Generic complexity of the subset sum problem for some semigroups

Alexander Rybalov

Mathematical Center in Akademgorodok, Omsk

29 october 2020

Classical Subset sum problem

The subset sum problem (SSP)

- INPUT: Natural numbers $a_1, a_2, \ldots, a_n$ and S in binary representation.
- OUTPUT: YES, if $\exists\ I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = S$,
  NO, otherwise.

Classical Subset sum problem

The subset sum problem (SSP)

- INPUT: Natural numbers $a_1, a_2, \ldots, a_n$ and S in binary representation.
- OUTPUT: YES, if $\exists\ I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = S$,
  NO, otherwise.

Theorem (Karp, 1972)

SSP is NP-complete.

SSP and cryptography

- Merkle–Hellman knapsack cryptosystem (1978), based on SSP.

SSP and cryptography

- Merkle–Hellman knapsack cryptosystem (1978), based on SSP.
- Shamir (1982) polynomial attack on Merkle–Hellman cryptosystem, used a reduction to the problem of finding the shortest vector in lattice and Lenstra-Lenstra-Lovasz algorithm.

SSP and cryptography

- Merkle–Hellman knapsack cryptosystem (1978), based on SSP.
- Shamir (1982) polynomial attack on Merkle–Hellman cryptosystem, used a reduction to the problem of finding the shortest vector in lattice and Lenstra-Lenstra-Lovasz algorithm.
- Chor-Rivest cryptosystem (1985) - still unbroken.

Complexity of SSP for almost all inputs

- Lagarias, Odlyzko (1985): a polynomial algorithm for almost all inputs of low-density SSP.

Complexity of SSP for almost all inputs

- Lagarias, Odlyzko (1985): a polynomial algorithm for almost all inputs of low-density SSP.
- **Density** is $d = n/\log(\max a_i)$. The algorithm works for $d < 0.645$.

Complexity of SSP for almost all inputs

- Lagarias, Odlyzko (1985): a polynomial algorithm for almost all inputs of low-density SSP.
- **Density** is $d = n/\log(\max a_i)$. The algorithm works for $d < 0.645$.
- Main idea: reduction to to the problem of finding the shortest vector in lattice and Lenstra-Lenstra-Lovasz algorithm.

SSP in groups and semigroups

Myasnikov, Nikolaev, Ushakov (2015) defined an analog of SSP for any group (semigroup) G:

SSP in groups and semigroups

Myasnikov, Nikolaev, Ushakov (2015) defined an analog of SSP for any group (semigroup) G:

SSP over G

- INPUT: Elements $g_1, g_2, \ldots, g_n$ and g.
- OUTPUT: YES, if there exist $1 \leq i_1 < \ldots < i_k \leq n$ such that $g_{i_1} \ldots g_{i_k} = g$,
  NO, otherwise.

Complexity of SSP for groups

SSP in P for the following groups:

- Hyperbolic groups (Myasnikov, Nikolaev, Ushakov).

Complexity of SSP for groups

SSP in P for the following groups:

- Hyperbolic groups (Myasnikov, Nikolaev, Ushakov).
- Nilpotent finitely generated groups (Myasnikov, Nikolaev, Ushakov).

SSP is NP-complete for the following groups:

- Baumslag-Solitair group $BS(1,2)$ (Myasnikov, Nikolaev, Ushakov).

Complexity of SSP for groups

SSP in P for the following groups:

- Hyperbolic groups (Myasnikov, Nikolaev, Ushakov).
- Nilpotent finitely generated groups (Myasnikov, Nikolaev, Ushakov).

SSP is NP-complete for the following groups:

- Baumslag-Solitair group $BS(1,2)$ (Myasnikov, Nikolaev, Ushakov).
- Thompson's group (Myasnikov, Nikolaev, Ushakov).

Complexity of SSP for groups

SSP in P for the following groups:

- Hyperbolic groups (Myasnikov, Nikolaev, Ushakov).
- Nilpotent finitely generated groups (Myasnikov, Nikolaev, Ushakov).

SSP is NP-complete for the following groups:

- Baumslag-Solitair group $BS(1,2)$ (Myasnikov, Nikolaev, Ushakov).
- Thompson's group (Myasnikov, Nikolaev, Ushakov).
- Lamplighter group (Mishchenko, Treier).

Complexity of SSP for groups

SSP in P for the following groups:

- Hyperbolic groups (Myasnikov, Nikolaev, Ushakov).
- Nilpotent finitely generated groups (Myasnikov, Nikolaev, Ushakov).

SSP is NP-complete for the following groups:

- Baumslag-Solitair group $BS(1,2)$ (Myasnikov, Nikolaev, Ushakov).
- Thompson's group (Myasnikov, Nikolaev, Ushakov).
- Lamplighter group (Mishchenko, Treier).
- Some polycyclic groups (König, Lohrey, Zetzsche).

Generic-case approach

Kapovich, Myasnikov, Schupp and Shpilrain in 2003 developed generic approach to algorithmic problems, which considers an algorithmic problem on "most"of the inputs (i.e., on a generic set) instead of the entire domain and ignores it on the rest of inputs (a negligible set). It turned out, that many famous undecidable problems are easily decidable on most inputs.

Generic sets

Let I be the set of all inputs and I_n be the set of all inputs of size n (sphere of radius n). For a subset $S \subseteq I$ define the following sequence

$$\rho_n(S) = \frac{|S \cap I_n|}{|I_n|}, \ n = 1, 2, 3, \ldots$$

The asymptotic density of set S is the following limit (if it exists)

$$\rho(S) = \lim_{n \to \infty} \rho_n(S).$$

S is called **generic** if $\rho(S) = 1$ and **negligible** if $\rho(S) = 0$.

Generic algorithms

Algorithm $\mathcal{A} : I \to I \cup \{?\}$ is called **generic** if

1. $\mathcal{A}$ halts on all inputs from I,
2. set $\{x \in I : \mathcal{A}(x) = ?\}$ is negligible.

SSP over monoid $SL(2, \omega)$

- ω is the set of natural numbers with 0.

SSP over monoid $SL(2, \omega)$

- ω is the set of natural numbers with 0.
- Monoid $SL(2, \omega)$ is the set of all matrices of order 2 with entries from ω with determinant 1.

SSP over monoid $SL(2, \omega)$

- ω is the set of natural numbers with 0.
- Monoid $SL(2, \omega)$ is the set of all matrices of order 2 with entries from ω with determinant 1.
- Size of matrix $M \in SL(2, \omega)$ is the maximum of binary length of its entries.

SSP over monoid $SL(2, \omega)$

- ω is the set of natural numbers with 0.
- Monoid $SL(2, \omega)$ is the set of all matrices of order 2 with entries from ω with determinant 1.
- Size of matrix $M \in SL(2, \omega)$ is the maximum of binary length of its entries.
- In input $(M_1, \ldots, M_n, M)$ for SSP of size n all M_i have size $\leq n$, and M has size $\leq n^2$.

NP-completeness

Proposition

SSP over $SL(2, \omega)$ is NP-complete.

Proposition

SSP over $SL(2, \omega)$ is NP-complete.

$(a_1, \ldots, a_n, S)$ – instance of classical SSP.

$$a_1 \to \begin{pmatrix} 1 & a_1 \\ 0 & 1 \end{pmatrix}, \ldots, a_n \to \begin{pmatrix} 1 & a_n \\ 0 & 1 \end{pmatrix}, S \to \begin{pmatrix} 1 & S \\ 0 & 1 \end{pmatrix}.$$

Proposition

SSP over $SL(2, \omega)$ is NP-complete.

$(a_1, \ldots, a_n, S)$ – instance of classical SSP.

$$a_1 \to \begin{pmatrix} 1 & a_1 \\ 0 & 1 \end{pmatrix}, \ldots, a_n \to \begin{pmatrix} 1 & a_n \\ 0 & 1 \end{pmatrix}, S \to \begin{pmatrix} 1 & S \\ 0 & 1 \end{pmatrix}.$$

$a_{i_1} + \ldots + a_{i_k} = S \Leftrightarrow$

$$\begin{pmatrix} 1 & a_{i_1} \\ 0 & 1 \end{pmatrix} \times \ldots \times \begin{pmatrix} 1 & a_{i_k} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & S \\ 0 & 1 \end{pmatrix}.$$

Generic complexity of SSP over $SL(2, \omega)$

Theorem

SSP over $SL(2, \omega)$ is generically decidable in polynomial time.

Structure of $SL(2, \omega)$

Theorem (Nielsen, 1924)

$SL(2, \omega)$ is free monoid with two generators:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Structure of $SL(2,\omega)$

Theorem (Nielsen, 1924)

$SL(2,\omega)$ is free monoid with two generators:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Finding of the word representation for matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ by subtractive Euclidean algorithm:

$$A^{-1}M = \begin{pmatrix} a-c & b-d \\ c & d \end{pmatrix}, \; B^{-1}M = \begin{pmatrix} a & b \\ c-a & d-b \end{pmatrix}.$$

$$M \to M_1 \to M_2 \to \ldots \to M_t = E.$$

Simple example

1. $M = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix}$,

Simple example

1. $M = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix}$,
2. $A^{-1}M = \begin{pmatrix} 1 & 1 \\ 2 & 3 \end{pmatrix}$,

Simple example

1. $M = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix}$,
2. $A^{-1}M = \begin{pmatrix} 1 & 1 \\ 2 & 3 \end{pmatrix}$,
3. $B^{-1}A^{-1}M = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$,

Simple example

1. $M = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix}$,
2. $A^{-1}M = \begin{pmatrix} 1 & 1 \\ 2 & 3 \end{pmatrix}$,
3. $B^{-1}A^{-1}M = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$,
4. $B^{-1}B^{-1}A^{-1}M = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$,

Simple example

1. $M = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix}$,
2. $A^{-1}M = \begin{pmatrix} 1 & 1 \\ 2 & 3 \end{pmatrix}$,
3. $B^{-1}A^{-1}M = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$,
4. $B^{-1}B^{-1}A^{-1}M = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$,
5. $A^{-1}B^{-1}B^{-1}A^{-1}M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = E$,

Simple example

1. $M = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix}$,
2. $A^{-1}M = \begin{pmatrix} 1 & 1 \\ 2 & 3 \end{pmatrix}$,
3. $B^{-1}A^{-1}M = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$,
4. $B^{-1}B^{-1}A^{-1}M = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$,
5. $A^{-1}B^{-1}B^{-1}A^{-1}M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = E$,

So $M = ABBA$.

Word representation is short for almost all matrices

Lemma

The set of matrices $M \in SL(2, \omega)$ such that its word representation $\leq size(M)^2$, is generic.

Word representation is short for almost all matrices

Lemma

The set of matrices $M \in SL(2, \omega)$ such that its word representation $\leq size(M)^2$, is generic.

Theorem (Knuth, Yao, 1975)

Let $t(m, n)$ be the number of steps of subtractive Euclidean algorithm on input (m, n). Then

$$\sum_{m \leq n} t(m, n) = \frac{6}{\pi} n (\log n)^2 + O(n \log n (\log \log n)^2).$$

Word representation is short for almost all matrices

Lemma

The set of matrices $M \in SL(2, \omega)$ such that its word representation $\leq size(M)^2$, is generic.

Theorem (Knuth, Yao, 1975)

Let $t(m, n)$ be the number of steps of subtractive Euclidean algorithm on input (m, n). Then

$$\sum_{m \leq n} t(m, n) = \frac{6}{\pi} n (\log n)^2 + O(n \log n (\log \log n)^2).$$

Gurevich (1990) used this fact for constructing average-case NP-complete problems over group $SL(2, \mathbb{Z})$.

SSP for the free monoid

Lemma

SSP over the free monoid $\{A, B\}^*$ is decidable in polynomial time.

SSP for the free monoid

Lemma

SSP over the free monoid $\{A, B\}^*$ is decidable in polynomial time.

Dynamic programming: algorithm $\mathcal{A}$ works on $(w_1, \ldots, w_n, w)$ in the following way

1. If w is empty, stop all recursive calls of algorithm $\mathcal{A}$ and output YES.
2. Find all $w_{i_1}, \ldots, w_{i_k}$ such that w_{i_l} is a prefix of w. If there is no such words, output NO.
3. Run $\mathcal{A}$ on every subproblem $(w_{i_l+1}, \ldots, w_n, w_{i_l}^{-1} w)$, $l = 1, \ldots, k$ for which the algorithm $\mathcal{A}$ has not been run.
4. If all recursive calls of $\mathcal{A}$ halt and output NO, output NO.

SSP for the free monoid

Lemma

SSP over the free monoid $\{A, B\}^*$ is decidable in polynomial time.

Dynamic programming: algorithm $\mathcal{A}$ works on $(w_1, \ldots, w_n, w)$ in the following way

1. If w is empty, stop all recursive calls of algorithm $\mathcal{A}$ and output YES.
2. Find all $w_{i_1}, \ldots, w_{i_k}$ such that w_{i_l} is a prefix of w. If there is no such words, output NO.
3. Run $\mathcal{A}$ on every subproblem $(w_{i_l+1}, \ldots, w_n, w_{i_l}^{-1} w)$, $l = 1, \ldots, k$ for which the algorithm $\mathcal{A}$ has not been run.
4. If all recursive calls of $\mathcal{A}$ halt and output NO, output NO.

Number of possible subproblems $\leq n \cdot |w|$.

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$

$SL(2,\mathbb{Z})$ is the group of unimodular integer matrices of order 2.
$PSL(2,\mathbb{Z}) = SL(2,\mathbb{Z})/\{E,-E\}$ – the modular group.

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$

$SL(2,\mathbb{Z})$ is the group of unimodular integer matrices of order 2.
$PSL(2,\mathbb{Z}) = SL(2,\mathbb{Z})/\{E,-E\}$ – the modular group.

Proposition

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$ are NP-complete.

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$

$SL(2,\mathbb{Z})$ is the group of unimodular integer matrices of order 2.
$PSL(2,\mathbb{Z}) = SL(2,\mathbb{Z})/\{E,-E\}$ – the modular group.

Proposition

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$ are NP-complete.

Theorem

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$ are generically decidable in polynomial time.

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$

1. Transform matrix representation to words over A, B, A^{-1}, B^{-1} by subtractive Euclidean algorithm.

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$

1. Transform matrix representation to words over A, B, A^{-1}, B^{-1} by subtractive Euclidean algorithm.
2. Transform to words over generators

$$S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \; R = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}.$$

SSP over $SL(2,\mathbb{Z})$ and $PSL(2,\mathbb{Z})$

1. Transform matrix representation to words over A, B, A^{-1}, B^{-1} by subtractive Euclidean algorithm.
2. Transform to words over generators

$$S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \; R = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}.$$

3. Groups

$$PSL(2,\mathbb{Z}) = \langle S, R : S^2 = 1, R^3 = 1 \rangle$$

$$SL(2,\mathbb{Z}) = \langle S, R : S^4 = 1, R^6 = 1, S^2 = R^3 \rangle$$

are hyperbolic, we can apply Myasnikov-Nikolaev-Ushakov algorithm.

SSP over $Mat(k, \omega)$

- $Mat(k, \omega)$ is monoid of $k \times k$ matrices over ω.

SSP over $Mat(k, \omega)$

- $Mat(k, \omega)$ is monoid of $k \times k$ matrices over ω.
- Size of matrix $M \in Mat(k, \omega)$ is the maximum of binary length of its entries.

SSP over $Mat(k, \omega)$

- $Mat(k, \omega)$ is monoid of $k \times k$ matrices over ω.
- Size of matrix $M \in Mat(k, \omega)$ is the maximum of binary length of its entries.
- In input $(M_1, \ldots, M_n, M)$ for SSP of size n all M_i have size $\leq n$, and M has size $\leq n^2$.

SSP over $Mat(k, \omega)$

- $Mat(k, \omega)$ is monoid of $k \times k$ matrices over ω.
- Size of matrix $M \in Mat(k, \omega)$ is the maximum of binary length of its entries.
- In input $(M_1, \ldots, M_n, M)$ for SSP of size n all M_i have size $\leq n$, and M has size $\leq n^2$.

Proposition

SSP over $Mat(k, \omega)$ are NP-complete.

SSP over $Mat(k, \omega)$

- $Mat(k, \omega)$ is monoid of $k \times k$ matrices over ω.
- Size of matrix $M \in Mat(k, \omega)$ is the maximum of binary length of its entries.
- In input $(M_1, \ldots, M_n, M)$ for SSP of size n all M_i have size $\leq n$, and M has size $\leq n^2$.

Proposition

SSP over $Mat(k, \omega)$ are NP-complete.

Theorem

SSP over $Mat(k, \omega)$ are generically decidable in polynomial time.

SSP over $Mat(k, \omega)$

Lemma

Let S be the set of $M \in Mat(k, \omega)$ such that equation $M = XY$ has more than $p(n) = (2(n+1))^{k^2+1}$ solutions $X, Y \in Mat(k, \omega)$, $size(X), size(Y) \leq n$, where $n = size(M)$. Then S is negligible.

SSP over $Mat(k, \omega)$

Lemma

Let S be the set of $M \in Mat(k, \omega)$ such that equation $M = XY$ has more than $p(n) = (2(n+1))^{k^2+1}$ solutions $X, Y \in Mat(k, \omega)$, $size(X), size(Y) \leq n$, where $n = size(M)$. Then S is negligible.

Generic polynomial algorithm $\mathcal{A}$ works on $(M_1, \ldots, M_n, M)$ in the following way:

SSP over $Mat(k, \omega)$

1. If $det(M) = 0$, output «?» .

SSP over $Mat(k, \omega)$

1. If $det(M) = 0$, output «?» .
2. Delete singular matrices from $M_1, \ldots, M_n$.

SSP over $Mat(k, \omega)$

1. If $det(M) = 0$, output «?» .
2. Delete singular matrices from $M_1, \ldots, M_n$.
3. Counter of recursive calls $R := 0$.

SSP over $Mat(k, \omega)$

1. If $det(M) = 0$, output «?» .
2. Delete singular matrices from $M_1, \ldots, M_n$.
3. Counter of recursive calls $R := 0$.
4. If $R = np(n^2)$, stop all recursive calls and output «?».

SSP over $Mat(k, \omega)$

1. If $det(M) = 0$, output «?» .
2. Delete singular matrices from $M_1, \ldots, M_n$.
3. Counter of recursive calls $R := 0$.
4. If $R = np(n^2)$, stop all recursive calls and output «?».
5. If $M = E$, stop all recursive calls and output YES.

SSP over $Mat(k, \omega)$

1. If $det(M) = 0$, output «?» .
2. Delete singular matrices from $M_1, \ldots, M_n$.
3. Counter of recursive calls $R := 0$.
4. If $R = np(n^2)$, stop all recursive calls and output «?».
5. If $M = E$, stop all recursive calls and output YES.
6. For every M_i, $i = 1, \ldots, n$ solve equation $M_i X = M$. If it has solution in $Mat(k, \omega)$, then run $\mathcal{A}(M_{i+1}, \ldots, M_n, X)$ for which the algorithm $\mathcal{A}$ has not been run and $R := R + 1$.

SSP over $Mat(k, \omega)$

1. If $det(M) = 0$, output «?» .
2. Delete singular matrices from $M_1, \ldots, M_n$.
3. Counter of recursive calls $R := 0$.
4. If $R = np(n^2)$, stop all recursive calls and output «?».
5. If $M = E$, stop all recursive calls and output YES.
6. For every M_i, $i = 1, \ldots, n$ solve equation $M_i X = M$. If it has solution in $Mat(k, \omega)$, then run $\mathcal{A}(M_{i+1}, \ldots, M_n, X)$ for which the algorithm $\mathcal{A}$ has not been run and $R := R + 1$.
7. If there are no solutions, output NO.

SSP over $Mat(k, \omega)$

1. If $det(M) = 0$, output «?» .
2. Delete singular matrices from $M_1, \ldots, M_n$.
3. Counter of recursive calls $R := 0$.
4. If $R = np(n^2)$, stop all recursive calls and output «?».
5. If $M = E$, stop all recursive calls and output YES.
6. For every M_i, $i = 1, \ldots, n$ solve equation $M_i X = M$. If it has solution in $Mat(k, \omega)$, then run $\mathcal{A}(M_{i+1}, \ldots, M_n, X)$ for which the algorithm $\mathcal{A}$ has not been run and $R := R + 1$.
7. If there are no solutions, output NO.
8. If all recursive calls of $\mathcal{A}$ halt and output NO, output NO.

Brandt semigroups

Let G be a group, I a set with at least 2 elements, and $0 \notin G \cup I$. Define a multiplication operation on the set $B(G, I) = I \times G \times I \cup \{0\}$ as follows:

$$(i, g, j)(k, h, l) = \begin{cases} (i, gh, l), & \text{if } j = k, \\ 0, & \text{otherwise,} \end{cases}$$

for all $i, j, k, l \in I$ and all $g, h \in G$. Also $0x = 0$ and $x0 = 0$ for all $x \in B(G, I)$. The set $B(G, I)$ with defined multiplication is called the **Brandt semigroup over the group** G with **index set** I.

Brandt semigroups

Let G be a group, I a set with at least 2 elements, and $0 \notin G \cup I$. Define a multiplication operation on the set $B(G, I) = I \times G \times I \cup \{0\}$ as follows:

$$(i, g, j)(k, h, l) = \begin{cases} (i, gh, l), & \text{if } j = k, \\ 0, & \text{otherwise,} \end{cases}$$

for all $i, j, k, l \in I$ and all $g, h \in G$. Also $0x = 0$ and $x0 = 0$ for all $x \in B(G, I)$. The set $B(G, I)$ with defined multiplication is called the **Brandt semigroup over the group** G with **index set** I.
We are interested in Brandt semigroup $B(G, \mathbb{N})$ with index set $\mathbb{N}$ of natural numbers. Also we will use Brandt semigroup $B(E, \mathbb{N})$ over the trivial one-element group E, which we will denote by $B(\mathbb{N})$ and its elements are just pairs (i, j) of natural numbers.

SSP over Brandt semigroups

For a given input

$$\alpha = \big((a_1, g_1, b_1), \ldots, (a_n, g_n, b_n); (a, g, b)\big),$$

where $a_i, b_i \leq n$, $|g_i| \leq n$, $i = 1, \ldots, n$ and $a, b \leq n$, $|g| \leq n$, decide do there exist $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ such that

$$(a_{i_1}, g_{i_1}, b_{i_1})(a_{i_2}, g_{i_2}, b_{i_2}) \ldots (a_{i_k}, g_{i_k}, b_{i_k}) = (a, g, b).$$

The number n is the size of input α.

SSP over Brandt semigroups

For a given input

$$\alpha = \big((a_1, g_1, b_1), \ldots, (a_n, g_n, b_n); (a, g, b)\big),$$

where $a_i, b_i \leq n$, $|g_i| \leq n$, $i = 1, \ldots, n$ and $a, b \leq n$, $|g| \leq n$, decide do there exist $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ such that

$$(a_{i_1}, g_{i_1}, b_{i_1})(a_{i_2}, g_{i_2}, b_{i_2}) \ldots (a_{i_k}, g_{i_k}, b_{i_k}) = (a, g, b).$$

The number n is the size of input α.
It follows from the definition of multiplication in Brandt semigroup, that this equality holds if and only if hold the following both equalities:

$$(a_{i_1}, b_{i_1})(a_{i_2}, b_{i_2}) \ldots (a_{i_k}, b_{i_k}) = (a, b)$$

in semigroup $B(\mathbb{N})$ and

$$g_{i_1} g_{i_2} \cdots g_{i_n} = g$$

in the group G. If G is finitely defined group with polynomially decidable word problem, then the subset sum problem over $B(G, \mathbb{N})$ is in class NP.

SSP over Brandt semigroups

Lemma

If the subset sum problem over semigroup $B(G, \mathbb{N})$ is polynomially decidable, then the subset sum problem over group G is polynomially decidable.

SSP over Brandt semigroups

Lemma

If the subset sum problem over semigroup $B(G, \mathbb{N})$ is polynomially decidable, then the subset sum problem over group G is polynomially decidable.

$$(g_1, g_2, \ldots, g_n; g) \to \big((1, g_1, 1), (1, g_2, 1), \ldots, (1, g_n, 1); (1, g, 1)\big)$$

SSP over Brandt semigroups

Lemma

If the subset sum problem over semigroup $B(G, \mathbb{N})$ is polynomially decidable, then the subset sum problem over group G is polynomially decidable.

$$(g_1, g_2, \ldots, g_n; g) \to \big((1, g_1, 1), (1, g_2, 1), \ldots, (1, g_n, 1); (1, g, 1)\big)$$

Corrolary

SSP over $B(BS(1, 2), \mathbb{N})$ is NP-complete.

SSP over $B(\mathbb{N})$

Lemma

SSP over $B(\mathbb{N})$ is decidable in polynomial time.

SSP over $B(\mathbb{N})$

Lemma

SSP over $B(\mathbb{N})$ is decidable in polynomial time.

Lemma

Let S be the set of inputs α of SSP over $B(\mathbb{N})$ such that there are more than n^2 solutions for α, where $n = size(\alpha)$. Then S is negligible.

SSP over $B(\mathbb{N})$

Lemma

SSP over $B(\mathbb{N})$ is decidable in polynomial time.

Lemma

Let S be the set of inputs α of SSP over $B(\mathbb{N})$ such that there are more than n^2 solutions for α, where $n = size(\alpha)$. Then S is negligible.

Lemma

There is a generic polynomial algorithm $\mathcal{B}$, which for almost every input α of SSP over $B(\mathbb{N})$ outputs the list of all solutions.

SSP over Brandt semigroups

Theorem

Let G be a finitely defined group with word problem, decidable in polynomial time. Then SSP over Brandt semigroup $B(G, \mathbb{N})$ is generically decidable in polynomial time.

SSP over Brandt semigroups

Theorem

Let G be a finitely defined group with word problem, decidable in polynomial time. Then SSP over Brandt semigroup $B(G, \mathbb{N})$ is generically decidable in polynomial time.

A polynomial generic algorithm $\mathcal{A}$ works on input $\alpha = \big((a_1, g_1, b_1), \ldots, (a_n, g_n, b_n); (a, g, b)\big)$ in the following way.

SSP over Brandt semigroups

Theorem

Let G be a finitely defined group with word problem, decidable in polynomial time. Then SSP over Brandt semigroup $B(G, \mathbb{N})$ is generically decidable in polynomial time.

A polynomial generic algorithm $\mathcal{A}$ works on input $\alpha = \big((a_1, g_1, b_1), \ldots, (a_n, g_n, b_n); (a, g, b)\big)$ in the following way.

1. Find all solutions (if its number $\nu \leq n^2$) for the input

$$\alpha' = \big((a_1, b_1), \ldots, (a_n, b_n); (a, b)\big)$$

for SSP over $B(\mathbb{N})$.

SSP over Brandt semigroups

Theorem

Let G be a finitely defined group with word problem, decidable in polynomial time. Then SSP over Brandt semigroup $B(G, \mathbb{N})$ is generically decidable in polynomial time.

A polynomial generic algorithm $\mathcal{A}$ works on input $\alpha = \big((a_1, g_1, b_1), \ldots, (a_n, g_n, b_n); (a, g, b)\big)$ in the following way.

1. Find all solutions (if its number $\nu \leq n^2$) for the input

$$\alpha' = \big((a_1, b_1), \ldots, (a_n, b_n); (a, b)\big)$$

   for SSP over $B(\mathbb{N})$.
2. If $\nu > n^2$ then output «?».

SSP over Brandt semigroups

Theorem

Let G be a finitely defined group with word problem, decidable in polynomial time. Then SSP over Brandt semigroup $B(G, \mathbb{N})$ is generically decidable in polynomial time.

A polynomial generic algorithm $\mathcal{A}$ works on input $\alpha = \big((a_1, g_1, b_1), \ldots, (a_n, g_n, b_n); (a, g, b)\big)$ in the following way.

1. Find all solutions (if its number $\nu \leq n^2$) for the input

$$\alpha' = \big((a_1, b_1), \ldots, (a_n, b_n); (a, b)\big)$$

   for SSP over $B(\mathbb{N})$.
2. If $\nu > n^2$ then output «?».
3. If $\nu \leq n^2$, try all these solutions for SSP over group G.

Open problems

Is SSP generically decidable in polynomial time for:

1. Baumslag-Solitair group $BS(1,2)$?

Open problems

Is SSP generically decidable in polynomial time for:

1. Baumslag-Solitair group $BS(1,2)$?
2. Groups $SL(k,\mathbb{Z})$, $PSL(k,\mathbb{Z})$?

Open problems

Is SSP generically decidable in polynomial time for:

1. Baumslag-Solitair group $BS(1,2)$?
2. Groups $SL(k,\mathbb{Z})$, $PSL(k,\mathbb{Z})$?
3. Lamplighter group?

Open problems

Is SSP generically decidable in polynomial time for:

1. Baumslag-Solitair group $BS(1,2)$?
2. Groups $SL(k,\mathbb{Z})$, $PSL(k,\mathbb{Z})$?
3. Lamplighter group?

Is SSP decidable in polynomial time for finitely defined semigroups with small overlaps? (analog of Myasnikov-Nikolaev-Ushakov algorithm for hyperbolic groups)

The end

Thank you for your attention!