Compression and Complexity

Alexei Miasnikov
(Stevens Institute)

WEBinar "Symbolic Computations and Cryptography",
January 19, 2011

Outline

- Algorithmic problems in groups and crypto.
- Amplifying hardness of algorithmic problems.
- Compression.

Free groups

Let X be a set and $X^{\pm 1} = X \cup X^{-1}$.

A word w in $X^{\pm 1}$ is *reduced* if it does not contain subwords $xx^{-1}, x^{-1}x$ where $x \in X$.

$F(X) =$ the set of all reduced words in $X^{\pm 1}$.

Reduction process: $w \to red(w)$ by subsequent deletions of $xx^{-1}, x^{-1}x$.

A free group $F(X)$ with basis X is the set $F(X)$ with multiplication of $u, v \in F(X)$ defined as reduction of the concatenation of u and v.

Free groups

Let X be a set and $X^{\pm 1} = X \cup X^{-1}$.

A word w in $X^{\pm 1}$ is *reduced* if it does not contain subwords $xx^{-1}, x^{-1}x$ where $x \in X$.

$F(X) =$ the set of all reduced words in $X^{\pm 1}$.

Reduction process: $w \to red(w)$ by subsequent deletions of $xx^{-1}, x^{-1}x$.

A free group $F(X)$ with basis X is the set $F(X)$ with multiplication of $u, v \in F(X)$ defined as reduction of the concatenation of u and v.

Generators and relators

Let $R \subseteq F(X)$. Then

$$ncl(R) = \{\prod_{i=1}^{n} s_i^{-1} r_i s_i \mid r_i \in R^{\pm 1}, s_i \in F(X), n \in \mathbb{N}\}$$

is the smallest normal subgroup of $F(X)$ containing R.

We say that a set of generators X and a set of relators R define the factor group $G = F(X)/ncl(R)$ and write $G = \langle X \mid R \rangle$.

Notice, if $u, v \in F(X)$ then $u = v$ in G if v can be obtained from u by a sequence of insertions and deletions of $xx^{-1}, x^{-1}x, r \in R^{\pm 1}$.

Area and Dehn functions in $G = F(X)/ncl(R)$

A word $w \in F(X)$ is equal to 1 in G if and only if

$$w = \prod_{i=1}^{n} s_i^{-1} r_i^{\pm 1} s_i$$

for some product from $ncl(R)$.

The area $Area(w)$ of w is the minimal such n.

The Dehn function of G is defined by

$$f_G(n) = \max\{Area(w) \mid w \in F(X), |w| = n\}.$$

The Word Problem

Fix a group $G = \langle X \mid R \rangle$.

The Word Problem in G: for a given $w \in (X \cup X^{-1})^*$ verify if $w = 1$ in G or not.

In crypto: encode a bit string $b = b_1 b_2 \dots b_k$ by a word

$$\$w_1\$w_2\$\dots\$w_k$$

where $w_i = 1$ in G if $b_i = 1$ and $w_i \neq 1$ in G if $b_i = 0$.

To break: solve the word problem for w_i in G.

[Magyarik and Wagner, Shpirain-Zapata, etc]

Word Problem in random groups

Theorem [Gromov]

A random finitely presented group is hyperbolic.

A group is hyperbolic if it has a linear Dehn function, i.e., the area $Area(w)$ of every word $w = 1$ in G is bounded by a fixed linear function on the length of w.

The word problem is linear time in hyperbolic groups.

So random choice of a group is not going to work. One needs to choose groups carefully.

Groups with undecidable WP

In 1947 Markov and Post constructed independently first finitely presented semigroups with undecidable EP.

In 1955 Novikov, and soon after W.W. Boone, constructed independently finitely presented groups with undecidable EP.

More examples of undecidable WP

Now there are much shorter examples of semigroups with undecidable word problem constructed by G. S. Tseitin, D. Scott, Matiyasevich, Makanin.

Other examples of groups with undecidable Word Problem are due to J. L. Britton, V.V. Borisov, and D.J. Collins.

An excellent exposition of the results in this area with complete and improved proofs is given in the survey by S.I. Adian and V.G. Durnev.

Groups with undecidable WP

All the classical examples of groups with undecidable WP are based on the same idea:

given a Turing machine M one constructs a group $G(M)$ such that the WP in G simulates the Halting Problem of M.

Groups with undecidable WP

The Halting Problem for M

For a given initial configuration C of the tape decide whether or not M halts when started on C.

Turing: The Halting Problem for a universal Turing machine is undecidable.

Simulation: For a given configuration C one effectively constructs a word w_C in the generators of $G(T)$ such that M halts on C if and only if $w_C = 1$ in G.

Hence, if the Halting Problem for M is undecidable then the WP in $G(M)$ is undecidable.

Groups with undecidable WP

The Halting Problem for M

For a given initial configuration C of the tape decide whether or not M halts when started on C.

Turing: The Halting Problem for a universal Turing machine is undecidable.

Simulation: For a given configuration C one effectively constructs a word w_C in the generators of $G(T)$ such that M halts on C if and only if $w_C = 1$ in G.

Hence, if the Halting Problem for M is undecidable then the WP in $G(M)$ is undecidable.

Groups with undecidable WP

The Halting Problem for M

For a given initial configuration C of the tape decide whether or not M halts when started on C.

Turing: The Halting Problem for a universal Turing machine is undecidable.

Simulation: For a given configuration C one effectively constructs a word w_C in the generators of $G(T)$ such that M halts on C if and only if $w_C = 1$ in G.

Hence, if the Halting Problem for M is undecidable then the WP in $G(M)$ is undecidable.

Classical examples

Theorem [M., Ushakov, Won]

WP is PTime decidable on a generic set of inputs in all classical examples of groups (semigroups) with undecidable WP.

We need something else.

Classical examples

Theorem [M., Ushakov, Won]

WP is PTime decidable on a generic set of inputs in all classical examples of groups (semigroups) with undecidable WP.

We need something else.

Generic complexity

WP is generically decidable in a group G with a finite generating set X if there is a correct partial algorithm A that solves the Word Problem in G on most words from $F(X)$.

That is, the halting set of A is generic with respect to the stratification of $F(X)$ given by the standard length function $|\cdot|$ on $F(X)$.

Generic complexity

WP is generically decidable in a group G with a finite generating set X if there is a correct partial algorithm A that solves the Word Problem in G on most words from $F(X)$.

That is, the halting set of A is generic with respect to the stratification of $F(X)$ given by the standard length function $|\cdot|$ on $F(X)$.

Generic sets

Recall, that $T \subseteq F(X)$ is generic in $F(X)$ if

$$\rho_n(T) = \frac{|T \cap S_n|}{|S_n|} \to 1 \text{ for } n \to \infty,$$

where $S_n = \{w \in F(X) \mid |w| = n\}$.

Furthermore, T is exponentially generic if $\rho_n(T)$ converges to 1 exponentially fast.

Groups with hard WP

Question

What are groups $G = \langle X \mid R \rangle$ with really hard WP?

Known:

- There are groups $G(M)$ with undecidable word problem.
- For a universal Turing machine M the WP in $G(M)$ is as hard as possible: WP(G) is an m-complete c.e. set.
- However, in all classical examples of groups $G(M)$ the WP is linear time decidable on some generic sets of inputs.

Groups with hard WP

Question: Are there finitely (or recursively) presented groups with WP hard on most inputs?

Theorem [Hamkins and Myasnikov]

The halting problem for Turing machines is easy (linear time) on most inputs.

Groups with hard WP

Question: Are there finitely (or recursively) presented groups with WP hard on most inputs?

Theorem [Hamkins and Myasnikov]

The halting problem for Turing machines is easy (linear time) on most inputs.

Generic Complexity of the Word Problem

Theorem [Grigorchuk, Kesten]

Let $G = \langle X \mid R \rangle$ be an infinite group. Then

- The set of nontrivial elements in G is generic.
- If G is non-amenable then the set of nontrivial elements in G is exponentially generic.

Generic Complexity of the Word Problem

Theorem [Grigorchuk, Kesten]

Let $G = \langle X \mid R \rangle$ be an infinite group. Then

- The set of nontrivial elements in G is generic.
- If G is non-amenable then the set of nontrivial elements in G is exponentially generic.

Generic Complexity of the Word Problem

Theorem [Miasnikov, Ushakov]

Let $G = \langle X \mid R \rangle$ be a group a group given by a reduced symmetrized presentation. Then a random word w such that $w = 1$ is "hyperbolic", i.e., $Area(w)$ is bounded by a fixed linear function.

Theorem [Miasnikov, Ushakov]

Let $G = \langle X \mid R \rangle$ be a group given by a reduced symmetrized presentation. Then the Search Word Problem in G is in PTime on random inputs w (with $w = 1$ in G).

A presentation $\langle X \mid R \rangle$ is reduced if the generators $x \in X$ are non-trivial, and relations $r \in R$ do not contain proper trivial subwords.

So random choice of $w_i = 1$ is not going to work.

Generic Complexity of the Word Problem

Theorem [Miasnikov, Ushakov]

Let $G = \langle X \mid R \rangle$ be a group a group given by a reduced symmetrized presentation. Then a random word w such that $w = 1$ is "hyperbolic", i.e., $Area(w)$ is bounded by a fixed linear function.

Theorem [Miasnikov, Ushakov]

Let $G = \langle X \mid R \rangle$ be a group given by a reduced symmetrized presentation. Then the Search Word Problem in G is in PTime on random inputs w (with $w = 1$ in G).

A presentation $\langle X \mid R \rangle$ is reduced if the generators $x \in X$ are non-trivial, and relations $r \in R$ do not contain proper trivial subwords.

So random choice of $w_i = 1$ is not going to work.

Amplification of hardness

I will discuss three types of amplification:

- Amplification by cloning.
- Amplification via immune sets.
- Amplification by compression.

Amplification of hardness

I will discuss three types of amplification:

- Amplification by cloning.
- Amplification via immune sets.
- Amplification by compression.

Amplification of hardness

I will discuss three types of amplification:

- Amplification by cloning.
- Amplification via immune sets.
- Amplification by compression.

Amplification in semigroups

Let

$$\mathfrak{S} = \langle a_1, \ldots, a_n | r_1 = s_1, \ldots, r_k = s_k \rangle = \langle A \mid R \rangle$$

For a letter $x \notin A$ put

$$\mathfrak{S}_x = \langle A, x | R, x = xa_1, \ldots, x = xa_n, x = xx \rangle.$$

Theorem [Myasnikov, Rybalov]

If the word problem in $\mathfrak{S}$ is undecidable then the word problem in $\mathfrak{S}_x$ is super-undecidable (undecidable on every generic subset of inputs).

An easy reduction

Let $A_x = A \cup \{x\}$.

Lemma

For any $u, v \in A^*$ and $u', v' \in A_x^*$ the following holds:

$$uxu' = vxv' \text{ in } \mathfrak{S} \Leftrightarrow \ u = v \text{ in } \mathfrak{S}_x.$$

The Word Problem in $\mathfrak{S}_x$ is Linear Time reducible to the Word Problem in $\mathfrak{S}$.

So what are we gaining here? A lot of hard inputs!

An easy reduction

Let $A_x = A \cup \{x\}$.

Lemma

For any $u, v \in A^*$ and $u', v' \in A_x^*$ the following holds:

$$uxu' = vxv' \text{ in } \mathfrak{S} \Leftrightarrow \ u = v \text{ in } \mathfrak{S}_x.$$

The Word Problem in $\mathfrak{S}_x$ is Linear Time reducible to the Word Problem in $\mathfrak{S}$.

So what are we gaining here? A lot of hard inputs!

An easy reduction

Let $A_x = A \cup \{x\}$.

Lemma

For any $u, v \in A^*$ and $u', v' \in A_x^*$ the following holds:

$$uxu' = vxv' \text{ in } \mathfrak{S} \Leftrightarrow \ u = v \text{ in } \mathfrak{S}_x.$$

The Word Problem in $\mathfrak{S}_x$ is Linear Time reducible to the Word Problem in $\mathfrak{S}$.

So what are we gaining here? A lot of hard inputs!

The essence of amplification

The essence of amplification: cloning

The clone of $(u, v) \in A^* \times A^*$ is

$$C(u, v) = \{(uxp, vxq) \mid p, q \in A_x^*\}.$$

Lemma

For every $(u, v) \in A^* \times A^*$ the clone $C(u, v)$ is computably enumerable and non-negligible.

Non-negligible here: the size of a pair $(p, q) \in A_x^* \times A_x^*$ is equal to $|p| + |q|$.

The probability to hit a pair from $C(u, v)$ (u, v are fixed) of a given length n is

$$\frac{1}{|A_x|^{|u|+|v|+2}} + o(n)$$

The essence of amplification

The essence of amplification: cloning

The clone of $(u, v) \in A^* \times A^*$ is

$$C(u, v) = \{(uxp, vxq) \mid p, q \in A_x^*\}.$$

Lemma

For every $(u, v) \in A^* \times A^*$ the clone $C(u, v)$ is computably enumerable and non-negligible.

Non-negligible here: the size of a pair $(p, q) \in A_x^* \times A_x^*$ is equal to $|p| + |q|$.

The probability to hit a pair from $C(u, v)$ (u, v are fixed) of a given length n is

$$\frac{1}{|A_x|^{|u|+|v|+2}} + o(n)$$

The essence of amplification

The main idea: For a partial algorithm M for WP in $\mathfrak{S}_x$ design a new algorithm M_x for WP in $\mathfrak{S}$:

for $(u, v) \in A^* \times A^*$ start enumerating the clone

$$C(u, v) = \{(u_1, v_1), (u_2, v_2), \ldots\}$$

and apply M to each instance (u_i, v_i).

Tseitin Example

Example

In 1956 Tseitin constructed a semigroup $\mathfrak{T}$ presented by 5 generators and 7 relations with unsolvable word problem:

$$\mathfrak{T} = \langle a, b, c, d, e \mid ca, ad = da, bc = cb, bd = db,$$

$$ce = eca, de = edb, cca = ccae \rangle.$$

In this case the super-undecidable semigroup $\mathfrak{T}_x$ has 6 generators and 13 relators whose total length is equal to 49.

Amplification in groups

Theorem (Gilman, Myasnikov, Osin)

Let G be a finitely presented amenable group with an unsolvable word problem. The word problem for G is not solvable on any exponentially generic set of inputs.

Finitely presented amenable groups with an unsolvable word problems exist [Kharlampovich].

Algorithmically finite groups

A finitely generated group G is algorithmically finite if there is no any algorithmic way to produce an infinite set of pairwise distinct elements of G.

More precisely, let G be be a group generated by a finite set X and $\eta : F(X) \to G$ the canonical projection.

Definition

A group G is algorithmically finite if for every infinite computably enumerable subset $W \subseteq F(X)$ there exist at least two distinct words $u, v \in W$ such that $\eta(u) = \eta(v)$.

Algorithmically finite groups

A finitely generated group G is algorithmically finite if there is no any algorithmic way to produce an infinite set of pairwise distinct elements of G.

More precisely, let G be be a group generated by a finite set X and $\eta : F(X) \to G$ the canonical projection.

Definition

A group G is algorithmically finite if for every infinite computably enumerable subset $W \subseteq F(X)$ there exist at least two distinct words $u, v \in W$ such that $\eta(u) = \eta(v)$.

Algorithmically finite groups

Independence of generators

If a finitely generated group G is algorithmically finite with respect to some finite generating set X then it is algorithmically finite with respect to any finite generating set of G.

Hence, algorithmic finiteness is a property of a group, not a presentation.

Clearly every finite group is algorithmically finite.

Dehn Monsters

Theorem [Myasnikov, Osin]

There exists a recursively presented infinite algorithmically finite group.

We will see later that WP in algorithmically finite groups is decidable only on negligible sets of inputs (sets of measure zero).

Motivated by this observation, we call recursively presented infinite algorithmically finite groups Dehn monsters.

Dehn Monsters

The proof of this result is based on new ideas and does not interpret any machines.

Instead, it uses Golod-Shafarevich presentations as a tool to control consequences of relations and simple and immune sets from computability theory.

These groups are not finitely presented.

What is decidable in a Dehn monster?

Theorem

Let $G = \langle X \mid R \rangle$ be a Dehn monster. Then:

- Every computably enumerable subset $W \subseteq F(X)$ with decidable WP in G is negligible, i.e.,

$$\lim_{n\to\infty} \rho_n(W) = 0.$$

- If G is non-amenable, then W is exponentially negligible, i.e., there exists $t > 1$ such that

$$\rho_n(W) = O(t^{-n}).$$

Here, as before, $\rho_n(W) = \frac{|W \cap S_n|}{|S_n|}$.

Golod-Shafarevich and algorithmically finite groups

Theorem [Myasnikov, Osin]

For every Golod-Shafarevich group $\langle X \mid S \rangle$ there exists a simple set of relations $R \subseteq F(X)$ such that the quotient $\langle X \mid S \cup R \rangle$ is again Golod-Shafarevich algorithmically finite group.

Corollary

There exists a recursively presented non-amenable algorithmically finite group (Dehn Monsters).

Compression and amplification

Idea: compress inputs to force the algorithm to work with short images of very long inputs, so the time allocated for computation is much smaller.

Compression in crypto:

- Compress your keys
- Public manipulation with images should be efficient
- Legitimate decoding should be efficient

Compression of integers

In Peano arithmetic, as well as in ZFC axiomatic of set theory, natural numbers are presented in the unary form.

$n = 11\ldots1$ (here 1 occurs n times), so the length of the representation is n.

Binary representation gives an exponential compression: $n = b_1 b_2 \ldots b_k$, where $b_i \in \{0, 1\}$ and $k \sim log_2 n$.

Prime factorization in unary

Prime factorization is in PTime if the numbers are given in the unary notation.

Perhaps, Prime Factorization is not a number theory problem!

Plandowski Compression

Compress words by straight line programs.

This method resembles Lempel-Ziv compression.

A straight line program P:

$$
\begin{aligned}
X_9 &= X_8X_7 && babbababbabbababbababbabbababbabba = w(P)\\
X_8 &= X_7X_6 && babbababbabbababbabab\\
X_7 &= X_6X_5 && babbababbabba\\
X_6 &= X_5X_4 && babbabab\\
X_5 &= X_4X_3 && babba\\
X_4 &= X_3X_2 && bab\\
X_3 &= X_2X_1 && ba\\
X_2 &= b\\
X_1 &= a
\end{aligned}
$$

P computes a single word, $w(P)$.

The size of P is the number of lines of code.

$a^n = w(P')$ for some P' of size $O(\log n)$.

Computations

It is convenient to operate on compressed words without decompressing them.

The following operations are feasible for compressed words.

1. Compute the length of $w(P)$
2. Find the i^{th} letter of $w(P)$
3. Compressed string matching: Given P_1 and P_2, find the first occurrence and the number of occurrences of $w(P_1)$ in $w(P_2)$.

Applications to group theory

The word problem for $Aut(F_n)$ can be solved in polynomial time. (Schleimer, 2007).

Let $\phi = \phi_1\phi_2 \dots \phi_k \in Aut(G)$ be a product of elementary automorphisms.

Then the image $\phi(w)$ on a word w can be exponentially long relative to the length of w

But it can be presented by a straight program of size $|w|$.

Inverting automorphisms

Fix a group $G = \langle X \mid R \rangle$.

Inverting automorphisms: Given an automorphism $\phi \in Aut(G)$ by its images on the generators $w_i = \phi(x_i), x_i \in X$ find the inverse automorphism ϕ^{-1} (its images on the generators).

In free groups: Nielsen, Whitehead,

It is known to be in PTime in free groups.

Inverting automorphisms in crypto

In crypto:

Let E be a finite generating set of $Aut(G)$. For every automorphism from E its inverse is known.

- choose a sequence $\phi_1, \phi_2, \ldots, \phi_k$ of $\phi_i \in E$.
- compose them into $\phi = \phi_1\phi_2 \ldots \phi_k \in Aut(G)$. Notice, that $\phi^{-1} = \phi_k^{-1} \ldots \phi_1^{-1}$ is easy to find (knowing $\phi_1, \phi_2, \ldots, \phi_k$).
- encode w by $\phi(w)$.
- Decode w by applying ϕ^{-1} to $\phi(w)$.

[Moh, Romankov, ...]

Inverting compressed automorphisms

As above, let E be a finite set of generators of $Aut(F)$ where F is a free group with basis X.

Let $\phi = \phi_1\phi_2 \dots \phi_k$, where $\phi \in E$.

If ϕ is given by the compressed images on the generators from X, then complexity of the inverting of ϕ is unknown (conjectured to be exponential).

Compressing tuples of elements

Non-deterministic straight line programs (NSLP) allow several productions of the type $X_i \to X_j X_k$ with the same left-hand part.

Her we view the straight line programs as a set of productions (not as circuits).

The size $|P|$ of P is the number of productions in P.

Let P be a NSLP. Every derivation in P yields a word in the alphabet of terminals A.

By $Eval(P)$ we denote the set of all words in A^* produced by P.

Compressing tuples of elements

Lemma

Let $U = (P_1, \ldots, P_n)$ be a tuple of deterministic SLP, $Eval(P_i) = u_i$. Then there exists a NSLP P_U such that

$$Eval(P_U) = \{u_1^{\varepsilon_1} u_2^{\varepsilon_2} \ldots u_n^{\varepsilon_n} \mid \varepsilon_i \in \{0, 1\}\}.$$

Moreover, $|P_U| = |P_1| + \ldots + |P_n| + 2n$.

Compressed triviality problem

Let G be a group generated by a finite set A.

Compressed triviality problem in G: given a NSLP P over A determine if $Eval(P)$ contains a word that defines the identity in G.

Theorem

Let $G = \langle A \rangle$ be a group with an element of infinite order. Then the classical Subset Sum Problem is polynomial time reducible to the Compressed Triviality Problem in G.

Compressed triviality problem

Let G be a group generated by a finite set A.

Compressed triviality problem in G: given a NSLP P over A determine if $Eval(P)$ contains a word that defines the identity in G.

Theorem

Let $G = \langle A \rangle$ be a group with an element of infinite order. Then the classical Subset Sum Problem is polynomial time reducible to the Compressed Triviality Problem in G.

Reduction in free products

Let $G = \langle A \rangle$ and $H = \langle B \rangle$ be groups.

Then elements in the free product $G * H$ can be presented in the form

$$w = u_1 v_1 u_2 v_2 \dots u_k v_k$$

where $u_i \in A^*, v_i \in B^*$, $i = 1, \dots, k$.

The word is reduced if all u_i, v_j are non-trivial in the corresponding factors.

Reduction in free products

Let P and Q are NSLP such that $Eval(P) = (u_1, \ldots, u_k)$ and $Eval(Q) = (v_1, \ldots, v_m)$.

(P, Q) represents the word $u_1 v_1 \ldots u_k v_m v_{k+1} \ldots v_m$ if $k \leq m$; or the word $u_1 v_1 \ldots u_k v_m u_{k+1} \ldots u_m$, otherwise.

We denote this word by $w(P, Q)$.

Complexity of the reduction problem

Compressed Reducibility Problem in $G * H$: given two NSLP P, Q over the alphabets A, B verify if the word $w(P, Q)$ is reduced or not.

Theorem

If *GandH* have elements of infinite order then the Compressed Reducibility Problem is NP-hard.

Compressed MP in free groups

Compressed MP: given a subgroup H by its compressed generators in a free group F and a compressed word w determine if $w \in H$ or not.

Complexity of the Compressed MP in F is not known, it seems very hard.

If so, The Length Based Attack is not efficient.

Super compression in integers

One can represent integers by algebraic circuits with operations $+, -, x \cdot 2^y$.

Such circuits represent numbers in finite towers of exponents.

Theorem [Myasnikov, Ushakov]

- For each n there is a unique circuit P_n computing n;
- This P_n can be found quickly.
- One can do manipulations over circuits in PTime