

Rewriting in black-box (sporadic and classical) groups

Csaba Schneider

Centro de Álgebra
Universidade de Lisboa
csaba.schneider@gmail.com
www.sztaki.hu/~schneider

Matrix Groups Recognition Meeting
Edinburgh, 27 July 2009

Joint work with Ambrose, Murray, Neunhöffer, Praeger



The rewriting problem

An approach to solve the constructive membership problem.
Suppose that G is a black-box group and we have

- (i) identified the name of G ;
- (ii) found standard generators for G .

Given $x \in G$, we want to write x as an SLP in the standard generators.



Use a chain

$$G_0 = G > G_1 > G_2 > \cdots > G_{k-1} > G_k = 1$$

of subgroups.

- Find $x_0 \in G_0$ such that $xx_0 \in G_1$.
- Find $x_1 \in G_1$ such that $xx_0x_1 \in G_2$.

Then $xx_0x_1 \cdots x_{k-1} = 1$ and so $x = x_{k-1}^{-1} \cdots x_1^{-1}x_0^{-1}$ and a straight-line program for G can be computed.

The x_i can be found by either storing a transversal for $[G_i : G_{i+1}]$ or taking random elements in G_i .



Generalized sifting

Use a chain

$$S_0 = G \supset S_1 \supset S_2 \supset \cdots \supset S_{k-1} \supset S_k$$

of subsets. Let $x \in G$.

Step 1: Find $x_0 \in G$ or report FAIL such that

$$P(\text{FAIL or } xx_0 \notin S_1) \leq \varepsilon_0$$

Step 2: Find x_1 with $S_1 x_1 \subseteq S_1$ report FAIL such that

$$P(\text{FAIL or } xx_0 x_1 \notin S_1 | xx_0 \in S_1) \leq \varepsilon_1$$

And continue... Finally $xx_0 x_1 \cdots x_k = 1$ with “high probability”.



Generalized sifting

Use a chain

$$S_0 = G \supset S_1 \supset S_2 \supset \cdots \supset S_{k-1} \supset S_k$$

of subsets. Let $x \in G$.

Step 1: Find $x_0 \in G$ or report FAIL such that

$$P(\text{FAIL or } xx_0 \notin S_1) \leq \varepsilon_0$$

Step 2: Find x_1 with $S_1 x_1 \subseteq S_1$ report FAIL such that

$$P(\text{FAIL or } xx_0 x_1 \notin S_1 | xx_0 \in S_1) \leq \varepsilon_1$$

And continue... Finally $xx_0 x_1 \cdots x_k = 1$ with “high probability”.



Generalized sifting

Use a chain

$$S_0 = G \supset S_1 \supset S_2 \supset \cdots \supset S_{k-1} \supset S_k$$

of subsets. Let $x \in G$.

Step 1: Find $x_0 \in G$ or report FAIL such that

$$P(\text{FAIL or } xx_0 \notin S_1) \leq \varepsilon_0$$

Step 2: Find x_1 with $S_1 x_1 \subseteq S_1$ report FAIL such that

$$P(\text{FAIL or } xx_0 x_1 \notin S_1 | xx_0 \in S_1) \leq \varepsilon_1$$

And continue... Finally $xx_0 x_1 \cdots x_k = 1$ with “high probability”.



Generalized sifting

Use a chain

$$S_0 = G \supset S_1 \supset S_2 \supset \cdots \supset S_{k-1} \supset S_k$$

of subsets. Let $x \in G$.

Step 1: Find $x_0 \in G$ or report FAIL such that

$$P(\text{FAIL or } xx_0 \notin S_1) \leq \varepsilon_0$$

Step 2: Find x_1 with $S_1 x_1 \subseteq S_1$ report FAIL such that

$$P(\text{FAIL or } xx_0 x_1 \notin S_1 | xx_0 \in S_1) \leq \varepsilon_1$$

And continue... Finally $xx_0 x_1 \cdots x_k = 1$ with “high probability”.

Generalized sifting

Ingredients: Membership tests for S_i .

In HS:

$$HS \supseteq CT_1 U_2(5) : 2 \supseteq CT_2 5^{1+2} : (8 : 2) \supseteq \dots$$

Details are worked out for about half of the sporadic groups;
see [ANPSch, LMS JCM, 2005].



Black-box general linear groups

Rewriting problem for black-box $SX(n, q)$ ($X \in \{L, p\}$).

The standard generators Σ , Δ , U , V for $SL(n, q)$ are:

$$\Sigma : b_1 \mapsto b_1 + b_2$$

$$\Delta : b_1 \mapsto \delta b_1, b_2 \mapsto \delta^{-1} b_2$$

$$U : b_1 \leftrightarrow -b_2$$

$$V : b_1 \mapsto b_n, b_2 \mapsto -b_1, \dots, b_n \mapsto -b_{n-1}$$

Given $x \in G$, write it as a straight-line program in $\{\Sigma, \Delta, U, V\}$.



Black-box general linear groups

Rewriting problem for black-box $SX(n, q)$ ($X \in \{L, p\}$).

The standard generators Σ , Δ , U , V for $SL(n, q)$ are:

$$\Sigma : b_1 \mapsto b_1 + b_2$$

$$\Delta : b_1 \mapsto \delta b_1, b_2 \mapsto \delta^{-1} b_2$$

$$U : b_1 \leftrightarrow -b_2$$

$$V : b_1 \mapsto b_n, b_2 \mapsto -b_1, \dots, b_n \mapsto -b_{n-1}$$

Given $x \in G$, write it as a straight-line program in $\{\Sigma, \Delta, U, V\}$.



Step 1

Use the idea of generalized sifting and define

$$S = \{g \in G \mid g_{1,n} = 0\}.$$

Membership testing for S :

Define $X_{i,j}(\alpha) = I + \alpha E(i,j)$ and $X_{i,j} = X_{i,j}$.

For $x \in G$ the following are equivalent:

- $x \in S$
- $[X_{n,1} \widehat{(X_{n,1} \widehat{x})}, X_{n,i}] = 1$ for all $i \in \{1, \dots, n-1\}$.

Step 1: Find $x_1 \in G$ such that $xx_1 \in S$.

- Check, for all $\alpha \in \mathbb{F}_q$, if $xX_{1,n}(\alpha) \in S$. Must work if $x_{1,1} \neq 0$.
- Otherwise $xP(1,n) \in S$.

Cost: $O(nq)$ multiplications.



Step 1

Use the idea of generalized sifting and define

$$S = \{g \in G \mid g_{1,n} = 0\}.$$

Membership testing for S :

Define $X_{i,j}(\alpha) = I + \alpha E(i,j)$ and $X_{i,j} = X_{i,j}$.

For $x \in G$ the following are equivalent:

- $x \in S$
- $[X_{n,1} \widehat{(X_{n,1} \widehat{x})}, X_{n,i}] = 1$ for all $i \in \{1, \dots, n-1\}$.

Step 1: Find $x_1 \in G$ such that $xx_1 \in S$.

- Check, for all $\alpha \in \mathbb{F}_q$, if $xX_{1,n}(\alpha) \in S$. Must work if $x_{1,1} \neq 0$.
- Otherwise $xP(1, n) \in S$.

Cost: $O(nq)$ multiplications.



Step 1

Use the idea of generalized sifting and define

$$S = \{g \in G \mid g_{1,n} = 0\}.$$

Membership testing for S :

Define $X_{i,j}(\alpha) = I + \alpha E(i,j)$ and $X_{i,j} = X_{i,j}$.

For $x \in G$ the following are equivalent:

- $x \in S$
- $[X_{n,1} \widehat{(X_{n,1} \widehat{x})}, X_{n,i}] = 1$ for all $i \in \{1, \dots, n-1\}$.

Step 1: Find $x_1 \in G$ such that $xx_1 \in S$.

- Check, for all $\alpha \in \mathbb{F}_q$, if $xX_{1,n}(\alpha) \in S$. Must work if $x_{1,1} \neq 0$.
- Otherwise $xP(1,n) \in S$.

Cost: $O(nq)$ multiplications.



Step 1

Use the idea of generalized sifting and define

$$S = \{g \in G \mid g_{1,n} = 0\}.$$

Membership testing for S :

Define $X_{i,j}(\alpha) = I + \alpha E(i,j)$ and $X_{i,j} = X_{i,j}$.

For $x \in G$ the following are equivalent:

- $x \in S$
- $[X_{n,1} \widehat{(X_{n,1} \widehat{x})}, X_{n,i}] = 1$ for all $i \in \{1, \dots, n-1\}$.

Step 1: Find $x_1 \in G$ such that $xx_1 \in S$.

- Check, for all $\alpha \in \mathbb{F}_q$, if $xX_{1,n}(\alpha) \in S$. Must work if $x_{1,1} \neq 0$.
- Otherwise $xP(1, n) \in S$.

Cost: $O(nq)$ multiplications.



Step 2

Assume now that $x \in S$; that is $x_{1,n} = 0$.

Define

$$T = \{g \in G \mid g_{1,2} = \cdots = g_{1,n} = 0\}.$$

Then $T \leq G$. How to get into T ?

$$x \cdot \begin{pmatrix} 1 & -x_{1,1}^{-1}x_{1,2} & \cdots & -x_{1,1}^{-1}x_{1,n-1} & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \in T.$$

Problem: We don't know the entries $x_{i,j}$.



Step 2

Assume now that $x \in S$; that is $x_{1,n} = 0$.

Define

$$T = \{g \in G \mid g_{1,2} = \cdots = g_{1,n} = 0\}.$$

Then $T \leq G$. How to get into T ?

$$x \cdot \begin{pmatrix} 1 & -x_{1,1}^{-1}x_{1,2} & \cdots & -x_{1,1}^{-1}x_{1,n-1} & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \in T.$$

Problem: We don't know the entries $x_{i,j}$.



Step 2

Let

$$y = x^{-1}$$

$$r = X_{n,1}^{-1} X_{n,1} \hat{(X_{n,1} \hat{x})}$$

$$s = [X_{1,2}, r \hat{U}]$$

Then

$$rs = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & y_{1,n} x_{1,2} & \cdots & y_{1,n} x_{1,n-1} & 1 \end{pmatrix}$$



Step 2

Let

$$y = x^{-1}$$

$$r = X_{n,1}^{-1} X_{n,1} \widehat{(X_{n,1} x)}$$

$$s = [X_{1,2}, r \widehat{U}]$$

Then

$$rs = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & y_{1,n} x_{1,2} & \cdots & y_{1,n} x_{1,n-1} & 1 \end{pmatrix}$$



Step 2

Membership test in T : For $x \in G$ the following are equivalent:

- $x \in T$.
- $[X_{2,1} \hat{x}, X_{i,1}] = 1$ for all $i \in \{1, \dots, n-1\}$.

From rs we may obtain the right element using $O(nq)$ multiplications. It can be written as a straight-line program with length $O(n \log q)$ using $O(nq)$ multiplications.



Step 2

Membership test in T : For $x \in G$ the following are equivalent:

- $x \in T$.
- $[X_{2,1} \hat{x}, X_{i,1}] = 1$ for all $i \in \{1, \dots, n-1\}$.

From rs we may obtain the right element using $O(nq)$ multiplications. It can be written as a straight-line program with length $O(n \log q)$ using $O(nq)$ multiplications.



Finishing off

After steps 1 and 2 we may assume that $x_{1,2} = \cdots = x_{1,n} = 0$, and similarly $x_{2,1} = \cdots = x_{n,1} = 0$.

There are two ways of proceeding:

- **Method 1:** Clear the first row and the first column; then the 2nd row and the 2nd column; etc. This requires $O(n^2 q)$ group operations and the length of the result is $O(n^2 \log q)$.
- **Method 2:** We may obtain the remaining $(n - 1) \times (n - 1)$ matrix explicitly.



Finishing off

After steps 1 and 2 we may assume that $x_{1,2} = \dots = x_{1,n} = 0$, and similarly $x_{2,1} = \dots = x_{n,1} = 0$.

There are two ways of proceeding:

- **Method 1:** Clear the first row and the first column; then the 2nd row and the 2nd column; etc. This requires $O(n^2 q)$ group operations and the length of the result is $O(n^2 \log q)$.
- **Method 2:** We may obtain the remaining $(n - 1) \times (n - 1)$ matrix explicitly.



Finishing off

After steps 1 and 2 we may assume that $x_{1,2} = \cdots = x_{1,n} = 0$, and similarly $x_{2,1} = \cdots = x_{n,1} = 0$.

There are two ways of proceeding:

- **Method 1:** Clear the first row and the first column; then the 2nd row and the 2nd column; etc. This requires $O(n^2 q)$ group operations and the length of the result is $O(n^2 \log q)$.
- **Method 2:** We may obtain the remaining $(n - 1) \times (n - 1)$ matrix explicitly.



Obtaining the remaining matrix

Assume that $x_{1,2} = \cdots = x_{1,n} = 0 = x_{2,1} = \cdots = x_{n,1} = 0$. If $i \in \{2, \dots, n\}$, $j \in \{2, \dots, n-1\}$, then

$$[X_{1,i} \widehat{X}, X_{j,n}] = X_{1,n}(x_{i,j}x_{1,1}^{-1}).$$

Using Elliot Conti's implementation, we write the matrix obtained as SLP in the standard generators of $SL(n-1, q)$.



Obtaining the remaining matrix

Assume that $x_{1,2} = \cdots = x_{1,n} = 0 = x_{2,1} = \cdots = x_{n,1} = 0$. If $i \in \{2, \dots, n\}$, $j \in \{2, \dots, n-1\}$, then

$$[X_{1,i} \widehat{X}, X_{j,n}] = X_{1,n}(x_{i,j}x_{1,1}^{-1}).$$

Using Elliot Conti's implementation, we write the matrix obtained as SLP in the standard generators of $SL(n-1, q)$.



Using an $SL(3, q)$ -oracle

Theorem

If $x \in G$ and some conditions hold, then

$$H = \langle X_{n,1}, X_{n,1} \hat{x}, X_{1,2} \rangle \cong SL(3, q)$$

and $h \in H$ can be found such that $xh \in T$.

Corollary

$x \in G$ can be written as a straight-line program using polynomially many group operations and polynomially many calls to an $SL(3, q)$ -oracle.

Theorem (Lübeck, Magaard, and O'Brien '97)

The constructive membership problem in a black-box $SL(3, q)$ can be solved using a poly number of group operations, a poly number of random elements, and a poly many calls to an $SL(2, q)$ -oracle.



Using an $SL(3, q)$ -oracle

Theorem

If $x \in G$ and some conditions hold, then

$$H = \langle X_{n,1}, X_{n,1} \hat{x}, X_{1,2} \rangle \cong SL(3, q)$$

and $h \in H$ can be found such that $xh \in T$.

Corollary

$x \in G$ can be written as a straight-line program using polynomially many group operations and polynomially many calls to an $SL(3, q)$ -oracle.

Theorem (Lübeck, Magaard, and O'Brien '97)

The constructive membership problem in a black-box $SL(3, q)$ can be solved using a poly number of group operations, a poly number of random elements, and a poly many calls to an $SL(2, q)$ -oracle.



Using an $SL(3, q)$ -oracle

Theorem

If $x \in G$ and some conditions hold, then

$$H = \langle X_{n,1}, X_{n,1} \hat{x}, X_{1,2} \rangle \cong SL(3, q)$$

and $h \in H$ can be found such that $xh \in T$.

Corollary

$x \in G$ can be written as a straight-line program using polynomially many group operations and polynomially many calls to an $SL(3, q)$ -oracle.

Theorem (Lübeck, Magaard, and O'Brien '97)

The constructive membership problem in a black-box $SL(3, q)$ can be solved using a poly number of group operations, a poly number of random elements, and a poly many calls to an $SL(2, q)$ -oracle.



Implementation

Magma implementation is written for $SL(n, q)$ and $Sp(n, q)$.

group	#	avg time of rewriting	avg length of SLP
$SL(4, 5) \leq GL(155, 7)$		0.06s	72
$SL(4, 5)$		0.001s	43
$SL(3, 9) \leq GL(90, 11)$		0.62s	52
$SL(3, 9)$		0.001s	38

