Short(er) SLPs in group recognition

Max Neunhöffer

University of St Andrews

(joint work with Ákos Seress)

Columbus, 17 March 2008

Constructive recognition

Problem

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- a procedure that, given $M \in \mathcal{G}$,
  - decides, whether or not $M \in G$ and
  - if so, expresses M as an SLP in the M_i.

Constructive recognition

Problem

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- a procedure that, given $M \in \mathcal{G}$,
  - decides, whether or not $M \in G$ and
  - if so, expresses M as an SLP in the M_i.
- The runtime should be bounded from above by a polynomial in n, k and $\log q$.

Constructive recognition

Problem

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- a procedure that, given $M \in \mathcal{G}$,
  - decides, whether or not $M \in G$ and
  - if so, expresses M as an SLP in the M_i.
- The runtime should be bounded from above by a polynomial in n, k and $\log q$.
- A Monte Carlo Algorithmus is enough.

Constructive recognition

Problem

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- a procedure that, given $M \in \mathcal{G}$,
  - decides, whether or not $M \in G$ and
  - if so, expresses M as an SLP in the M_i.
- The runtime should be bounded from above by a polynomial in n, k and $\log q$.
- A Monte Carlo Algorithmus is enough. (Verification!)

Constructive recognition

Problem

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- a procedure that, given $M \in \mathcal{G}$,
  - decides, whether or not $M \in G$ and
  - if so, expresses M as an SLP in the M_i.
- The runtime should be bounded from above by a polynomial in n, k and $\log q$.
- A Monte Carlo Algorithmus is enough. (Verification!)

If this problem is solved, we call

$\langle M_1, \ldots, M_k \rangle$ recognised constructively.

Recursion

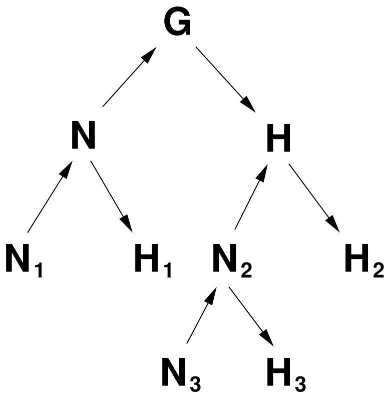
We build a so-called composition tree:

$$\begin{tikzcd}
& \mathbf{G} \arrow[dr] & \\
\mathbf{N} \arrow[ur] & & \mathbf{H}
\end{tikzcd}$$

Up arrows: inclusions
Down arrows: homomorphisms

Recursion

We build a so-called composition tree:



Up arrows: inclusions
Down arrows: homomorphisms

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$$|G| = |H| \cdot |N|.$$

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$|G| = |H| \cdot |N|$. And for $M \in \mathcal{G}$:

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$|G| = |H| \cdot |N|$. And for $M \in \mathcal{G}$:

1. map M with φ onto $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$,

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$|G| = |H| \cdot |N|$. And for $M \in \mathcal{G}$:

1. map M with φ onto $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$,
2. express $\varphi(M) = \mathrm{SLP}_1(P_1, \ldots, P_k)$,

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$$|G| = |H| \cdot |N|. \text{ And for } M \in \mathcal{G}:$$

1. map M with φ onto $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$,
2. express $\varphi(M) = \mathrm{SLP}_1(P_1, \ldots, P_k)$,
3. evaluate the same SLP: $M' := \mathrm{SLP}_1(M_1, \ldots, M_k)$,

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$$|G| = |H| \cdot |N|. \text{ And for } M \in \mathcal{G}:$$

1. map M with φ onto $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$,
2. express $\varphi(M) = \mathrm{SLP}_1(P_1, \ldots, P_k)$,
3. evaluate the same SLP: $M' := \mathrm{SLP}_1(M_1, \ldots, M_k)$,
4. get element $M' \in G$ such that $M \cdot M'^{-1} \in N$,

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$|G| = |H| \cdot |N|$. And for $M \in \mathcal{G}$:

1. map M with φ onto $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$,
2. express $\varphi(M) = \mathrm{SLP}_1(P_1, \ldots, P_k)$,
3. evaluate the same SLP: $M' := \mathrm{SLP}_1(M_1, \ldots, M_k)$,
4. get element $M' \in G$ such that $M \cdot M'^{-1} \in N$,
5. express $M \cdot M'^{-1} = \mathrm{SLP}_2(N_1, \ldots, N_m)$,

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$|G| = |H| \cdot |N|$. And for $M \in \mathcal{G}$:

1. map M with φ onto $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$,
2. express $\varphi(M) = \mathrm{SLP}_1(P_1, \ldots, P_k)$,
3. evaluate the same SLP: $M' := \mathrm{SLP}_1(M_1, \ldots, M_k)$,
4. get element $M' \in G$ such that $M \cdot M'^{-1} \in N$,
5. express $M \cdot M'^{-1} = \mathrm{SLP}_2(N_1, \ldots, N_m)$,
6. get M as SLP in the M_i and N_j:

$$\Rightarrow M = \mathrm{SLP}_2(N_1, \ldots, N_m) \cdot \mathrm{SLP}_1(M_1, \ldots, M_k).$$

Recognising image and kernel suffices

Let $\varphi : G \to H$ be an epimorphism and assume that both H and the kernel $N = \langle N_1, \ldots, N_m \rangle$ of φ are already recognised constructively (assume the N_i are expressed in terms of the M_j).

Then we have recognised G constructively:

$|G| = |H| \cdot |N|$. And for $M \in \mathcal{G}$:

1. map M with φ onto $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$,
2. express $\varphi(M) = \mathrm{SLP}_1(P_1, \ldots, P_k)$,
3. evaluate the same SLP: $M' := \mathrm{SLP}_1(M_1, \ldots, M_k)$,
4. get element $M' \in G$ such that $M \cdot M'^{-1} \in N$,
5. express $M \cdot M'^{-1} = \mathrm{SLP}_2(N_1, \ldots, N_m)$,
6. get M as SLP in the M_i and N_j:

   $$\Rightarrow M = \mathrm{SLP}_2(N_1, \ldots, N_m) \cdot \mathrm{SLP}_1(M_1, \ldots, M_k).$$
7. If $M \notin G$, then at least one step does not work.

Long SLPs

Typical examples:

(1) $G := (2 \times 2^{1+8}) : U_4(2) : 2 < \mathrm{GL}_{78}(2)$

(7th maximal subgroup of the sporadic simple group Fi_{22})

Long SLPs

Typical examples:

(1) $G := (2 \times 2^{1+8}) : U_4(2) : 2 < \mathrm{GL}_{78}(2)$

(7th maximal subgroup of the sporadic simple group Fi_{22})

G has 53 084 160 elements, generated by 2 elements.

Long SLPs

Typical examples:

(1) $$G := (2 \times 2^{1+8}) : U_4(2) : 2 < \mathrm{GL}_{78}(2)$$

(7th maximal subgroup of the sporadic simple group Fi_{22})

G has 53 084 160 elements, generated by 2 elements.

Composition tree of depth 8 with 3 non-trivial leaves.

Long SLPs

Typical examples:

(1) $G := (2 \times 2^{1+8}) : U_4(2) : 2 < \mathrm{GL}_{78}(2)$

(7th maximal subgroup of the sporadic simple group Fi_{22})

G has 53 084 160 elements, generated by 2 elements.

Composition tree of depth 8 with 3 non-trivial leaves.

Typical elements in G give SLPs of length $\approx$ 900.

Long SLPs

Typical examples:

(1) $G := (2 \times 2^{1+8}) : U_4(2) : 2 < \mathrm{GL}_{78}(2)$

(7th maximal subgroup of the sporadic simple group Fi_{22})

G has 53 084 160 elements, generated by 2 elements.

Composition tree of depth 8 with 3 non-trivial leaves.

Typical elements in G give SLPs of length ≈ 900.

(2) $W := S_{12} \wr S_5 < S_{60}$

Long SLPs

Typical examples:

(1) $G := (2 \times 2^{1+8}) : U_4(2) : 2 < \mathrm{GL}_{78}(2)$

(7th maximal subgroup of the sporadic simple group Fi_{22})

G has 53 084 160 elements, generated by 2 elements.

Composition tree of depth 8 with 3 non-trivial leaves.

Typical elements in G give SLPs of length $\approx$ 900.

(2) $W := S_{12} \wr S_5 < S_{60}$

W has 3 025 980 091 991 082 565 958 286 705 898 291 200 000 000 000 elements and is generated by 12 elements.

Long SLPs

Typical examples:

(1) $G := (2 \times 2^{1+8}) : U_4(2) : 2 < \mathrm{GL}_{78}(2)$

(7th maximal subgroup of the sporadic simple group Fi_{22})

G has 53 084 160 elements, generated by 2 elements.

Composition tree of depth 8 with 3 non-trivial leaves.

Typical elements in G give SLPs of length $\approx$ 900.

(2) $W := S_{12} \wr S_5 < S_{60}$

W has 3 025 980 091 991 082 565 958 286 705 898 291 200 000 000 000 elements and is generated by 12 elements.

Composition tree of depth 4 with 6 non-trivial leaves.

Long SLPs

Typical examples:

(1) $G := (2 \times 2^{1+8}) : U_4(2) : 2 < \mathrm{GL}_{78}(2)$

(7th maximal subgroup of the sporadic simple group Fi_{22})

G has 53 084 160 elements, generated by 2 elements.

Composition tree of depth 8 with 3 non-trivial leaves.

Typical elements in G give SLPs of length $\approx$ 900.

(2) $W := S_{12} \wr S_5 < S_{60}$

W has 3 025 980 091 991 082 565 958 286 705 898 291 200 000 000 000 elements and is generated by 12 elements.

Composition tree of depth 4 with 6 non-trivial leaves.

Typical elements in W give SLPs of length $\approx$ 10000.

Learning from base and strong generators

The same groups with stabiliser chains:

$$G := (2 \times 2^{1+8}) : U_4(2) : 2 < S_{3510}$$

(7th maximal subgroup of the sporadic simple group Fi_{22})

Learning from base and strong generators

The same groups with stabiliser chains:

$$G := (2 \times 2^{1+8}) : U_4(2) : 2 < S_{3510}$$

(7th maximal subgroup of the sporadic simple group Fi_{22})

Stabiliser chain of length 4 with 14 strong generators.

Learning from base and strong generators

The same groups with stabiliser chains:

$$G := (2 \times 2^{1+8}) : U_4(2) : 2 < S_{3510}$$

(7th maximal subgroup of the sporadic simple group Fi_{22})

Stabiliser chain of length 4 with 14 strong generators.

Typical elements in G give SLPs of length $\approx$ 15.

Learning from base and strong generators

The same groups with stabiliser chains:

$$G := (2 \times 2^{1+8}) : U_4(2) : 2 < S_{3510}$$

(7th maximal subgroup of the sporadic simple group Fi_{22})

Stabiliser chain of length 4 with 14 strong generators.

Typical elements in G give SLPs of length $\approx$ 15.

$$W := S_{12} \wr S_5 < S_{60}$$

Learning from base and strong generators

The same groups with stabiliser chains:

$$G := (2 \times 2^{1+8}) : U_4(2) : 2 < S_{3510}$$

(7th maximal subgroup of the sporadic simple group Fi_{22})

Stabiliser chain of length 4 with 14 strong generators.

Typical elements in G give SLPs of length ≈ 15.

$$W := S_{12} \wr S_5 < S_{60}$$

Stabiliser chain of length 55 with 434 strong generators.

Learning from base and strong generators

The same groups with stabiliser chains:

$$G := (2 \times 2^{1+8}) : U_4(2) : 2 < S_{3510}$$

(7th maximal subgroup of the sporadic simple group Fi_{22})

Stabiliser chain of length 4 with 14 strong generators.

Typical elements in G give SLPs of length $\approx$ 15.

$$W := S_{12} \wr S_5 < S_{60}$$

Stabiliser chain of length 55 with 434 strong generators.

Typical elements in W give SLPs of length $\approx$ 500.

Comparison

We compare lengths of SLPs:

	Stabiliser chain		Composition tree	
	in strong			in gens
G	15			900
$S_{12} \wr S_5$	500			10000

Comparison

We compare lengths of SLPs:

	Stabiliser chain		Composition tree	
	in strong	in gens		in gens
G	15	290		900
$S_{12} \wr S_5$	500	4300		10000

Comparison

We compare lengths of SLPs:

	Stabiliser chain		Composition tree	
	in strong	in gens		in gens
G	15	290		900
$S_{12} \wr S_5$	500	4300		10000

We want to change the generating system!

$\Longrightarrow$ "nice generators"

Comparison

We compare lengths of SLPs:

	Stabiliser chain		Composition tree	
	in strong	in gens	in nice	in gens
G	15	290	15	900
$S_{12} \wr S_5$	500	4300	300	10000

We want to change the generating system!

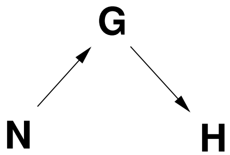
$\Longrightarrow$ "nice generators"

Problems with recursion

$$\begin{tikzcd} & \mathbf{G} \arrow[dr] & \\ \mathbf{N} \arrow[ur] & & \mathbf{H} \end{tikzcd}$$

Recall: Generators of H were images of those of G.

Problems with recursion

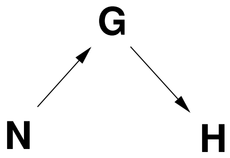


Recall: Generators of H were images of those of G.

Having changed the generators in H,

we can no longer find preimages!

Problems with recursion



Recall: Generators of H were images of those of G.

Having changed the generators in H,

we can no longer find preimages!

Solution: Nice generators of G are

- preimages of the nice generators of H together with
- nice generators of N.

Problems with recursion

$$\begin{tikzcd} & \mathbf{G} \arrow[dr] & \\ \mathbf{N} \arrow[ur] & & \mathbf{H} \end{tikzcd}$$

Recall: Generators of H were images of those of G.

Having changed the generators in H,

we can no longer find preimages!

Solution: Nice generators of G are

- preimages of the nice generators of H together with
- nice generators of N.

Note: The first allows to compute N once H is recognised!

Constructive recognition revisited

Problem — new formulation

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$,

Constructive recognition revisited

Problem — new formulation

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$,
- new nice generators $G = \langle N_1, \ldots, N_m \rangle$ and

Constructive recognition revisited

Problem — new formulation

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$,
- new nice generators $G = \langle N_1, \ldots, N_m \rangle$ and
- a procedure that, given $M \in \mathcal{G}$,
  - decides, whether or not $M \in G$ and
  - if so, expresses M as an SLP in the N_j and

Constructive recognition revisited

Problem — new formulation

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$,
- new nice generators $G = \langle N_1, \ldots, N_m \rangle$ and
- a procedure that, given $M \in \mathcal{G}$,
  - decides, whether or not $M \in G$ and
  - if so, expresses M as an SLP in the N_j and
- another procedure that, given preimages $\hat{M}_1, \ldots, \hat{M}_k$ of the M_i under some homomorphism onto G, produces preimages of the nice generators.

Constructive recognition revisited

Problem — new formulation

Let $\mathcal{G}$ be S_n or $\mathrm{GL}_n(\mathbb{F}_q)$ or $\mathrm{PGL}_n(\mathbb{F}_q)$ and

$$M_1, \ldots, M_k \in \mathcal{G}.$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$,
- new nice generators $G = \langle N_1, \ldots, N_m \rangle$ and
- a procedure that, given $M \in \mathcal{G}$,
  - decides, whether or not $M \in G$ and
  - if so, expresses M as an SLP in the N_j and
- another procedure that, given preimages $\hat{M}_1, \ldots, \hat{M}_k$ of the M_i under some homomorphism onto G, produces preimages of the nice generators.

If this problem is solved, we call

$\langle M_1, \ldots, M_k \rangle$ recognised constructively.

Recursion works again

Having recognised H in this sense, we can:

- ask H to generate preimages of its nice generators,

Recursion works again

Having recognised H in this sense, we can:

- ask H to generate preimages of its nice generators,
- compute generators for N,

Recursion works again

Having recognised H in this sense, we can:

- ask H to generate preimages of its nice generators,
- compute generators for N,
- recursively recognise N and

Recursion works again

Having recognised H in this sense, we can:

- ask H to generate preimages of its nice generators,
- compute generators for N,
- recursively recognise N and
- put together the nice generators for G.

Recursion works again

Having recognised H in this sense, we can:

- ask H to generate preimages of its nice generators,
- compute generators for N,
- recursively recognise N and
- put together the nice generators for G.

If we remember how we created the generators for N, then we have recognised G constructively:

- Using H and N we can test membership in G,

Recursion works again

Having recognised H in this sense, we can:

- ask H to generate preimages of its nice generators,
- compute generators for N,
- recursively recognise N and
- put together the nice generators for G.

If we remember how we created the generators for N, then we have recognised G constructively:

- Using H and N we can test membership in G,
- express elements as SLPs in the nice generators,

Recursion works again

Having recognised H in this sense, we can:

- ask H to generate preimages of its nice generators,
- compute generators for N,
- recursively recognise N and
- put together the nice generators for G.

If we remember how we created the generators for N, then we have recognised G constructively:

- Using H and N we can test membership in G,
- express elements as SLPs in the nice generators,
- and, given preimages of the original generators of G under some homomorphism, we can find preimages of the nice generators.

Example

Let

$$G := S_{12} \wr M_{12} < S_{144}$$

Example

Let

$$G := S_{12} \wr M_{12} < S_{144}$$

G has about 10^{109} elements and is generated by 27 elements.

Example

Let

$$G := S_{12} \wr M_{12} < S_{144}$$

G has about 10^{109} elements and is generated by 27 elements.

Composition tree of depth 5 with 13 non-trivial leaves.

Example

Let

$$G := S_{12} \wr M_{12} < S_{144}$$

G has about 10^{109} elements and is generated by 27 elements.

Composition tree of depth 5 with 13 non-trivial leaves.

Typical elements in G give

Example

Let

$$G := S_{12} \wr M_{12} < S_{144}$$

G has about 10^{109} elements and is generated by 27 elements.

Composition tree of depth 5 with 13 non-trivial leaves.

Typical elements in G give

SLPs of length $\approx$ 800 in 33 nice generators.

Example

Let

$$G := S_{12} \wr M_{12} < S_{144}$$

G has about 10^{109} elements and is generated by 27 elements.

Composition tree of depth 5 with 13 non-trivial leaves.

Typical elements in G give

SLPs of length $\approx$ 800 in 33 nice generators.

SLPs of length $\approx$ 40000 in 27 original generators!

Example

Let

$$G := S_{12} \wr M_{12} < S_{144}$$

G has about 10^{109} elements and is generated by 27 elements.

Composition tree of depth 5 with 13 non-trivial leaves.

Typical elements in G give

SLPs of length $\approx$ 800 in 33 nice generators.

SLPs of length $\approx$ 40000 in 27 original generators!

Note that a stabiliser chain for G has

- length 132 and 2512 strong generators,
- typical SLP in the strong generators: $\approx$ 2700 lines,
- typical SLP in the original generators: $\approx$ 12000 lines.