The word problem in Hanoi Towers groups

Ievgen Bondarenko

Communicated by V. I. Sushchansky

ABSTRACT. We prove that the elements of the Hanoi Towers groups $\mathcal{H}_m$ have depth bounded from above by a poly-logarithmic function $O(\log^{m-2} n)$, where n is the length of an element. Therefore the word problem in groups $\mathcal{H}_m$ is solvable in subexponential time $exp(O(\log^{m-2} n))$.

Introduction

We consider deterministic finite-state transducers (Mealy automata) with the same input and output alphabets. Such automata process words over the alphabet letter by letter: automaton reads the first letter from the current state, produces the output letter, and changes its state; the output letter and the new state depend only on the current state and the input letter. In this way, every state s taken as the initial state defines a transformation of words over the alphabet. If all transformations defined by the states of an automaton A are invertible, they generate a group under composition of functions, which is called the automaton group $\mathcal{G}_A$ generated by A. Further, we always assume that the states of automata determine a symmetric generating set of the group $\mathcal{G}_A$, so that every element of $\mathcal{G}_A$ can be given by a word $s_1 s_2 \ldots s_n$ over states.

The word problem in every automaton group is solvable. To describe the algorithm, define the section $s|_v$ of a state s at a word v over the

---

2010 MSC: 68R05, 20F10.
Key words and phrases: the Tower of Hanoi game, automaton group, word problem, complexity.

alphabet as the end state of the automaton after processing the word v from the initial state s. This notion naturally extends to words over the states of an automaton: the section of $w = s_1 s_2 \ldots s_n$ at a word v is defined by

$$(s_1 s_2 \ldots s_n)|_v = s_1' s_2' \ldots s_n', \text{ where } s_i' = s_i|_{(s_{i+1} \ldots s_n)(v)} \qquad (1)$$

(we are using left actions). The section $(s_1 s_2 \ldots s_n)|_v$ has a natural interpretation: take n copies of the automaton A, choose the initial state s_i in the i-th copy, and connect the output of $(i+1)$-th automaton to the input of i-th automaton. Then the final configuration of states after processing a word v is the section $(s_1 s_2 \ldots s_n)|_v$. Note that we treat sections as words over states, together with the natural action on words over the alphabet.

Now the algorithm solving the word problem in automaton groups follows from the fact: a word w over the states of an automaton A defines the trivial transformation if and only if the sections of w at all words over the alphabet act trivially on the alphabet. If the automaton A has k states, then $s_1 s_2 \ldots s_n$ has at most k^n sections. Therefore this algorithm solves the word problem in at most exponential time. The precise complexity of the word problem in the class of automaton groups is unknown. For contracting automaton groups the word problem is solvable in polynomial time [8, Proposition 2.13.10]. The word problem in groups generated by polynomial automata is solvable in subexponential time [1]. And of course, if the automaton group is free, nilpotent, etc., then the complexity of the word problem is smaller than exponential as well.

The complexity of the above algorithm directly depends on the number of sections that elements have. Define the *section growth function* of an automaton A:

$$\theta_{\mathsf{A}}(n) = \max\{\#Sections(s_1 s_2 \ldots s_k) : s_i \in \mathsf{A}, k \leq n\}, \ n = 1, 2, \ldots.$$

If the function θ_{A} is bounded from above by a polynomial (subexponential) function then the word problem in the group $\mathcal{G}_{\mathsf{A}}$ is solvable in polynomial (subexponential) time.

The number of sections can be bounded by the depth of state words. Let $d_{\mathsf{A}}(s_1 s_2 \ldots s_n)$ be the least integer d with the property that for every word v over the alphabet there exists a word u of length $\leq d$ such that $(s_1 s_2 \ldots s_n)|_v = (s_1 s_2 \ldots s_n)|_u$. Define the *depth function* of an automaton A:

$$d_{\mathsf{A}}(n) = \max\{d_{\mathsf{A}}(s_1 s_2 \ldots s_k) : s_i \in \mathsf{A}, k \leq n\}, \ n = 1, 2, \ldots.$$

If the alphabet has a letters, then $\theta_{\mathsf{A}}(n) \leq 1 + a^1 + \ldots + a^{d_{\mathsf{A}}(n)}$. Therefore, if the depth function is bounded from above by a poly-logarithmic function, then the word problem is solvable in subexponential time.

Instead of computing sections as words over states, we can compute a few relations in a given automaton group, and then reduce sections using these relations. This may highly reduce the number of sections and, as a consequence, the complexity of the algorithm. For example, in contracting automaton groups, computing certain relations in a finite time will guarantee the depth function $d(n) = O(\log n)$ and polynomial word problem.

The goal of this note is to estimate the depth function of the Hanoi automata H_m and the complexity of the word problem in the Hanoi Towers groups $\mathcal{H}_m = \mathcal{G}_{\mathsf{H}_m}$, which model the Tower of Hanoi game on m pegs [3]. This classical game is played with k disks of distinct size placed on m pegs, $m \geq 3$. Initially, all disks are placed on the first peg according to their size so that the smallest disk is at the top, and the largest disk is at the bottom. A player can move only one top disk at a time from one peg to another peg, and can never place a bigger disk over a smaller disk. The goal of the game is to transfer the disks from the first peg to another peg. For more information about this game, its history, solutions, and open problems, we refer the reader to [5, 6, 9] and the references therein. The automaton model H_m presented in [3] encodes configurations of disks on pegs by words over the alphabet $\{1, 2, \ldots, m\}$, and the action of each state of the automaton H_m corresponds to a single disk move between two pegs. Therefore each strategy of the player can be encoded by a word over states of the automaton H_m. The Hanoi Towers game has subexponential complexity for $m \geq 4$: it can be solved in $n = exp(O(k^{\frac{1}{m-2}}))$ moves, and this is an asymptotically optimal solution (see more precise estimates in [6, 9]). Note that if we express the height of the tower k in terms of the length n of an optimal solution, we get $k = O(\log^{m-2} n)$. I do not see how the estimate on the game's complexity immediately implies the estimate on the complexity of the word problem in the groups $\mathcal{H}_m$. Nevertheless, we prove the following results.

Theorem 1. *The depth function of the Hanoi automaton* H_m *satisfies* $d(n) = O(\log^{m-2} n)$.

Corollary 1. *The section growth function of the Hanoi automaton* H_m *satisfies* $\theta(n) = exp(O(\log^{m-2} n))$.

Corollary 2. *The word problem in the Hanoi Towers group $\mathcal{H}_m$ is solvable in subexponential time $exp(O(\log^{m-2} n))$.*

1. Automaton groups and Hanoi automata

In this section we briefly review necessary information about automata, automaton groups, and describe the construction of Hanoi automata H_m. See [2, 4, 8] for more details.

Let X be a finite alphabet and X^* be the free monoid over X. The elements of X^* are finite words $x_1 x_2 \ldots x_n$, $x_i \in X$, $n \in \mathbb{N} \cup \{0\}$, the identity element is the empty word $\varnothing$, and the operation is concatenation of words. The length of $v = x_1 x_2 \ldots x_n$ is $|v| = n$.

An automaton A over the alphabet X is a finite directed labeled graph, whose vertices are called the states of the automaton, and for each vertex $s \in \mathsf{A}$ and every letter $x \in X$ there exists a unique outgoing arrow at s labeled by $x|y$ for some $y \in X$.

Every state $s \in \mathsf{A}$ defines a transformation of X^* as follows. Given a word $v = x_1 x_2 \ldots x_n \in X^*$, there exists a unique directed path in the automaton A starting at the state s and labeled by $x_1|y_1$, $x_2|y_2, \ldots, x_n|y_n$ for some $y_i \in X$. Then the word $y_1 y_2 \ldots y_n$ is called the *image of $x_1 x_2 \ldots x_n$ under s*, and the end vertex of this path is called the *section of s at v* denoted $s|_v$. A word $s_1 s_2 \ldots s_n$ over states acts on X^* by composition: $(s_1 s_2 \ldots s_n)(v) = (s_1 s_2 \ldots s_{n-1})(s_n(v))$. The section of a word $s_1 s_2 \ldots s_n$ over states at a word $v \in X^*$ is defined by Equation (1). Further, by *states in section $(s_1 s_2 \ldots s_n)|_v$* we mean states s_i' given by Equation (1).

If all transformations defined by the states of A are invertible, the automaton A is called invertible, and the group $\mathcal{G}_\mathsf{A}$ generated by these transformations under composition of functions is called the automaton group generated by A.

The *Hanoi automaton H_m* is defined over $X_m = \{1, 2, \ldots, m\}$. It has the trivial state e and the state $a_{(ij)}$ for every transposition (i, j) on X. All arrows outgoing from e end in e and are labeled by $x|x$ for each $x \in X$. Each state $a_{(ij)}$ has two outgoing arrows $a_{(ij)} \to e$ labeled by $i|j$ and $j|i$, and the other arrows are loops at $a_{(ij)}$ labeled by $x|x$ for every $x \in X \setminus \{i, j\}$. For example, the automaton H_4 is shown in Figure 1 (the loops at the trivial state e are not drawn).

The automaton H_m is invertible and produces a symmetric generating set of the Hanoi Towers group $\mathcal{H}_m = \mathcal{G}_{\mathsf{H}_m}$. The action of states $a_{(ij)}$ on

the space X_m^* can be given recursively as follows:

$$a_{(ij)}(iv) = jv, \quad a_{(ij)}(jv) = iv, \quad a_{(ij)}(xv) = xa_{(ij)}(v) \text{ for } x \notin \{i, j\}.$$

In other words, the state $a_{(ij)}$ changes the first occurrence of letter i or j to the other one, and leaves the other letters unchanged. The states of the Hanoi automata satisfy the following important property: for any $x \in X_m$ and $s \in \mathsf{H}_m$,

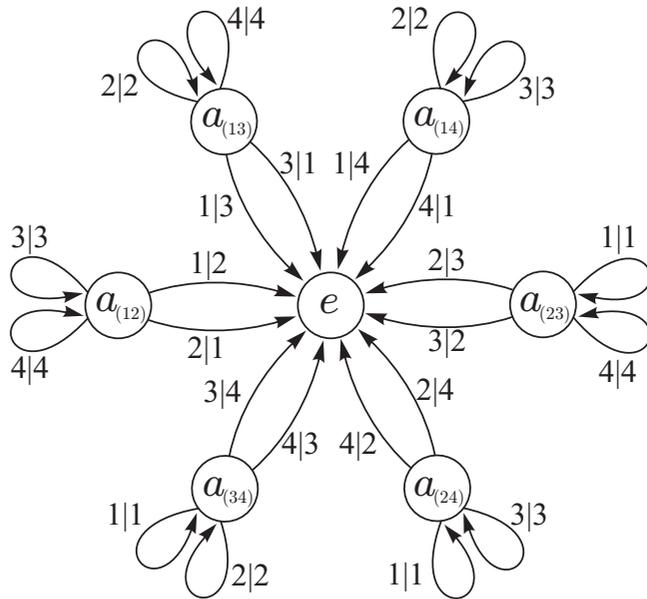$$\text{if } s(x) \neq x \text{ then } s|_x = e, \quad \text{if } s(x) = x \text{ then } s|_x = s. \tag{2}$$

FIGURE 1. The Hanoi automaton H_4

2.   Proof of Theorem 1

Proof. Let d_m be the depth function of the Hanoi automaton H_m. We will prove

$$d_m(n) \leq m^2(m-1)^2 \ldots 3^2(\log n)^{m-2} + d_{m-1}(n), \tag{3}$$
$$d_3(n) \leq \log n + 1.$$

(For simplicity, to avoid extra brackets, by $\log n$ we mean the least integer greater than the binary logarithm of n).

Case $m = 3$. Let us estimate the depth of certain words over states. If $w = ee \ldots e$ then $d_3(w) = 0$. We say that a word $w = s_1 s_2 \ldots s_n$ is a one-letter word, if there exists $s \in \mathsf{H}_3$ such that $s_i \in \{e, s\}$ for all i. Note that $d_3(w) \leq 1$ for every one-letter word w. If w_1, w_2 are one-letter words (maybe for different letters s), then the properties (2) imply that the section $(w_1 w_2)|_x$ is a one-letter word for every $x \in X_3$, and $d_3(w_1 w_2) \leq 2$. It follows that if a word w is a concatenation of n one-letter words, then $w|_x$ is a concatenation of at most $(n+1)/2$ one-letter words. The estimate $d_3(n) \leq \log n + 1$ follows.

Case $m > 3$. Fix $x \in X_m$. Let $\mathsf{H}_m^{(x)}$ be the set of all states $s \in \mathsf{H}_m$ that fix x. Note that $\mathsf{H}_m^{(x)}$ is a subautomaton of H_m. In particular, for any $s_i \in \mathsf{H}_m^{(x)}$ and $v \in X_m^*$ every state in the section $(s_1 s_2 \ldots s_n)|_v$ fixes x. Application of inductive hypothesis is based on the following observation: If we restrict the automaton $\mathsf{H}_m^{(x)}$ to the alphabet $X_m \setminus \{x\}$, we get the automaton H_{m-1}. This gives an estimate on the number of sections at words over $X_m \setminus \{x\}$. But since $s(x) = x$ and $s|_x = s$ for each $s \in \mathsf{H}_m^{(x)}$, we have

$$(s_1 s_2 \ldots s_n)|_v = (s_1 s_2 \ldots s_n)|_u$$

for every $v \in X_m^*$ and $s_i \in \mathsf{H}_m^{(x)}$, where u is the word over $X_m \setminus \{x\}$ that is obtained from v by removing every occurrence of letter x. Therefore it is sufficient to count only sections at words over $X_m \setminus \{x\}$. We have proved the estimate

$$d_m(s_1 s_2 \ldots s_n) \leq d_{m-1}(n) \tag{4}$$

for all $s_i \in \mathsf{H}_m^{(x)}$, $n \in \mathbb{N}$, $m > 3$.

Now estimate (3) follows once we prove the following statement.

We say that a *section* $(s_1 s_2 \ldots s_n)|_v$ *satisfies the property (*)* if there exists $x \in X_m$ such that every state in $(s_1 s_2 \ldots s_n)|_v$ fixes x, i.e., belongs to $\mathsf{H}_m^{(x)}$.

Claim. *For every* $n \in \mathbb{N}$ *and any states* $s_1, \ldots, s_n \in \mathsf{H}_m$, *the section* $(s_1 s_2 \ldots s_n)|_v$ *satisfies the property (*) for all words* $v \in X_m^*$ *of length* $|v| \geq C_m (\log n)^{m-2}$ *with constant* $C_m = 3^2 4^2 \ldots m^2$. *In particular,* $(s_1 s_2 \ldots s_n)|_v$ *is a word over the states of* $\mathsf{H}_m^{(x)}$ *for some* $x \in X$, *and* $d_m((s_1 s_2 \ldots s_n)|_v) \leq d_{m-1}(n)$.

Proof. We prove the claim by induction on m. The basis of induction $m = 3$ is shown above. Suppose the claim holds for less than m and consider the case m. We are going to use the inductive hypothesis as

follows: for every $x \in X_m$, any $s_i \in \mathsf{H}_m^{(x)}$, and all words $v \in X_m^*$ that contain at least $C_{m-1}(\log n)^{m-3}$ letters different from x, there exists a letter $y \in X_m$, $y \neq x$, such that every state in the section $(s_1 s_2 \ldots s_n)|_v$ fixes y.

Take two different letters $x, y \in X_m$ and elements $g = s_1 s_2 \ldots s_k$ for $s_i \in \mathsf{H}_m^{(x)}$ and $h = t_1 t_2 \ldots t_l$ for $t_i \in \mathsf{H}_m^{(y)}$. Consider the section $(gh)|_v$ for a word $v \in X^*$ of length $\geq |X_m| C_{m-1}(\log n)^{m-3}$, $n = k + l$. Then v contains at least $C_{m-1}(\log n)^{m-3}$ letters z for certain $z \in X_m$. If $z \neq y$ then we can apply inductive hypothesis to $h|_v$: every state in $h|_v$ fixes some letter besides y. If $z = y$ then $h(v)$ contains at least $C_{m-1}(\log n)^{m-3}$ letters $z = y \neq x$, and every state in $g|_{h(v)}$ fixes some letter besides x. If we get a common letter for $g|_{h(v)}$ and $h|_v$, then $(gh)|_v$ satisfies the property (*). Otherwise, we get at least three letters, each one fixed either by all states in $h|_v$, or in $g|_{h(v)}$. We can proceed further and consider $(gh)|_v|_u$ for words u of length $\geq |X_m| C_{m-1}(\log n)^{m-3}$. Then either $(gh)|_v|_u$ satisfies the property (*), or there are at least four letters, each one fixed either by all states in $h|_v$, or in $g|_{h(v)}$. It follows that for all words v of length $\geq |X_m|^2 C_{m-1}(\log n)^{m-3}$ the section $(gh)|_v$ satisfies the property (*).

Now consider any word $w = s_1 s_2 \ldots s_n$, $s_i \in \mathsf{H}_m$. We partition w on subwords

$$w = w_1 w_2 \ldots w_k, \quad k \leq n \tag{5}$$

such that every subword $w_i = s_{j_i} s_{j_i+1} \ldots s_{j_{i+1}-1}$ satisfies the property (*). Consider the words $w_1 w_2$, $w_3 w_4$, ..., and their sections at words v of length $\geq |X_m|^2 C_{m-1}(\log n)^{m-3}$. Then, using the fact proved in the previous paragraph, the section $w|_v$ can be represented in the form (5) with $\leq (k+1)/2$ subwords. Applying this procedure $\log k \leq \log n$ times, we get that $g|_v$ satisfies the property (*) for all words v of length $\geq |X_m|^2 C_{m-1}(\log n)^{m-2}$. The claim is proved.   □

□

Using program package [7] we have calculated the values of the depth function $d_4(n)$ and the section growth function $\theta_4(n)$ of the Hanoi automaton H_4 for small values of n:

n	1	2	3	4	5	6	7	8	9	10	11	12
$d_4(n)$	1	2	2	3	4	4	5	5	6	6	6	7
$\theta_4(n)$	2	4	8	13	17	24	31	39	48	60	70	81

This suggests that $\theta_4(n)$ may have polynomial (quadratic?) growth, which would imply polynomial word problem.

References

[1] I. Bondarenko, *Growth of Schreier graphs of automaton groups,* Mathematische Annalen, **354** (2012), no. 2, 765–785.

[2] R.I. Grigorchuk, V.V. Nekrashevych, V.I. Sushchansky, *Automata, dynamical systems and groups*, Proceedings of the Steklov Institute of Mathematics, **231** (2000), 128–203.

[3] R.I. Grigorchuk, Z. Šuniḱ, *Asymptotic aspects of Schreier graphs and Hanoi Towers groups*, C. R. Math. Acad. Sci. Paris, **342** (2006), no. 8, 545–550.

[4] R.I. Grigorchuk, Z. Šuniḱ, *Self-similarity and branching in group theory*, London Mathematical Society Lecture Note Series, **339** (2007), 36–95.

[5] A.M. Hinz, *The Tower of Hanoi*, Enseign. Math. (2) **35** (1989), no. 3-4, 289–321.

[6] S. Klavžar, U. Milutinovic, C. Petr, *On the Frame-Stewart algorithm for the multi-peg Tower of Hanoi problem*, Discrete Appl. Math. **120** (2002), no. 1-3, 141–157.

[7] Y. Muntyan, D. Savchuk, *AutomGrp – GAP package for computations in self-similar groups and semigroups, Version 1.1.2*, 2008.

[8] V. Nekrashevych, *Self-similar groups*, Mathematical Surveys and Monographs **117**, AMS, Providence, RI, 2005.

[9] M. Szegedy, *In how many steps the k peg version of the Towers of Hanoi game can be solved?* In: STACS 99 (Trier), Lecture Notes in Comput. Sci. **1563**, Springer, Berlin, (1999), 356–361.

CONTACT INFORMATION

I. Bondarenko   Department of Algebra and Mathematical Logic, Mechanics and Mathematics Faculty, National Taras Shevchenko University of Kyiv, vul.Volodymyrska 64, 01033, Kyiv, Ukraine
E-Mail: ibond.univ@gmail.com