto & ab \\ b & \mapsto & b \\ c & \mapsto & b^{-1}cb \end{array}$$

$a \qquad b \qquad c$

$a^{-1} \qquad b^{-1} \qquad c^{-1}$

# Cut in a graph

Second ingredient: Cut in a graph.

## Definition

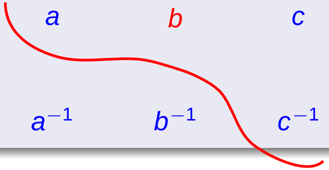
*Given a Whitehead's automorphism $\alpha$, we represent it as the $(a, a^{-1})$-cut*

$$(T = \{a\} \cup \{\text{letters that go multiplied on the right by } a\}, a)$$

*of the set $A^{\pm 1}$.*

## Example

$$\begin{array}{rcl} \langle a,b,c\rangle = F_3 & \to & F_3 \\ a & \mapsto & ab \\ b & \mapsto & b \\ c & \mapsto & b^{-1}cb \end{array}$$



# Rephrasement of Wh. Lemma

## Lemma (Whitehead)

*Given a word $u \in F_k$ and a Whitehead automorphism $\alpha$, think $\alpha$ as a cut in $Wh(u)$, say $\alpha = (T, a)$, and then*

$$\|\alpha(u)\| - \|u\| = \mathrm{cap}(T) - \deg(a).$$

**Proof:** Analyzing combinatorial cases (see Lyndon-Schupp).

# Rephrasement of Wh. Lemma

## Lemma (Whitehead)

*Given a word $u \in F_k$ and a Whitehead automorphism $\alpha$, think $\alpha$ as a cut in $Wh(u)$, say $\alpha = (T, a)$, and then*

$$\|\alpha(u)\| - \|u\| = \mathrm{cap}(T) - \deg(a).$$

**Proof:** Analyzing combinatorial cases (see Lyndon-Schupp).

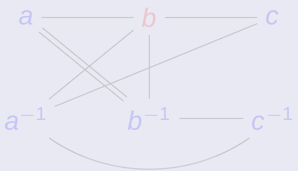
# Example

## Example

*Consider* $u = aba^{-1}c^{-1}bbabc^{-1}$ *and*

$$\begin{array}{rcl} \alpha\colon F_3 & \to & F_3 \\ a & \mapsto & ab \\ b & \mapsto & b \\ c & \mapsto & b^{-1}cb \end{array}$$

*like before. We have* $\alpha(u) = aba^{-1}b^{-1}c^{-1}bbbabc^{-1}b$. *Furthermore,*

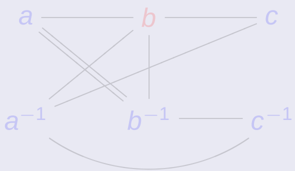



*and, in fact,*

$$12 - 9 = \|\alpha(u)\| - \|u\| = \mathrm{cap}(T) - \deg(b) = 7 - 4.$$

## Example

*Consider* $u = aba^{-1}c^{-1}bbabc^{-1}$ *and* $\begin{array}{rcl} \alpha\colon F_3 & \to & F_3 \\ a & \mapsto & ab \\ b & \mapsto & b \\ c & \mapsto & b^{-1}cb \end{array}$ *like before. We have* $\alpha(u) = aba^{-1}b^{-1}c^{-1}bbbabc^{-1}b$. *Furthermore,*



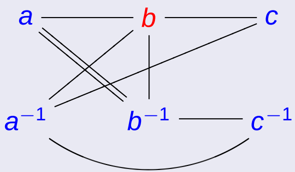

*and, in fact,*

$$12 - 9 = \|\alpha(u)\| - \|u\| = \mathrm{cap}(T) - \deg(b) = 7 - 4.$$

# Example

## Example

*Consider* $u = aba^{-1}c^{-1}bbabc^{-1}$ *and* $\begin{array}{rcl} \alpha\colon F_3 & \to & F_3 \\ a & \mapsto & ab \\ b & \mapsto & b \\ c & \mapsto & b^{-1}cb \end{array}$ *like before. We have* $\alpha(u) = aba^{-1}b^{-1}c^{-1}bbbabc^{-1}b$. *Furthermore,*




*and, in fact,*

$$12 - 9 = \|\alpha(u)\| - \|u\| = \mathrm{cap}(T) - \deg(b) = 7 - 4.$$

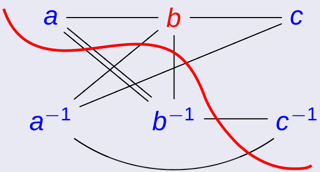
# Example

## Example

*Consider* $u = aba^{-1}c^{-1}bbabc^{-1}$ *and*

$$\begin{array}{rcl} \alpha\colon F_3 & \to & F_3 \\ a & \mapsto & ab \\ b & \mapsto & b \\ c & \mapsto & b^{-1}cb \end{array}$$

*like before. We have* $\alpha(u) = aba^{-1}b^{-1}c^{-1}bbbabc^{-1}b$. *Furthermore,*




*and, in fact,*

$$12 - 9 = \|\alpha(u)\| - \|u\| = \mathrm{cap}(T) - \deg(b) = 7 - 4.$$

# Max-flow min-cut algorithm

Third ingredient: Max-flow min-cut algorithm.

Hence, Whitehead's Minimization Problem reduces to:

- *run over all possible multipliers, say a, (there are $2k$),*
- *find an $(a, a^{-1})$-cut with minimal possible capacity.*

This can be done by using the classical max-flow min-cut algorithm ...

...which works in polynomial time w.r.t. the number of edges of the graph $(= \|u\|)$ and the number of vertices $(= 2k)$.

# Max-flow min-cut algorithm

Third ingredient: Max-flow min-cut algorithm.

Hence, Whitehead's Minimization Problem reduces to:

- *run over all possible multipliers, say a, (there are 2k),*
- *find an* $(a, a^{-1})$*-cut with minimal possible capacity.*

This can be done by using the classical max-flow min-cut algorithm ...

...which works in polynomial time w.r.t. the number of edges of the graph $(= \|u\|)$ and the number of vertices $(= 2k)$.

# Max-flow min-cut algorithm

Third ingredient: Max-flow min-cut algorithm.

Hence, Whitehead's Minimization Problem reduces to:

- *run over all possible multipliers, say a, (there are $2k$),*
- *find an $(a, a^{-1})$-cut with minimal possible capacity.*

This can be done by using the classical max-flow min-cut algorithm ...

...which works in polynomial time w.r.t. the number of edges of the graph $(= \|u\|)$ and the number of vertices $(= 2k)$.

# Max-flow min-cut algorithm

Third ingredient: Max-flow min-cut algorithm.

Hence, Whitehead's Minimization Problem reduces to:

- *run over all possible multipliers, say a, (there are 2k),*
- *find an $(a, a^{-1})$-cut with minimal possible capacity.*

This can be done by using the classical max-flow min-cut algorithm ...

...which works in polynomial time w.r.t. the number of edges of the graph ($= \|u\|$) and the number of vertices ($= 2k$).

Hence we have proved

**Theorem (Roig, V., Weil, 2007)**

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**Corollary (Roig, V., Weil, 2007)**

*Given a word $u \in F_k$, one can check whether $u$ is primitive in $F_k$ in time $O(n^2 k^3)$, where $n = \|u\|$.*

# Primitivity

Hence we have proved

**Theorem (Roig, V., Weil, 2007)**

*There is an algorithm which solves Whitehead Minimization Problem for $F_k$ in time $O(n^2 k^3)$.*

**Corollary (Roig, V., Weil, 2007)**

*Given a word $u \in F_k$, one can check whether $u$ is primitive in $F_k$ in time $O(n^2k^3)$, where $n = \|u\|$.*

# Outline

# Whitehead's hypergraph

A cyclically reduced word can be thought as a circular graph; and then, its Whitehead graph $Wh(u)$ just describes the in-links of the vertices.

**Definition**

*Let $H \leqslant F_k$ be a f.g. subgroup, and let $\Gamma(H)$ be its core graph. We define the Whitehead hyper-graph of $H$, denoted $Wh(H)$, as:*

- *vertices: $A^{\pm 1}$,*
- *hyper-edges: for every vertex $v$ in $\Gamma(H)$, put a hyper-edge consisting on the in-link of $v$.*

**Lemma (Roig, V., Weil, 2007)**

*Given a f.g. subgroup $H \leqslant F_k$ and a Whitehead automorphism $\alpha$, think $\alpha$ as a cut in $Wh(H)$, say $\alpha = (T, a)$, and then*

$$\|\alpha(u)\| - \|u\| = \operatorname{cap}(T) - \deg(a),$$

*where $\|H\|$ is the number of vertices in $\Gamma(H)$.*

# Whitehead's hypergraph

A cyclically reduced word can be thought as a circular graph; and then, its Whitehead graph $Wh(u)$ just describes the in-links of the vertices.

## Definition

*Let $H \leqslant F_k$ be a f.g. subgroup, and let $\Gamma(H)$ be its core graph. We define the Whitehead hyper-graph of $H$, denoted $Wh(H)$, as:*

- *vertices: $A^{\pm 1}$,*
- *hyper-edges: for every vertex $v$ in $\Gamma(H)$, put a hyper-edge consisting on the in-link of $v$.*

## Lemma (Roig, V., Weil, 2007)

*Given a f.g. subgroup $H \leqslant F_k$ and a Whitehead automorphism $\alpha$, think $\alpha$ as a cut in $Wh(H)$, say $\alpha = (T, a)$, and then*

$$\|\alpha(u)\| - \|u\| = \operatorname{cap}(T) - \deg(a),$$

*where $\|H\|$ is the number of vertices in $\Gamma(H)$.*

# Whitehead's hypergraph

A cyclically reduced word can be thought as a circular graph; and then, its Whitehead graph $Wh(u)$ just describes the in-links of the vertices.

## Definition

*Let $H \leqslant F_k$ be a f.g. subgroup, and let $\Gamma(H)$ be its core graph. We define the Whitehead hyper-graph of $H$, denoted $Wh(H)$, as:*

- *vertices: $A^{\pm 1}$,*
- *hyper-edges: for every vertex $v$ in $\Gamma(H)$, put a hyper-edge consisting on the in-link of $v$.*

## Lemma (Roig, V., Weil, 2007)

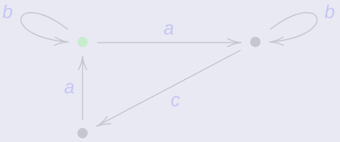
*Given a f.g. subgroup $H \leqslant F_k$ and a Whitehead automorphism $\alpha$, think $\alpha$ as a cut in $Wh(H)$, say $\alpha = (T, a)$, and then*

$$\|\alpha(u)\| - \|u\| = \operatorname{cap}(T) - \deg(a),$$

*where $\|H\|$ is the number of vertices in $\Gamma(H)$.*

# Whitehead's hypergraph

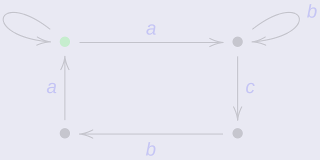
*Consider* $H = \langle b, aba^{-1}, aca \rangle \leqslant F_3$. *Its core graph* $\Gamma(H)$, *and Whitehead hyper-graph* $Wh(H)$ *are:*







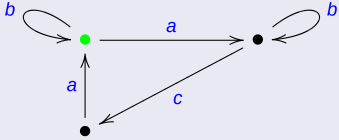
*In fact,* $\alpha(H) = \langle b, aba^{-1}, acbab \rangle$ *and then*

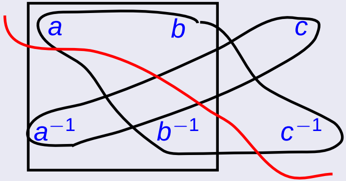
$\Gamma(\alpha(H)) =$ 


*and so,* $4 - 3 = \|\alpha(H)\| - \|H\| = 3 - 2.$

# Whitehead's hypergraph

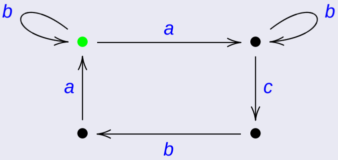
*Consider* $H = \langle b, aba^{-1}, aca \rangle \leqslant F_3$. *Its core graph* $\Gamma(H)$, *and Whitehead hyper-graph* $Wh(H)$ *are:*




*In fact,* $\alpha(H) = \langle b, aba^{-1}, acbab \rangle$ *and then*

$\Gamma(\alpha(H)) =$




*and so,* $4 - 3 = \|\alpha(H)\| - \|H\| = 3 - 2$.

# Minimizing capacities in hyper-graphs

So, Whitehead's Minimization Problem for subgroups reduces to:

- *run over all possible multipliers, say $a$, (there are $2k$),*
- *find an $(a, a^{-1})$-cut with minimal possible capacity in the given hyper-graph.*

Unfortunately, there is no analog of max-flow min-cut algorithm for hyper-graphs ...

...but it is still possible to find minimal cuts in polynomial time because of sub-modularity:

**Observation**

*For every f.g. $H \leqslant F_k$, let $W = Wh(H)$ and then the map $\mathcal{P}(A^{\pm 1}) \to \mathbb{N}$, $T \mapsto \operatorname{cap}_W(T)$ is sub-modular.*

# Minimizing capacities in hyper-graphs

So, Whitehead's Minimization Problem for subgroups reduces to:

- *run over all possible multipliers, say a, (there are 2k),*
- *find an $(a, a^{-1})$-cut with minimal possible capacity in the given hyper-graph.*

Unfortunately, there is no analog of max-flow min-cut algorithm for hyper-graphs ...

...but it is still possible to find minimal cuts in polynomial time because of sub-modularity:

**Observation**

*For every f.g. $H \leqslant F_k$, let $W = Wh(H)$ and then the map $\mathcal{P}(A^{\pm 1}) \to \mathbb{N}$, $T \mapsto \mathrm{cap}_W(T)$ is sub-modular.*

# Minimizing capacities in hyper-graphs

So, Whitehead's Minimization Problem for subgroups reduces to:

- *run over all possible multipliers, say a, (there are $2k$),*
- *find an $(a, a^{-1})$-cut with minimal possible capacity in the given hyper-graph.*

Unfortunately, there is no analog of max-flow min-cut algorithm for hyper-graphs ...

...but it is still possible to find minimal cuts in polynomial time because of sub-modularity:

**Observation**

*For every f.g. $H \leqslant F_k$, let $W = Wh(H)$ and then the map $\mathcal{P}(A^{\pm 1}) \to \mathbb{N}$, $T \mapsto \mathrm{cap}_W(T)$ is sub-modular.*

# Minimizing capacities in hyper-graphs

So, Whitehead's Minimization Problem for subgroups reduces to:

- *run over all possible multipliers, say a, (there are $2k$),*
- *find an $(a, a^{-1})$-cut with minimal possible capacity in the given hyper-graph.*

Unfortunately, there is no analog of max-flow min-cut algorithm for hyper-graphs ...

...but it is still possible to find minimal cuts in polynomial time because of sub-modularity:

**Observation**

*For every f.g. $H \leqslant F_k$, let $W = Wh(H)$ and then the map $\mathcal{P}(A^{\pm 1}) \to \mathbb{N}$, $T \mapsto \mathrm{cap}_W(T)$ is sub-modular.*

# Minimizing capacities in hyper-graphs

So, Whitehead's Minimization Problem for subgroups reduces to:

- *run over all possible multipliers, say a, (there are 2k),*
- *find an $(a, a^{-1})$-cut with minimal possible capacity in the given hyper-graph.*

Unfortunately, there is no analog of max-flow min-cut algorithm for hyper-graphs ...

...but it is still possible to find minimal cuts in polynomial time because of sub-modularity:

**Observation**

*For every f.g. $H \leqslant F_k$, let $W = Wh(H)$ and then the map $\mathcal{P}(A^{\pm 1}) \to \mathbb{N}$, $T \mapsto \mathrm{cap}_W(T)$ is sub-modular.*

# Sub-modularity

## Definition

*A map $f : \mathcal{P}(V) \to \mathbb{N}$ is called* **sub-modular** *if, for every $A, B \subseteq V$, $f(A \cup B) + f(A \cap B) \leqslant f(A) + f(B)$.*

Efficient minimization of sub-modular functions is an active research topic in computer science. One of the classical results is the following

## Proposition

*There exists a algorithm which, given a sub-modular function $f : \mathcal{P}(V) \to \mathbb{N}$ computes its minimum with a number of queries to evaluate $f$ bounded above by a polynomial on $|V|$.*

## Corollary

*There is an algorithm which solves Whitehead Minimization Problem for subgroups $H \leqslant F_k$, in time $O((n^2k^4 + n^3k^2)\log(nk))$, where $n = \|H\|$.*

# Sub-modularity

## Definition

*A map $f : \mathcal{P}(V) \to \mathbb{N}$ is called sub-modular if, for every $A, B \subseteq V$, $f(A \cup B) + f(A \cap B) \leqslant f(A) + f(B)$.*

Efficient minimization of sub-modular functions is an active research topic in computer science. One of the classical results is the following

## Proposition

*There exists a algorithm which, given a sub-modular function $f : \mathcal{P}(V) \to \mathbb{N}$ computes its minimum with a number of queries to evaluate $f$ bounded above by a polynomial on $|V|$.*

## Corollary

*There is an algorithm which solves Whitehead Minimization Problem for subgroups $H \leqslant F_k$, in time $O((n^2k^4 + n^3k^2)\log(nk))$, where $n = \|H\|$.*

# Sub-modularity

## Definition

*A map $f : \mathcal{P}(V) \to \mathbb{N}$ is called sub-modular if, for every $A, B \subseteq V$, $f(A \cup B) + f(A \cap B) \leqslant f(A) + f(B)$.*

Efficient minimization of sub-modular functions is an active research topic in computer science. One of the classical results is the following

## Proposition

*There exists a algorithm which, given a sub-modular function $f : \mathcal{P}(V) \to \mathbb{N}$ computes its minimum with a number of queries to evaluate $f$ bounded above by a polynomial on $|V|$.*

## Corollary

*There is an algorithm which solves Whitehead Minimization Problem for subgroups $H \leqslant F_k$, in time $O((n^2k^4 + n^3k^2)\log(nk))$, where $n = \|H\|$.*

# Sub-modularity

## Definition

*A map $f : \mathcal{P}(V) \to \mathbb{N}$ is called sub-modular if, for every $A, B \subseteq V$, $f(A \cup B) + f(A \cap B) \leqslant f(A) + f(B)$.*

Efficient minimization of sub-modular functions is an active research topic in computer science. One of the classical results is the following

## Proposition

*There exists a algorithm which, given a sub-modular function $f : \mathcal{P}(V) \to \mathbb{N}$ computes its minimum with a number of queries to evaluate $f$ bounded above by a polynomial on $|V|$.*

## Corollary

*There is an algorithm which solves Whitehead Minimization Problem for subgroups $H \leqslant F_k$, in time $O((n^2k^4 + n^3k^2)\log(nk))$, where $n = \|H\|$.*

# Deciding free-factorness

## Observation

*A given subgroup $H \leqslant F_k$ of rank $r(H) = r \leqslant k$ is a free factor of $F_k$ if and only if $\exists\, \varphi \in Aut(F_k)$ such that $\|\varphi(H)\| = 1$.*

## Corollary (Roig, V., Weil, 2007)

*Given a f.g. subgroup $H \leqslant F_k$, one can check whether $H$ is a free factor of $F_k$ in time $O((n^2k^4 + n^3k^2)\log(nk))$, where $n = \|H\|$.*

## Corollary (Roig, V., Weil, 2007)

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can check whether $H$ is a free factor of $K$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

# Deciding free-factorness

## Observation

*A given subgroup $H \leqslant F_k$ of rank $r(H) = r \leqslant k$ is a free factor of $F_k$ if and only if $\exists\, \varphi \in Aut(F_k)$ such that $\|\varphi(H)\| = 1$.*

## Corollary (Roig, V., Weil, 2007)

*Given a f.g. subgroup $H \leqslant F_k$, one can check whether $H$ is a free factor of $F_k$ in time $O((n^2k^4 + n^3k^2)\log(nk))$, where $n = \|H\|$.*

## Corollary (Roig, V., Weil, 2007)

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can check whether $H$ is a free factor of $K$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

# Deciding free-factorness

## Observation

*A given subgroup $H \leqslant F_k$ of rank $r(H) = r \leqslant k$ is a free factor of $F_k$ if and only if $\exists\, \varphi \in Aut(F_k)$ such that $\|\varphi(H)\| = 1$.*

## Corollary (Roig, V., Weil, 2007)

*Given a f.g. subgroup $H \leqslant F_k$, one can check whether $H$ is a free factor of $F_k$ in time $O((n^2k^4 + n^3k^2)\log(nk))$, where $n = \|H\|$.*

## Corollary (Roig, V., Weil, 2007)

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can check whether $H$ is a free factor of $K$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

# Outline

# Computing algebraic closures in polynomial time

**Theorem (V. 2009)**

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can compute the $K$-algebraic closure $Cl_K(H)$ of $H$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

**Proof:**

- Find bases for $H$, and for $K$ (say $\{x_1, \ldots, x_r\}$),
- write $H$ in terms of $\{x_1, \ldots, x_r\}$,
- compute $H_{min}$ and $\varphi \in Aut(K)$ such that $\varphi(H) = H_{min}$, using WMP relative to $K$,
- consider the smallest set of letters $X_0 \subseteq \{x_1, \ldots, x_r\}$ such that $H_{min} \leqslant \langle X_0 \rangle$;
- now, $Cl_K(H) = \varphi^{-1}(\langle X_0 \rangle)$. $\square$

# Computing algebraic closures in polynomial time

**Theorem (V. 2009)**

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can compute the $K$-algebraic closure $Cl_K(H)$ of $H$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

**Proof:**

- Find bases for $H$, and for $K$ (say $\{x_1, \ldots, x_r\}$),
- write $H$ in terms of $\{x_1, \ldots, x_r\}$,
- compute $H_{min}$ and $\varphi \in Aut(K)$ such that $\varphi(H) = H_{min}$, using WMP relative to $K$,
- consider the smallest set of letters $X_0 \subseteq \{x_1, \ldots, x_r\}$ such that $H_{min} \leqslant \langle X_0 \rangle$;
- now, $Cl_K(H) = \varphi^{-1}(\langle X_0 \rangle)$. $\square$

# Computing algebraic closures in polynomial time

## Theorem (V. 2009)

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can compute the $K$-algebraic closure $Cl_K(H)$ of $H$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

**Proof:**

- Find bases for $H$, and for $K$ (say $\{x_1, \ldots, x_r\}$),
- write $H$ in terms of $\{x_1, \ldots, x_r\}$,
- compute $H_{min}$ and $\varphi \in Aut(K)$ such that $\varphi(H) = H_{min}$, using WMP relative to $K$,
- consider the smallest set of letters $X_0 \subseteq \{x_1, \ldots, x_r\}$ such that $H_{min} \leqslant \langle X_0 \rangle$;
- now, $Cl_K(H) = \varphi^{-1}(\langle X_0 \rangle)$. $\square$

# Computing algebraic closures in polynomial time

**Theorem (V. 2009)**

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can compute the $K$-algebraic closure $Cl_K(H)$ of $H$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

**Proof:**

- Find bases for $H$, and for $K$ (say $\{x_1, \ldots, x_r\}$),
- write $H$ in terms of $\{x_1, \ldots, x_r\}$,
- compute $H_{min}$ and $\varphi \in Aut(K)$ such that $\varphi(H) = H_{min}$, using WMP relative to $K$,
- consider the smallest set of letters $X_0 \subseteq \{x_1, \ldots, x_r\}$ such that $H_{min} \leqslant \langle X_0 \rangle$;
- now, $Cl_K(H) = \varphi^{-1}(\langle X_0 \rangle)$. $\square$

# Computing algebraic closures in polynomial time

**Theorem (V. 2009)**

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can compute the $K$-algebraic closure $Cl_K(H)$ of $H$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

**Proof:**

- Find bases for $H$, and for $K$ (say $\{x_1, \ldots, x_r\}$),
- write $H$ in terms of $\{x_1, \ldots, x_r\}$,
- compute $H_{min}$ and $\varphi \in Aut(K)$ such that $\varphi(H) = H_{min}$, using WMP relative to $K$,
- consider the smallest set of letters $X_0 \subseteq \{x_1, \ldots, x_r\}$ such that $H_{min} \leqslant \langle X_0 \rangle$;
- now, $Cl_K(H) = \varphi^{-1}(\langle X_0 \rangle)$. $\square$

# Computing algebraic closures in polynomial time

**Theorem (V. 2009)**

*Given f.g. subgroups $H \leqslant K \leqslant F_k$, one can compute the $K$-algebraic closure $Cl_K(H)$ of $H$ in polynomial time w.r.t. the given generators of $H$ and $K$.*

**Proof:**

- Find bases for $H$, and for $K$ (say $\{x_1, \ldots, x_r\}$),
- write $H$ in terms of $\{x_1, \ldots, x_r\}$,
- compute $H_{min}$ and $\varphi \in Aut(K)$ such that $\varphi(H) = H_{min}$, using WMP relative to $K$,
- consider the smallest set of letters $X_0 \subseteq \{x_1, \ldots, x_r\}$ such that $H_{min} \leqslant \langle X_0 \rangle$;
- now, $Cl_K(H) = \varphi^{-1}(\langle X_0 \rangle)$. $\square$

# Computing algebraic closures in polynomial time

Because...

**Proposition (see I.5.4 in Lyndon-Schupp)**

*Let $F$ be a free group with basis $X$, and let $w$ be a word or cyclic word of minimal length (w.r.t. the action of $Aut(F)$). If exactly $n$ letters occur in $w$ then at least $n$ letters will occur in $\varphi(w)$, for every $\varphi \in Aut(F)$.*

And the similar statement is true as well, for subgroups.

Because...

**Proposition (see I.5.4 in Lyndon-Schupp)**

*Let $F$ be a free group with basis $X$, and let $w$ be a word or cyclic word of minimal length (w.r.t. the action of $Aut(F)$). If exactly $n$ letters occur in $w$ then at least $n$ letters will occur in $\varphi(w)$, for every $\varphi \in Aut(F)$.*

And the similar statement is true as well, for subgroups.

THANKS