

## О СЛОЖНОСТИ ВЫЧИСЛЕНИЙ В ГРУППАХ

А. Ю. ОЛЬШАНСКИЙ

Московский государственный университет им. М.В. Ломоносова

THE COMPUTATION COMPLEXITY  
IN GROUPS

A. Yu. OLSHANSKII

*Any group  $G$  can be described in terms of generators and relations. The key question is whether given two words in generators represent the same element of  $G$  or not. Examples, the setting of the algorithmic problem and formulations of some recent results on the complexity of the word problem for groups are given.*

*Любая группа  $G$  может быть задана с помощью порождающих элементов и соотношений между ними. При этом основным оказывается вопрос о существовании алгоритма для распознавания, представляют ли два слова от порождающих один и тот же элемент в  $G$  или нет. В статье приведены примеры, сформулирована проблема и описаны недавние результаты исследований сложности проблемы слов для групп.*

Группы появляются в математике вместе с симметриями и преобразованиями. Для знакомства с примерами и основными понятиями рекомендуем опубликованные в “Соросовском Образовательном Журнале” статьи [1, 2]. Напомним только, что всякая группа  $G$  является множеством, наделенным ассоциативной операцией  $a \cdot b$  для  $a, b \in G$  (элемент  $c = a \cdot b$  группы  $G$  называют обычно произведением элементов  $a$  и  $b$ ), причем выполнены аксиомы единицы (существует элемент  $e \in G$ , такой, что  $ae = ea = a$  для любого  $a \in G$ ) и обратного (для любого  $a \in G$  существует элемент  $b = a^{-1} \in G$ , такой, что  $ab = ba = e$ ). Из аксиом непосредственно следует единственность единицы и единственность обратного для всякого  $a \in G$ . Некоторые другие особенности групповых исчислений обсуждаются в [2].

Для эффективного вычисления произведений в группе  $G$  должен быть какой-то единообразный способ описания ее элементов. Во многих группах (так называемых группах Ли) эта задача решается с помощью задания локальной (а иногда глобальной) системы координат. Характерным примером является группа всех движений трехмерного евклидова пространства, где каждое движение задается шестью параметрами в соответствии с нашим представлением о шести степенях свободы твердого тела в пространстве.

Но есть и другие группы, устроенные дискретно в том смысле, что в малой окрестности элемента группы других элементов вообще нет (см. примеры ниже). В статье речь пойдет об универсальном способе задания элементов любой группы в виде слов от порождающих. Мы затронем основной вопрос, возникающий в исчислении слов, – проблему распознавания равенства слов в группе.

## ПОРОЖДАЮЩИЕ

Начнем с примеров.

1. Вообразим неограниченный лист клетчатой бумаги, клетки которого – единичные квадраты и который покрывает всю евклидову плоскость. Пусть  $G_1$  – группа всех параллельных переносов плоскости вдоль себя (или сдвигов), сохраняющих данную клетчатую

решетку: узлы решетки — ее точки с целыми координатами — должны перемещаться опять-таки в узлы. Операция умножения в  $G_1$  — это суперпозиция двух переносов, то есть их последовательное выполнение.

Понятно, что в  $G_1$  входят любые сдвиги плоскости на векторы  $(m, n)$  с целыми  $m$  и  $n$ . Любой такой перенос можно получить как результат многократного умножения (то есть повторного применения) всего из двух переносов и им обратных: из сдвига  $a$  на вектор  $(1; 0)$  и сдвига  $b$  на вектор  $(0; 1)$ . Например, перенос  $t_{2,-3}$  на вектор  $(2; -3)$  можно записать как  $aab^{-1}b^{-1}b^{-1}$  или  $a^2b^{-3}$ , то есть, выполняя последовательно три раза перенос  $b^{-1}$  на вектор  $(0; -1)$ , а затем два раза перенос  $a$ , мы в итоге имеем перенос  $t_{2,-3}$ .

Говорят, что сдвиги  $a$  и  $b$  составляют систему порождающих группы  $G_1$ . (Систему порождающих можно выбрать многими способами. Проверьте, что ее можно составить, в частности, из сдвигов на векторы  $(2; 1)$  и  $(3; 2)$ .)

2. Напомним, что числовые матрицы  $\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$

размера  $2 \times 2$  умножаются по правилу

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} \alpha' & \beta' \\ \gamma' & \delta' \end{pmatrix} = \begin{pmatrix} \alpha\alpha' + \beta\gamma' & \alpha\beta' + \beta\delta' \\ \gamma\alpha' + \delta\gamma' & \gamma\beta' + \delta\delta' \end{pmatrix}.$$

Обозначим через  $G_2$  подмножество всех матриц вида  $A = \begin{pmatrix} 2^i & r \\ 0 & 1 \end{pmatrix}$ , где  $i$  — любое целое число, а  $r$  — двоично-рациональная дробь (ее числитель целый, а знаменатель — степень двойки).

Предлагаем читателю проверить, что  $G_2$  тоже группа: произведение двух матриц из  $G_2$  содержится в этом же подмножестве, умножение ассоциативно, единичной является матрица  $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ , а обратной для  $A$  будет

матрица  $\begin{pmatrix} 2^{-i} & -2^{-i}r \\ 0 & 1 \end{pmatrix}$ . Докажите также, что матрицы

$a = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$  и  $b = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$  порождают группу  $G_2$ , то

есть с помощью многократного умножения из матриц  $a$ ,  $a^{-1}$ ,  $b$  и  $b^{-1}$  можно получить всякую матрицу из  $G_2$ .

3. Пусть  $G_3$  — группа всех перестановок на множестве  $\{1, 2, 3\}$  (см. [1]). Проверьте, что перестановки

$a = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$  и  $b = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$  порождают груп-

пу  $G_3$ . Например,  $\begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = aba$ , где произведе-

ние  $aba$  есть результат выполнения перестановок  $a$ ,  $b$  и опять  $a$ .

Далее рассматриваются только конечно-порожденные группы: это группы с конечной системой порождающих. Из примеров понятно, что под системой порождающих  $(a_1, \dots, a_m)$  группы  $G$  понимается такое подмножество ее элементов, что каждый элемент  $g \in G$  равен некоторому произведению  $g_1 \dots g_n$ , где любой сомножитель — это один из элементов  $a_1, \dots, a_m, a_1^{-1}, \dots, a_m^{-1}$ . При этом говорят также, что каждый элемент из  $G$  представляется некоторым словом от порождающих  $a_1, \dots, a_m$ . Например, слово  $a_1a_2a_1^{-1}a_1^{-1}a_2a_2a_2 = a_1a_2a_1^{-2}a_2^3$  имеет длину 7, а пустое слово (не содержащее ни одной буквы; обозначим его здесь  $w_\emptyset$ ) — длину 0. Считаем, что  $w_\emptyset$  представляет единицу группы  $G$ .

## СООТНОШЕНИЯ И ИХ СЛЕДСТВИЯ

В рассмотренных примерах запись элемента группы в виде слова от порождающих неоднозначна. К примеру, в группе  $G_1$  (здесь  $(a_1, a_2) = (a, b)$ ) имеем  $a^2b = aba = a^{-1}b^2a^3b^{-1}$ , а в  $G_2$   $aba^{-1} = b^2$  (проверьте!) и т.д. Поскольку в любой группе  $G$  равенство  $u = v$  равносильно равенству  $uv^{-1} = e$ , говоря о соотношениях группы, обычно имеют в виду равенства вида  $w = w_\emptyset$ , где  $w$  — слово от порождающих, представляющее единичный элемент группы. Соотношение записывают также в форме  $w = e$ , и, допуская вольность речи, слово  $w$  тоже называют соотношением. Подчеркнем, что в этом контексте  $w$  рассматривается формально, то есть именно как слово (конечная последовательность букв), так как при содержательном толковании (как произведения в группе  $G$ ) левая часть равенства  $w = w_\emptyset$  неотличима от правой. Для графического же (то есть побуквенного) равенства слов будем употреблять символ  $\equiv$ . Обратным для слова  $w \equiv b_1 \dots b_n$  (где  $b_1, \dots, b_n$  — любые буквы из множества  $a_1, a_1^{-1}, \dots, a_m, a_m^{-1}$ ) считается слово  $w^{-1} \equiv b_n^{-1} \dots b_1^{-1}$ . Ясно, что если  $w$  — соотношение группы  $G$ , то и  $w^{-1}$  также соотношение, так как  $ww^{-1} = w^{-1}w = e$  в  $G$  для любого слова  $w$ .

Множество соотношений произвольной группы  $G$  бесконечно, но из него в наиболее важных случаях удается выделить конечное подмножество так называемых определяющих соотношений, из которых следуют все остальные. Нужно лишь уточнить правила вывода следствий.

Для каждого множества слов  $R$  в алфавите  $A^{\pm 1} = \{a_1^{\pm 1}, \dots, a_m^{\pm 1}\}$  введем следующие виды  $R$ -преобразований, применяемых далее к произвольному слову  $w$  в том же алфавите. Их определение основано на простом

наблюдении, что в любой группе  $xu = xeu$  и  $xzz^{-1}u = xu$  для любых элементов  $x, y, z$ .

I. Если  $w \equiv uaa^{-1}v$  (где  $a$  – буква из  $A^{\pm 1}$ , а  $u, v$  – какие-то слова, возможно пустые), то можно сделать сокращение:  $uaa^{-1}v \longrightarrow uv$ .

I'. Вставка (обратное преобразование для I):  $uv \longrightarrow uaa^{-1}v$ .

II. Если  $w = urv$  для некоторых слов  $u, v$ , где  $r \in R$  или  $r^{-1} \in R$ , то можно сделать вычеркивание:  $urv \longrightarrow uv$ .

II'. Обратное преобразование для II:  $uv \longrightarrow urv$ .

Некоторое множество слов-соотношений  $R$  группы  $G$  (в алфавите  $A^{\pm 1}$ ) называется множеством определяющих слов (или соотношений), если всякое слово-соотношение  $w$  может быть приведено к пустому слову с помощью нескольких последовательно выполненных  $R$ -преобразований типов I–II'. Например, в группе сдвигов  $G_1$  слово  $w$  в алфавите  $\{a, a^{-1}, b, b^{-1}\}$  тогда и только тогда задает тождественное преобразование  $e$  (сдвиг на нулевой вектор), когда буква  $a$  встречается в  $w$  столько же раз, сколько и буква  $a^{-1}$ , а  $b$  – столько раз, сколько  $b^{-1}$ . Тогда оно может быть приведено к пустому слову посредством нескольких перестановок букв  $a^{\pm 1}$  с буквами  $b^{\pm 1}$  и последующих сокращений. Такая перестановка букв получается с помощью цепочки  $R$ -преобразований для множества  $R$ , состоящего из одного слова  $r \equiv a^{-1}b^{-1}ab$ . Для примера:  $ba \xrightarrow{\text{II}'} baa^{-1}b^{-1}ab \xrightarrow{\text{I}'} bb^{-1}ab \xrightarrow{\text{I}'} ab$ .

Следовательно, в качестве единственного определяющего слова для группы  $G_1$  можно взять  $a^{-1}b^{-1}ab$ .

В качестве более трудной задачи мы оставляем читателю проверку того факта, что в качестве одного определяющего слова для группы матриц  $G_2$  можно взять  $aba^{-1}b^2$  (соотношение  $aba^{-1}b^2 = e$  чаще записывается в равносильной форме  $aba^{-1} = b^2$ ). Докажите также, что множество  $\{a^2, b^2, (ab)^3\}$  будет множеством определяющих соотношений для группы  $G_3$  (здесь  $(ab)^3 \equiv ababab$ ).

Вообще запись вида

$$G_4 = \langle a, b, c, d \mid aba^{-1}b^{-1}cdc^{-1}d^{-1} \rangle$$

означает, что группа  $G_4$  порождается элементами  $a, b, c, d$ , а определяющим словом является слово  $aba^{-1}b^{-1}cdc^{-1}d^{-1}$ . На вопрос, существует ли такая группа, ответ положительный. В качестве ее элементов можно взять классы  $R$ -эквивалентных слов (то есть в  $G_4$  два слова равны тогда и только тогда, когда одно из другого можно получить с помощью нескольких  $R$ -преобразований для  $R = \{aba^{-1}b^{-1}cdc^{-1}d^{-1}\}$ ). Произведение же  $uv$  слов  $u$  и  $v$  получается (с точностью до  $R$ -эквивалентности) посредством простого приписывания слова  $u$  к слову  $v$ .

С одной стороны, “синтаксический” подход такого рода дает много новых примеров групп. С другой – описание групп с помощью порождающих и соотноше-

ний естественно для многих вопросов алгебры, геометрии, топологии и математической логики.

## ПРОБЛЕМА РАВЕНСТВА СЛОВ

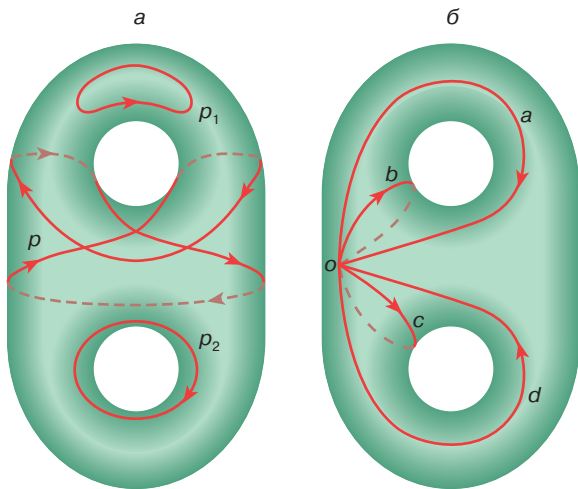
Итак, каждый элемент группы  $G$  задается некоторым словом от порождающих  $\{a_1^{\pm 1}, \dots, a_m^{\pm 1}\}$ . При таком задании основным оказывается вопрос об идентификации элементов: как по любым двум словам  $u, v$  узнать, представляют ли они один и тот же элемент, то есть верно ли, что  $w = e$  в  $G$  для частного  $w = uv^{-1}$ ?

Для группы  $G_1$  проблема равенства слов (короче, проблема слов) решается очень просто, так как каждое слово легко приводится  $R$ -преобразованиями к единственному каноническому виду  $a^s b^t$  с целыми показателями  $s$  и  $t$ . В случае группы матриц  $G_2$  для решения проблемы равенства двух слов  $u, v$  в алфавите  $\{a^{\pm 1}, b^{\pm 1}\}$  нужно просто вычислить задаваемые этими словами произведения матриц, а в случае группы  $G_3$  – соответствующие произведения перестановок. Для группы  $G_4$  (и других фундаментальных групп поверхностей) М. Дэн в начале века обосновал алгоритм распознавания равенства слов  $w$  единице, основанный на том, что для такого слова  $w$  существует цепь  $R$ -преобразований  $w \longrightarrow \dots \longrightarrow w_\phi$ , где каждое последующее слово короче предыдущего. При этом сама группа  $G_4$  интересна как фундаментальная группа поверхности, а именно сферы с двумя ручками  $S$  (рис. 1, а).

Опишем общее правило построения фундаментальной группы  $G$  поверхности  $S$ . Два замкнутых пути (петли)  $p$  и  $p'$ , начинающиеся и кончающиеся в фиксированной на  $S$  точке  $o$ , называются гомотопными, если один в другой можно превратить с помощью непрерывной деформации, сохраняющей точку  $o$ , по поверхности  $S$ . (Например, на сфере все такие петли гомотопны тривиальному пути, состоящему только из точки  $o$ , то есть они стягиваемы по сфере в точку.)

Элементом группы  $G$  считается класс  $[p]$  всех петель, гомотопных некоторой петле  $p$ . Умножение задается равенством  $[p][q] = [pq]$ , где  $pq$  – петля, для прохождения которой сначала проходит  $p$ , а потом  $q$ . Можно проверить, что введенная операция удовлетворяет аксиомам группы. В частности, роль единицы играет класс тривиальной петли, а класс, обратный к  $[p]$ , получается изменением обхода петель на противоположное. Фундаментальная группа показывает, насколько сложна структура петель на поверхности. Так, в случае сферы группа  $G$  состоит лишь из единичного элемента, а для поверхности тора фундаментальной группой оказывается группа из примера 1.

Вопрос о том, какие замкнутые кривые  $p$  на  $S$  могут быть непрерывно стянуты в точку по данной поверхности, является естественным и важным в топологии.



**Рис. 1**

Очевидно, что кривая  $p_1$  стягиваема по  $S$  в точку, а кривая  $p_2$  нет (рис. 1, а). При естественном задании кривой  $p$  вопрос о ее стягиваемости в точку сводится к проблеме равенства единице слова в группе  $G_4$ . Для получения этого слова путь, проходящий через фиксированную точку  $o$ , можно предварительно непрерывно деформировать по  $S$  в произведение путей  $a, b, c, d$  и им обратных, изображенных на рис. 1, б.

Возвращаясь к проблеме слов для произвольной группы  $G$ , отметим, что она может быть и значительно сложнее, чем в указанных примерах. Более того, знаменитая теорема Новикова–Буна, доказанная в первой половине 50-х годов, утверждает, что существует конечно-определенная группа (то есть группа, заданная конечным числом порождающих и определяющих соотношений), для которой проблема слов алгоритмически неразрешима в принципе.

Определения алгоритма не требуется для положительного решения проблемы слов в группах  $G_1–G_4$ . (Алгоритмы вошли в науку и жизнь задолго до формализации этого понятия в первой половине XX века.) Наоборот, факт отсутствия какого-либо алгоритма, решающего ту или иную “массовую” проблему, может быть доказан только на основе точного определения этого понятия. Все разумные определения алгоритма оказываются равносильными, а наиболее известную формализацию – с помощью понятия машины Тьюринга – можно найти в [3].

Детерминированную машину Тьюринга, решающую проблему слов в группе  $G$ , в общих чертах можно представлять себе в виде компьютерной программы, которая по любому слову  $w$  подаваемому на вход (то есть определенным образом записываемому в конеч-

ную, но в принципе неограниченную память машины), на выходе дает ответ, равно слово  $w$  единице в  $G$  или нет. При этом время работы машины, реализующей алгоритм, – это число команд, выполненных компьютером. Каждая команда в момент времени  $n$  (где  $n = 0, 1, 2, \dots$ ) в однозначной зависимости от буквы, записанной в обозреваемой в этот момент ячейке памяти, и от внутреннего состояния машины (число которых ограничено) выполняет одно из следующих предписаний. Либо она изменяет букву в данной ячейке, либо предписывает к моменту  $n + 1$  перейти к одной из двух соседних ячеек памяти (условно расположенных на ленте), либо справа или слева от просматриваемой головки к ленте добавляется новая ячейка с определенной буквой. Одновременно команда указывает на изменение состояния машины к моменту  $n + 1$ . В момент времени  $n = 0$  машина находится в начальном состоянии, а завершается работа при переходе машины в заключительное состояние.

При всей своей значительности теорема Новикова–Буна не должна обескураживать, так как для естественно возникающих групп проблема слов, к счастью, оказывается алгоритмически разрешимой. В этих случаях в соответствии с современными тенденциями в развитии вычислительной математики и компьютерной техники все большее значение приобретает проблема асимптотической сложности алгоритмов.

### СЛОЖНОСТЬ ПРОБЛЕМЫ СЛОВ

Важнейшей характеристикой сложности алгоритма является длительность вычислительного процесса. Временная функция  $T(n)$  для алгоритма или реализующей его машины Тьюринга  $M$ , решающей проблему слов в группе  $G$  с порождающими  $a_1, \dots, a_m$ , определяется следующим образом. Пусть для каждого слова  $w$  в алфавите  $A^{\pm 1} = \{a_1^{\pm 1}, \dots, a_m^{\pm 1}\}$  за время  $T_w$  (= числу команд) машина  $M$  узнает, равно слово  $w$  единице в  $G$  или нет. Тогда  $T(n)$  – это максимум времен  $T_w$  для всех слов  $w$  длины, не превосходящей  $n$ . Таким образом, функция  $T(n)$  оценивает сверху продолжительность работы алгоритма в зависимости от длины исследуемого слова  $w$ .

Говорят, что сложность алгоритма полиномиальна, если функция  $T(n)$  полиномиальна. Под полиномиальностью функции  $T(n)$  здесь и далее мы понимаем лишь существование такого многочлена  $f(x)$ , что  $T(n) \leq f(n)$  для всех  $n \geq 0$ . Полиномиальная сложность алгоритмов считается относительно низкой по сравнению с экспоненциальной (когда значения  $T(n)$  растут со скоростью геометрической прогрессии) и более высокими степенями сложности. Группу  $G$  отнесем к классу  $\mathbf{P}$ , если проблема слов может быть решена в ней с помощью некоторого алгоритма полиномиальной сложности

(говорят также, что проблема решается за полиномиальное время).

Проблема слов в каждой из групп  $G_1-G_4$  имеет полиномиальную сложность. Например, в случае группы  $G_2$  в этом нетрудно убедиться, подсчитывая число арифметических операций, требуемых для перемножения  $n$  матриц вида  $a^{\pm 1}, b^{\pm 1}$ . (Здесь нужно учесть и удлинение процесса сложения и умножения рациональных чисел вместе с ростом их числителей и знаменателей.)

**ФУНКЦИЯ ДЭНА ГРУППЫ И НЕДЕТЕРМИНИРОВАННЫЕ АЛГОРИТМЫ**

Поскольку слово  $w$  единично в конечно-определенной группе  $G$  тогда и только тогда, когда оно может быть сведено к пустому слову с помощью  $R$ -преобразований, естественно попытаться построить соответствующий  $R$ -алгоритм в виде следующего ветвящегося процесса. Сначала выписываются все слова  $w_{11}, w_{12}, \dots, w_{1k_1}$ , которые можно получить из  $w$  с помощью одного  $R$ -преобразования. Затем применяем всевозможные  $R$ -преобразования к каждому из слов  $w_{11}, w_{12}, \dots, w_{1k_1}$  и получаем все слова  $w_{21}, w_{22}, \dots, w_{2k_2}$ , которые могут возникнуть, если к исходному слову  $w$  применить два  $R$ -преобразования. Все слова  $w_{31}, w_{32}, \dots, w_{3k_3}$  “глубины” 3 аналогично получаются из слов  $w_{21}, w_{22}, \dots, w_{2k_2}$  глубины 2 и т.д. (рис. 2). Если рано или поздно получится пустое слово, то можно заключить, что  $w = e$  в группе  $G$ .

Однако если  $R$ -алгоритм раз за разом выписывает только непустые слова, мы не будем знать, то ли пустое слово будет выдано уже в следующую секунду (а значит,  $w = e$  в  $G$ ), то ли пустое слово встретится через 10 млрд лет работы идеального компьютера (и тогда тоже  $w = e$  в  $G$ ), то ли пустое слово никогда не встретится, машина никогда не остановится (и поэтому  $w \neq e$  в группе  $G$ ). Этот “полуалгоритм” превращается в настоящий алгоритм, только если глубина перебора может быть ограничена значением некоторой алгоритмически вычислимой функции в зависимости от длины исследуемого слова  $w$ . Введем в этой связи функцию Дэна для конечно-определенной группы  $G$ .

Значением  $D(n)$  функции Дэна называют минимальное число  $R$ -преобразований, достаточных для

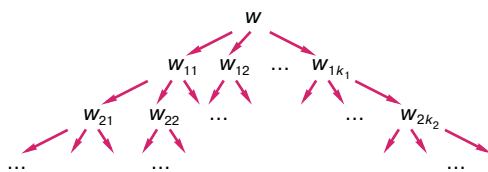


Рис. 2

приведения к пустому слову каждого слова  $w$  длины  $\leq n$ , представляющего в группе  $G$  единицу.

Конечно, функция Дэна может измениться при ином выборе конечных систем порождающих и определяющих соотношений группы  $G$ . Однако не очень сложно доказывается, что новая функция Дэна  $D_1(n)$  и старая функция Дэна  $D(n)$  для некоторой константы  $c > 0$  связаны неравенством  $D_1(n) \leq cD(n) + cn$ . Поэтому свойства функции Дэна быть полиномиальной, экспоненциальной и т.п. зависят только от группы  $G$  и не зависят от конкретного конечного задания этой группы порождающими и соотношениями. (Для читателя, знакомого с понятием риманова многообразия, отметим, что при “хорошем” действии группы  $G$  изометриями многообразия  $X$  функция Дэна для  $G$  оказывается эквивалентной изопериметрической функции  $f(n)$  для  $X$ , которая по определению ограничивает сверху площади двумерных пленок, которыми можно затянуть петли длины не больше  $n$  в  $X$ .)

Предположим, что  $D(n)$  – вычислимая функция натурального аргумента, то есть существует алгоритм, вычисляющий ее значения. (Для функций Дэна это равносильно формально менее сильному условию  $D(n) \leq f(n)$  для некоторой вычислимой функции  $f$ .) Тогда описанный выше процесс можно сделать эффективным, потому что если нужно выяснить, равно ли слово  $w$  длины  $n$  пустому слову в группе  $G$  или нет, то для получения ответа достаточно просмотреть всевозможные полученные  $R$ -преобразованиями цепочки  $w \rightarrow w_{1i_1} \rightarrow \dots \rightarrow w_{li_i}$  длин  $l \leq D(n)$ . Иными словами,  $D(n)$  является достаточной глубиной работы  $R$ -алгоритма при решении проблемы равенства пустому слову для любого слова  $w$  длины  $\leq n$ .

Таким образом, если функция Дэна вычислима, то проблема слов для конечно-определенной группы  $G$  решается с помощью недетерминированного  $R$ -алгоритма. Отличие от детерминированных алгоритмов состоит в том, что на каждом шагу работы  $R$ -алгоритма имеется произвол (хотя и ограниченный) в выборе одной из нескольких возможных команд. (При исследовании написанного на ленте слова можно делать вставки или сокращения, делать разрешенные  $R$ -преобразования видов I–II' либо смещаться влево-вправо вдоль ленты на одну ячейку.)

Временная функция  $T(n)$  произвольного недетерминированного алгоритма определяется как максимум длин кратчайших вычислений для всевозможных входных слов длины, не превосходящей  $n$ .

Конечно-порожденные группы, для которых проблема слов может быть решена за полиномиальное время с помощью некоторого недетерминированного алгоритма, относят к классу NP. В частности, в NP

попадают все группы с полиномиальной функцией Дэна. Понятно, что задача, решаемая недетерминированной машиной  $M$ , может быть решена, хотя и медленнее, и некоторой детерминированной машиной  $M'$ , которая будет перебирать в некотором предписанном порядке все варианты, которые могут встретиться при работе машины  $M$ . Насколько медленнее, чем  $M$ , работает  $M'$ , — очень важный вопрос теории сложности вычислений. Неизвестно, совпадают ли классы  $\mathbf{P}$  и  $\mathbf{NP}$ !

## РОСТ ФУНКЦИИ ДЭНА И СЛОЖНОСТЬ ПРОБЛЕМЫ СЛОВ

Нетрудно установить и обратную зависимость: если проблема слов в некоторой конечно-определенной группе  $G$  алгоритмически разрешима, то функция Дэна  $D(n)$  для  $G$  вычислима. Возникает вопрос, насколько сложность проблемы слов для  $G$  зависит от функции  $D(n)$ .

Пример группы  $G_2$  показывает, что невысокая сложность проблемы слов для группы  $G$  еще не означает, что функция Дэна для нее также растет не слишком быстро. Как мы видели, группа  $G_2$  находится в классе  $\mathbf{P}$  (а значит, и в классе  $\mathbf{NP}$ , поскольку к недетерминированным алгоритмам относятся и детерминированные как частный случай). Но функция Дэна для этой группы экспоненциальна.

Последнее утверждение проверяется с помощью следующей серии слов  $w_k \equiv (a^k b a^{-k}) b (a^k b^{-1} a^{-k}) b^{-1}$  длины  $4k + 4$  для каждого  $k = 1, 2, \dots$

Убедимся, что подслова, заключенные нами в скобки, равны в  $G_2$  соответственно  $b^{2^k}$  и  $b^{-2^k}$ , а значит,  $w_k = e$  в группе  $G_2$ . В самом деле, мы знаем, что  $aba^{-1} = b^2$  в  $G_2$ . Отсюда  $a^2 b a^{-2} = a(aba^{-1})a^{-1} = ab^2 a^{-1} = (aba^{-1}) \times (aba^{-1}) = b^2 b^2 = b^4$ . Аналогично  $a^3 b a^{-3} = b^8$  и т.д. Однако для вывода каждого равенства этой серии требуется вдвое больше преобразований типа  $\Pi'$ , чем для вывода предыдущего, плюс еще одно. (Замена  $aba^{-1}$  на  $b^2$  требует одного преобразования типа  $\Pi'$  с последующими сокращениями.)

Можно показать (хотя это уже неочевидно), что указанный способ приведения слова  $w_k$  к пустому слову оптимален, откуда следует, что функция Дэна группы  $G_2$  растет не медленнее экспоненциальной.

На самом деле пример группы  $G_2$  показывает лишь, что вопрос был поставлен нами в начале раздела не совсем корректно. Нужно было учесть, что если  $G$  — подгруппа некоторой группы  $H$  (то есть  $G$  — подмножество в  $H$  с той же операцией умножения [2]), то проблема слов для  $G$  не может быть сложнее, чем для  $H$ . Поэтому правильнее было спросить о связи сложности проблемы слов в группе  $G$  с функцией Дэна некоторой большей группы  $H$ . Но даже для группы  $G_2$  до последнего

времени не было известно, существует ли содержащая ее в качестве подгруппы группа  $H$  с полиномиальной функцией Дэна. Такая группа  $H$  для  $G_2$  недавно действительно была явно построена А.Ю. Ольшанским и М.В. Сапиром. Но это только проявление общей закономерности.

Опираясь на методы, развитые ранее в работах П.С. Новикова, У. Буна, Г. Хигмэна, Дж. Бриттона, Д. Коллинза, Ч. Миллера, С. Андера, а также на свои недавние результаты, в совместной работе, выполненной в 1998 году, Дж.-К. Бирже, А.Ю. Ольшанский, И. Рипс и М.В. Сапир доказали следующее утверждение.

**Теорема.** *Если проблема слов для некоторой конечно-порожденной группы  $G$  может быть решена с помощью некоторого (необязательно детерминированного) алгоритма с временной функцией  $T(n)$ , то группа  $G$  является подгруппой некоторой конечно-определенной группы  $H$ , функция Дэна которой не превосходит функции, эквивалентной  $n^2 T(n^2)^4$ .*

Основным является

**Следствие.** *Конечно-порожденная группа  $G$  содержится в классе  $\mathbf{NP}$  тогда и только тогда, когда  $G$  является подгруппой некоторой конечно-определенной группы  $H$  с полиномиальной функцией Дэна.*

Разумеется, нужно ставить задачу улучшения приведенной в теореме оценки. Но интересен и промежуточный итог. Упрощая, его можно сформулировать таким образом. Если некоторый алгоритм, возможно очень умный и изощренный (необязательно детерминированный), решает проблему слов в группе  $G$ , то эта проблема может быть решена и с помощью некоторого тупого и прямолинейного недетерминированного алгоритма (а именно  $R$ -алгоритма) за время, ненамного более долгое, чем в первом случае.

## ЛИТЕРАТУРА

1. *Ольшанский А.Ю.* Умножение симметрий и преобразований // Соросовский Образовательный Журнал. 1996. № 5. С. 115–120.
2. *Ольшанский А.Ю.* Групповые исчисления // Там же. № 10. С. 114–119.
3. *Мальцев А.И.* Алгоритмы и рекурсивные функции. М.: Наука, 1986. 368 с.

*Рецензент статьи Ю.Г. Борисович*

\* \* \*

Александр Юрьевич Ольшанский, доктор физико-математических наук, профессор кафедры высшей алгебры механико-математического факультета МГУ. Член редколлегий ряда международных математических журналов. Автор более 60 работ по теории групп и другим вопросам современной алгебры.