Mysteries of 2×2 matrices

Vladimir Shpilrain
The City College of New York

Newcastle, June 2016

Two theorems of Sanov

[I. N. Sanov, *A property of a representation of a free group* (Russian), Doklady Akad. Nauk SSSR (N. S.) **57** (1947), 657–659]

Denote $A(k) = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$, $B(k) = \begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix}$.

Two theorems of Sanov

[I. N. Sanov, *A property of a representation of a free group* (Russian), Doklady Akad. Nauk SSSR (N. S.) **57** (1947), 657–659]

Denote $A(k) = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$, $B(k) = \begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix}$.

Theorem

The subgroup of $SL_2(\mathbb{Z})$ generated by $A(2)$ and $B(2)$ is free.

Theorem

The subgroup of $SL_2(\mathbb{Z})$ generated by $A(2)$ and $B(2)$ consists of all matrices of the form $\begin{pmatrix} 1+4n_1 & 2n_2 \\ 2n_3 & 1+4n_4 \end{pmatrix}$ with determinant 1, where all n_i are arbitrary integers.

Corollaries

Corollary

The group $SL_2(\mathbb{Z})$ is virtually free.

Corollaries

Corollary

The group $SL_2(\mathbb{Z})$ is virtually free.

Corollary

The membership problem in the subgroup of $SL_2(\mathbb{Z})$ generated by $A(2)$ and $B(2)$ is solvable in constant time.

Corollaries

Corollary

The group $SL_2(\mathbb{Z})$ is virtually free.

Corollary

The membership problem in the subgroup of $SL_2(\mathbb{Z})$ generated by $A(2)$ and $B(2)$ is solvable in constant time.

This is, to the best of our knowledge, the only example of a natural (and nontrivial) algorithmic problem in group theory solvable in constant time. In fact, even problems solvable in sublinear time are very rare, and in those that are, one can typically get either "yes" or "no" answer in sublinear time, but not both.

Open problems

Problem

(Yu. Merzlyakov) For which rational k, $0 < k < 2$, is the group generated by $A(k)$ and $B(k)$ free? More generally, for which algebraic k, $0 < k < 2$, is this group free?

It is well known that this group is free if $|k| \geq 2$. No examples are known where this group is free if k is rational, $|k| < 2$.

Open problems

Problem

(Yu. Merzlyakov) For which rational k, $0 < k < 2$, is the group generated by $A(k)$ and $B(k)$ free? More generally, for which algebraic k, $0 < k < 2$, is this group free?

It is well known that this group is free if $|k| \geq 2$. No examples are known where this group is free if k is rational, $|k| < 2$.

On the other hand, if r and s are algebraic numbers that are Galois conjugate over $\mathbb{Q}$, then the group generated by $A(r)$ and $B(r)$ is free if and only if the group generated by $A(s)$ and $B(s)$ is. In particular, if $r = 2 - \sqrt{2}$, then $A(r)$ and $B(r)$ generate a free group.

Open problems

Problem

(Yu. Merzlyakov) For which rational k, $0 < k < 2$, is the group generated by $A(k)$ and $B(k)$ free? More generally, for which algebraic k, $0 < k < 2$, is this group free?

It is well known that this group is free if $|k| \geq 2$. No examples are known where this group is free if k is rational, $|k| < 2$.

On the other hand, if r and s are algebraic numbers that are Galois conjugate over $\mathbb{Q}$, then the group generated by $A(r)$ and $B(r)$ is free if and only if the group generated by $A(s)$ and $B(s)$ is. In particular, if $r = 2 - \sqrt{2}$, then $A(r)$ and $B(r)$ generate a free group.

There are many examples of rational k, $0 < k < 2$, such that this group is not free. In particular, $k = \frac{m}{mn+1}$ or $k = \frac{m+n}{mn}$, $m, n \in Z_+$.

References

[R. C. Lyndon and J. L. Ullman, *Groups Generated by Two Linear Parabolic Transformations*, Canadian J. Math. **21** (1969), 1388–1403]

[M. Gutan, *Diophantine equations and the freeness of Möbius groups*, Applied Math. **5** (2014), 1400–1411]

More open problems

Problem

The subgroup membership problem for the group $SL_2(\mathbb{Q})$.

More open problems

Problem

The subgroup membership problem for the group $SL_2(\mathbb{Q})$.

Problem

The subgroup membership problem for the subgroup of $SL_2(\mathbb{Q})$ generated by $A(k)$ and $B(k)$, $k \in \mathbb{Q}$.

Reference

[A. Chorna, K. Geller, V. Shpilrain, *On two-generator subgroups of* $SL_2(\mathbb{Z})$, $SL_2(\mathbb{Q})$, *and* $SL_2(\mathbb{R})$]

Our contribution: greedy algorithm

Theorem

Let $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$ *be a matrix from* $SL_2(\mathbb{R})$. *Call "elementary operations" on* M *the following 8 operations: multiplication of* M *by either* $A(k)^{\pm 1}$ *or by* $B(k)^{\pm 1}$, *on the right or on the left.*

Our contribution: greedy algorithm

Theorem

Let $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$ *be a matrix from* $SL_2(\mathbb{R})$. *Call "elementary operations" on* M *the following 8 operations: multiplication of* M *by either* $A(k)^{\pm 1}$ *or by* $B(k)^{\pm 1}$, *on the right or on the left.*

(a) *If* $k \in \mathbb{Z}$ *and* M *belongs to the subgroup of* $SL_2(\mathbb{Z})$ *generated by* $A(k)$ *and* $B(k)$, *then it has the form* $\begin{pmatrix} 1 + k^2 n_1 & k n_2 \\ k n_3 & 1 + k^2 n_4 \end{pmatrix}$ *for some integers* n_i.

Our contribution: greedy algorithm

Theorem

Let $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$ *be a matrix from* $SL_2(\mathbb{R})$. *Call "elementary operations" on* M *the following 8 operations: multiplication of* M *by either* $A(k)^{\pm 1}$ *or by* $B(k)^{\pm 1}$, *on the right or on the left.*

(a) *If* $k \in \mathbb{Z}$ *and* M *belongs to the subgroup of* $SL_2(\mathbb{Z})$ *generated by* $A(k)$ *and* $B(k)$, *then it has the form* $\begin{pmatrix} 1+k^2 n_1 & k n_2 \\ k n_3 & 1+k^2 n_4 \end{pmatrix}$ *for some integers* n_i.

If $k \in \mathbb{R}$ *and* M *belongs to the subgroup of* $SL_2(\mathbb{R})$ *generated by* $A(k)$ *and* $B(k)$, *then it has the form* $\begin{pmatrix} 1+\sum_i k^i n_i & \sum_j k^j n_j \\ \sum_r k^r n_r & 1+\sum_s k^s n_s \end{pmatrix}$ *where all* n_i *are integers and all exponents on* k *are positive integers.*

Our contribution: greedy algorithm

Theorem

Let $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$ *be a matrix from* $SL_2(\mathbb{R})$. *Call "elementary operations" on* M *the following 8 operations: multiplication of* M *by either* $A(k)^{\pm 1}$ *or by* $B(k)^{\pm 1}$, *on the right or on the left.*

(a) *If* $k \in \mathbb{Z}$ *and* M *belongs to the subgroup of* $SL_2(\mathbb{Z})$ *generated by* $A(k)$ *and* $B(k)$, *then it has the form* $\begin{pmatrix} 1+k^2 n_1 & k n_2 \\ k n_3 & 1+k^2 n_4 \end{pmatrix}$ *for some integers* n_i.

If $k \in \mathbb{R}$ *and* M *belongs to the subgroup of* $SL_2(\mathbb{R})$ *generated by* $A(k)$ *and* $B(k)$, *then it has the form* $\begin{pmatrix} 1+\sum_i k^i n_i & \sum_j k^j n_j \\ \sum_r k^r n_r & 1+\sum_s k^s n_s \end{pmatrix}$ *where all* n_i *are integers and all exponents on* k *are positive integers.*

(b) *Let* $k \in \mathbb{Z},\ k \geq 2$. *If* $M \in SL_2(\mathbb{Z})$ *and no elementary operation reduces* $\sum_{i,j} |m_{ij}|$, *then either* M *is the identity matrix or* M *does not belong to the subgroup generated by* $A(k)$ *and* $B(k)$.

Corollary 1

Corollary

The subgroup of $SL_2(\mathbb{Z})$ *generated by* $A(k)$ *and* $B(k)$, $k \in \mathbb{Z},\ k \geq 3$, *has infinite index in the group of all matrices of the form* $\begin{pmatrix} 1+k^2m_1 & km_2 \\ km_3 & 1+k^2m_4 \end{pmatrix}$ *with determinant 1.*

Corollary 2

Corollary

Let $k \in \mathbb{Z},\ k \geq 2$, and let the complexity $|M|$ of a matrix $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$ be the sum of all $|m_{ij}|$. There is an algorithm that decides whether or not a given matrix $M \in SL_2(\mathbb{Z})$ is in the subgroup of $SL_2(\mathbb{Z})$ generated by $A(k)$ and $B(k)$ (and if it does, finds a presentation of M as a group word in $A(k)$ and $B(k)$) in time $O(n \cdot \log n)$, where $n = |M|$.

Generic-case complexity

The $O(n \cdot \log n)$ is the worst-case complexity of this algorithm. It would be interesting to find out what the *generic-case complexity* (in the sense of [I. Kapovich, A. G. Myasnikov, P. Schupp, V. Shpilrain, *Generic-case complexity, decision problems in group theory and random walks*, J. Algebra **264** (2003), 665–694]) of this algorithm is.

Generic-case complexity

The $O(n \cdot \log n)$ is the worst-case complexity of this algorithm. It would be interesting to find out what the *generic-case complexity* (in the sense of [I. Kapovich, A. G. Myasnikov, P. Schupp, V. Shpilrain, *Generic-case complexity, decision problems in group theory and random walks*, J. Algebra **264** (2003), 665–694]) of this algorithm is.

Proposition 1 in [L. Bromberg, V. Shpilrain, A. Vdovina, *Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing*, Semigroup Forum] tacitly suggests that this complexity might be, in fact, sublinear in $n = |M|$.

Generic-case complexity

The $O(n \cdot \log n)$ is the worst-case complexity of this algorithm. It would be interesting to find out what the *generic-case complexity* (in the sense of [I. Kapovich, A. G. Myasnikov, P. Schupp, V. Shpilrain, *Generic-case complexity, decision problems in group theory and random walks*, J. Algebra **264** (2003), 665–694]) of this algorithm is.

Proposition 1 in [L. Bromberg, V. Shpilrain, A. Vdovina, *Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing*, Semigroup Forum] tacitly suggests that this complexity might be, in fact, sublinear in $n = |M|$.

Problem

Is the generic-case complexity of the algorithm claimed in Corollary 10 sublinear in $|M|$?

Generic-case complexity

The $O(n \cdot \log n)$ is the worst-case complexity of this algorithm. It would be interesting to find out what the *generic-case complexity* (in the sense of [I. Kapovich, A. G. Myasnikov, P. Schupp, V. Shpilrain, *Generic-case complexity, decision problems in group theory and random walks*, J. Algebra **264** (2003), 665–694]) of this algorithm is.

Proposition 1 in [L. Bromberg, V. Shpilrain, A. Vdovina, *Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing*, Semigroup Forum] tacitly suggests that this complexity might be, in fact, sublinear in $n = |M|$.

Problem

Is the generic-case complexity of the algorithm claimed in Corollary 10 sublinear in $|M|$?

Unlike most algorithms with low generic-case complexity, this algorithm has a good chance to have low generic-case complexity giving both "yes" and "no" answers.

Semigroups of matrices: fun facts

The semigroup generated by $A(1)$ and $B(1)$ is free (Nielsen (?))

Semigroups of matrices: fun facts

The semigroup generated by $A(1)$ and $B(1)$ is free (Nielsen (?))

The semigroup generated by $A(-1)$ and $B(-1)$ is free.

Semigroups of matrices: fun facts

The semigroup generated by $A(1)$ and $B(1)$ is free (Nielsen (?))

The semigroup generated by $A(-1)$ and $B(-1)$ is free.

The semigroup generated by $A(1)$ and $B(-1)$ is not free: if $A = A(1)$ and $B = B(-1)$, then $ABA = BAB$.

More fun facts

The semigroup generated by $A(1)$ and $B(1)$ consists of all matrices of the form $\begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix}$ with determinant 1, where all n_i are nonnegative integers.

More fun facts

The semigroup generated by $A(1)$ and $B(1)$ consists of all matrices of the form $\begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix}$ with determinant 1, where all n_i are nonnegative integers.

The semigroup generated by $A(-1)$ and $B(-1)$ consists of all matrices of the form $\begin{pmatrix} n_1 & -n_2 \\ -n_3 & n_4 \end{pmatrix}$ with determinant 1, where all n_i are nonnegative integers.

More fun facts

The semigroup generated by $A(1)$ and $B(1)$ consists of all matrices of the form $\begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix}$ with determinant 1, where all n_i are nonnegative integers.

The semigroup generated by $A(-1)$ and $B(-1)$ consists of all matrices of the form $\begin{pmatrix} n_1 & -n_2 \\ -n_3 & n_4 \end{pmatrix}$ with determinant 1, where all n_i are nonnegative integers.

The "naive" analog of Sanov's Theorem 2 does *not* hold for the *semigroup* generated by $A(2)$ and $B(2)$. Specifically, the matrix $\begin{pmatrix} 5 & 4 \\ 6 & 5 \end{pmatrix}$ is not in that semigroup, although it is in the *group* generated by $A(2)$ and $B(2)$.

References

We are now switching to (finite) groups $SL_2(\mathbb{F}_p)$ and their sub(semi)groups.

[H. A. Helfgott, *Growth and generation in $SL_2(\mathbb{Z}/p\mathbb{Z})$* Ann. of Math. (2) **167** (2008), 601–623]

[J. Bourgain, A. Gamburd, *Uniform expansion bounds for Cayley graphs of $SL_2(\mathbf{F}_p)$*. Ann. of Math. (2) **167** (2008), 625–642]

References

We are now switching to (finite) groups $SL_2(\mathbb{F}_p)$ and their sub(semi)groups.

[H. A. Helfgott, *Growth and generation in $SL_2(\mathbb{Z}/p\mathbb{Z})$* Ann. of Math. (2) **167** (2008), 601–623]

[J. Bourgain, A. Gamburd, *Uniform expansion bounds for Cayley graphs of $SL_2(\mathbf{F}_p)$*. Ann. of Math. (2) **167** (2008), 625–642]

We are specifically interested in the *girth* of the Cayley graph of the subgroup of $SL_2(\mathbb{F}_p)$ generated by $A(k)$ and $B(k)$.

Hash function

Definition

Let $n \in \mathbb{Z}_+$ and let $H\colon \{0,1\}^* \to \{0,1\}^n$ be a function that takes a bit string of an arbitrary length to a bit string of a fixed length n. We require a hash function H to satisfy the following conditions:

1. *preimage resistance*: given an output y, it is hard to find an input x such that $H(x) = y$
2. *second preimage resistance*: given an input x_1, it is hard to find another input $x_2 \neq x_1$ such that $H(x_1) = H(x_2)$
3. *collision resistance*: it is hard to find inputs $x_1 \neq x_2$ such that $H(x_1) = H(x_2)$.

Cayley hash functions

One can use two elements, A and B, of a semigroup S, such that the Cayley graph of the semigroup generated by A and B belongs to an *expander family*, in the hope that such a graph will have large girth and therefore there will be no short relations ('collisions').

Cayley hash functions

One can use two elements, A and B, of a semigroup S, such that the Cayley graph of the semigroup generated by A and B belongs to an *expander family*, in the hope that such a graph will have large girth and therefore there will be no short relations ('collisions').

To build a hash function from the Cayley graph, a message m (a bitstring comprised of 0's and 1's) corresponds to a word in the elements A and B of S, with A corresponding to 0 and B corresponding to 1. This is represented on the Cayley graph as a (nonbacktracking) walk; the endpoint of the walk is the hash value.

Tillich–Zémor hash function

Tillich and Zémor in [J.-P. Tillich and G. Zémor, *Group-theoretic hash functions*, Lecture Notes Comp. Sci. **781** (1993) 90–110] use matrices A, B from the group $\mathrm{SL}_2(R)$, where $R = \mathbb{F}_2[x]/(p(x))$ is a Galois field $\mathbb{F}_{2^n}$, where n is the degree of the irreducible polynomial $p(x)$. They took $p(x) = x^{131} + x^7 + x^6 + x^5 + x^4 + x + 1$.

Then, the matrices A and B are

$$A = \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \alpha & \alpha + 1 \\ 1 & 1 \end{pmatrix},$$

where α is a root of $p(x)$.

This particular hash function was successfully attacked in [M. Grassl, I. Ilić, S. Magliveras, R. Steinwandt, *Cryptanalysis of the Tillich–Zémor hash function*, J. Cryptology **24** (2011) 148–156] and [C. Petit, J. Quisquatar, *Preimages for the Tillich–Zémor hash function*, in SAC 10, Lecture Notes in Comp. Sci. **6544** (2010) 282–301].

Hashing with matrices over $\mathbb{F}_p$

Use a pair of 2×2 matrices A and B which generate a free monoid over $\mathbb{Z}$, and then reduce the entries modulo a large prime p to get matrices over $\mathbb{F}_p$.

Hashing with matrices over $\mathbb{F}_p$

Use a pair of 2×2 matrices A and B which generate a free monoid over $\mathbb{Z}$, and then reduce the entries modulo a large prime p to get matrices over $\mathbb{F}_p$.

Since there cannot be equality of two different products of positive powers of A and B unless at least one of the entries in at least one of the products is greater than or equal to p, this gives an explicit lower bound on the minimum length of bit strings where a collision may occur.

Matrices over $\mathbb{F}_p$

An example of a pair of matrices over $\mathbb{Z}$ which generate a free monoid is

$$A(1) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B(1) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

These matrices are invertible and thus actually generate the *group* $\mathrm{SL}_2(\mathbb{Z})$. This group is not free, but the *monoid* generated by $A(1)$ and $B(1)$ is free. Since only positive powers are used in hashing, this is all we need.

However, a collision for the hash function corresponding to these matrices over a large prime p was found by Tillich and Zémor in [J.-P. Tillich and G. Zémor, *Hashing with* SL_2, in CRYPTO 1994, Lecture Notes in Comp. Sci. **839** (1994) 40–49] by using what they called a "lifting attack".

Pairs of generators

Any pair of matrices

$$A(x) = \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix}, \quad B(y) = \begin{pmatrix} 1 & 0 \\ y & 1 \end{pmatrix},$$

generate a free *monoid* if $x, y \in Z_+$. We consider the cases where $x = y = 2$ and $x = 1, y = 2$.

Relations over $\mathbb{F}_p$

[L. Bromberg, V. Shpilrain, A. Vdovina, *Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing*, Semigroup Forum]

Relations over $\mathbb{F}_p$

[L. Bromberg, V. Shpilrain, A. Vdovina, *Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing*, Semigroup Forum]

Note: by the pigeonhole principle, there must be relations of length $O(\log p)$ between any two elements of $SL_2(\mathbb{F}_p)$, but a particular relation is typically *computationally hard to find*.

Relations over $\mathbb{F}_p$

[L. Bromberg, V. Shpilrain, A. Vdovina, *Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing*, Semigroup Forum]

Note: by the pigeonhole principle, there must be relations of length $O(\log p)$ between any two elements of $SL_2(\mathbb{F}_p)$, but a particular relation is typically *computationally hard to find*.

Theorem

There is an efficient probabilistic algorithm that finds particular relations of the form $w(A(2), B(2)) = 1$, where w is a group word of length $O(\log p)$, and the matrices $A(2)$ and $B(2)$ are considered over $\mathbb{F}_p$.

Relations over $\mathbb{F}_p$

[L. Bromberg, V. Shpilrain, A. Vdovina, *Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing*, Semigroup Forum]

Note: by the pigeonhole principle, there must be relations of length $O(\log p)$ between any two elements of $SL_2(\mathbb{F}_p)$, but a particular relation is typically *computationally hard to find*.

Theorem

There is an efficient probabilistic algorithm that finds particular relations of the form $w(A(2), B(2)) = 1$, where w is a group word of length $O(\log p)$, and the matrices $A(2)$ and $B(2)$ are considered over $\mathbb{F}_p$.

The proof uses Sanov's Theorem 2 and Tillich–Zémor's "lifting attack".

Semigroup vs group relations

This does not affect the security of the hash function based on $A(2)$ and $B(2)$ since only positive powers of $A(2)$ and $B(2)$ are used, and the group relations produced by the algorithm will involve both negative and positive powers with overwhelming probability.

Semigroup vs group relations

This does not affect the security of the hash function based on $A(2)$ and $B(2)$ since only positive powers of $A(2)$ and $B(2)$ are used, and the group relations produced by the algorithm will involve both negative and positive powers with overwhelming probability.

Note that the number of matrices in the above form which are representable by positive words is negligible. In fact, the number of distinct elements represented by all freely reducible words in $A(2)$ and $B(2)$ of length $n \geq 2$ is $4 \cdot 3^{n-1}$, while the number of distinct elements represented by positive words of length $n \geq 2$ is 2^n.

Girth of the Cayley graph generated by $A(k)$ and $B(k)$

For hashing, we use only positive powers, so we need only consider products of positive powers of $A(k)$ and $B(k)$. We note that entries in a matrix of a length n product of positive powers of $A(k)$ and $B(k)$ grow fastest (as functions of n) in the alternating product of $A(k)$ and $B(k)$. This is summarized in the following proposition.

Proposition

Let $w_n(a, b)$ be an arbitrary positive word of even length n, and let $W_n = w_n(A(k), B(k))$, with $k \geq 2$. Let $C_n = (A(k) \cdot B(k))^{n/2}$. Then: **(a)** *the sum of entries in any row of C_n is at least as large as the sum of entries in any row of W_n;* **(b)** *the largest entry of C_n is at least as large as the largest entry of W_n.*

Lower bound on girth

No entry of $(C(2))^n$ is larger than p as long as $n < \log_{3+\sqrt{8}} p$. Since $C(2) = A(2)B(2)$ is a product of two generators, $(C(2))^n$ has length $2n$ as a word in the generators $A(2)$ and $B(2)$.

Lower bound on girth

No entry of $(C(2))^n$ is larger than p as long as $n < \log_{3+\sqrt{8}} p$. Since $C(2) = A(2)B(2)$ is a product of two generators, $(C(2))^n$ has length $2n$ as a word in the generators $A(2)$ and $B(2)$.

Therefore, no two positive words of length $\leq m$ in the generators $A(2)$ and $B(2)$ (considered as matrices over $\mathbb{F}_p$) can be equal as long as

$$m < 2 \log_{3+\sqrt{8}} p = \log_{\sqrt{3+\sqrt{8}}} p.$$

Lower bound on girth

No entry of $(C(2))^n$ is larger than p as long as $n < \log_{3+\sqrt{8}} p$. Since $C(2) = A(2)B(2)$ is a product of two generators, $(C(2))^n$ has length $2n$ as a word in the generators $A(2)$ and $B(2)$.

Therefore, no two positive words of length $\leq m$ in the generators $A(2)$ and $B(2)$ (considered as matrices over $\mathbb{F}_p$) can be equal as long as

$$m < 2 \log_{3+\sqrt{8}} p = \log_{\sqrt{3+\sqrt{8}}} p.$$

In particular, the girth of the Cayley graph of the semigroup generated by $A(2)$ and $B(2)$ (considered as matrices over $\mathbb{F}_p$) is at least $\log_{\sqrt{3+\sqrt{8}}} p \approx \log_{2.4} p$. For example, if p is on the order of 2^{256}, there will be no collisions of the form $u(A(1), B(2)) = v(A(1), B(2))$ if positive words u and v are of length less than 203.

Problem

Problem

Determine which semigroup words in the matrices $A(1), B(2)$ will exhibit the fastest growth of their entries.

Problem

Problem

Determine which semigroup words in the matrices $A(1), B(2)$ will exhibit the fastest growth of their entries.

This problem is of interest also because if the alternating product again gives fastest growth, then a similar calculation as was done for $A(2), B(2)$ would show a lower bound with a smaller log base $\sqrt{2+\sqrt{3}}$, which is about 1.93. This would mean that for p on the order of 2^{256}, there will be no collisions of the form $u(A(1), B(2)) = v(A(1), B(2))$ if positive words u and v are of length less than 269. This is a stronger lower bound than for the $A(2), B(2)$ case.

Thank you