

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ЯРОСЛАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. П.Г. ДЕМИДОВА
МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

ТЕОРИЯ АЛГОРИТМОВ И СЛОЖНОСТЬ ВЫЧИСЛЕНИЙ

Методические указания

*Рекомендовано Научно-методическим советом
университета для студентов, обучающихся
по специальности Компьютерная безопасность*

ЯРОСЛАВЛЬ 2010

УДК 512
ББК В14я73
М 34

*Рекомендовано
Редакционно-издательским советом университета
в качестве учебного издания. План 2010 года*

Составитель В.Г. Дурнев

М 34 Материалы по дисциплине “ТЕОРИЯ АЛГОРИТМОВ И
СЛОЖНОСТЬ ВЫЧИСЛЕНИЙ”: метод. указания /
Сост. В.Г. Дурнев; Ярославский гос. ун-т. —
Ярославль: ЯрГУ, 2010. — ?? с.

Методические указания содержат материалы по дисциплине “ТЕОРИЯ АЛГОРИТМОВ И СЛОЖНОСТЬ ВЫЧИСЛЕНИЙ” для студентов, обучающихся по специальности 090102 Компьютерная безопасность: вводный теоретический материал, программу дисциплины, список рекомендованной литературы.

УДК 512
ББК В14я73

© Ярославский государственный университет
им. П.Г. Демидова, 2010

© В.Г. Дурнев, 2010

1. Понятие алгоритма. Машины Тьюринга

Примеры *алгоритмов*, а вместе с ними и расплывчатое, интуитивное понятие *алгоритма* были известны в математике со времен Древнего Египта и Вавилона. Но вплоть до 30-х годов XX века математикам не требовалось точное математическое понятие "*алгоритм*", они вполне довольствовались неточным интуитивным понятием алгоритма. Возникшие *алгоритмические проблемы* решались указанием соответствующих разрешающих процедур. При этом каждый раз, когда конкретный алгоритм для решения той или иной серии однотипных задач был построен, ни у кого не возникало сомнений в том, что указанная процедура является алгоритмом. Практически не было случаев, когда математики разошлись бы во мнениях по вопросу, является ли тот или иной конкретный предлагаемый вычислительный процесс алгоритмом.

В начале XX века были явно сформулированы важные алгоритмические проблемы, в различных разделах математики (алгебре, математической логике, теории чисел) для решения которых не удавалось построить соответствующие алгоритмы, несмотря на усилия многих математиков. Это поставило под сомнение возможность их положительного решения. Но для доказательства невозможности алгоритма, дающего решение той или иной серии задач, следовало точно математически определить, какой смысл мы вкладываем в понятие "*алгоритм*".

Под *алгоритмической проблемой* понимается задача построения единого алгоритма для решения заданной массовой задачи, т. е. бесконечной серии однотипных вопросов, зависящих от некоторых параметров.

В случае, когда искомый алгоритм невозможен, говорят, что данная *алгоритмическая проблема неразрешима*.

Неразрешимость алгоритмической проблемы, конечно, вовсе не означает, что какая-то из задач рассматриваемой серии неразрешима. Это означает лишь, что нельзя решить всю бесконечную серию задач единым методом (при этом не исключается возможность решения каждой из них своим способом).

Под *алгоритмом в интуитивном смысле* мы понимаем точное предписание, определяющее вычислительный процесс, который ведет от исходных данных, варьируемых в некотором заданном множестве, к искомому результату, причем, этот вычислительный процесс должен однозначно определяться заданием предписания и конкретного исходного

данного. Иногда называют алгоритмом и сам вычислительный процесс, определяемый точным предписанием.

Предписание, задающее алгоритм, должно быть конечным. Оно должно быть настолько четким, чтобы оно однозначно определяло искомый вычислительный процесс. Искомый результат должен получаться через конечное число шагов работы алгоритма.

Требование однозначности вычислительного процесса, определяемого алгоритмом, означает, что этот процесс осуществляется **вычислителем** (будь то человек или машина) чисто механически, т. е. без привлечения каких-либо творческих элементов, и может быть воспроизведен с тем же результатом другим вычислителем и в другое время. Иначе говоря, для выполнения вычислительного процесса не требуется никакой информации, которая не содержалась бы в соответствующем точном предписании и рассматриваемом исходном данном.

Математическое уточнение понятия **"алгоритм"**, т. е. замена интуитивного понятия **"алгоритм"** его математическим эквивалентом, стало возможным благодаря развитию математической логики в начале XX века. Оно было получено в середине тридцатых годов в работах К. Геделя, Д. Эрбрана, А. Черча, Э. Поста и А. Тьюринга почти одновременно в двух внешне различных формах: в виде точного математического описания класса вычислимых функций натурального аргумента (частично рекурсивные и рекурсивные функции) и в виде точного математического определения класса вычислительных процессов (машины Тьюринга). Вскоре было установлено, что эти два уточнения понятия алгоритма по существу эквивалентны друг другу, так же как и все другие уточнения, которые появились в науке позже (например, нормальные алгорифмы А.А. Маркова, алгоритмы А.Н. Колмогорова и др.)

Это дало основание уже в 30-х годах высказать тезис о том, что **всякий алгоритм в интуитивном смысле с точки зрения его вычислительных возможностей эквивалентен некоторому алгоритму в уточненном смысле.**

Этот тезис получил название **"Тезис Черча"** по имени американского математика, впервые высказавшего его в 1936 году [30]. В настоящее время **Тезис Черча** является общепризнанным. Его называют **"Тезис Тьюринга"**, если речь идет о машинах Тьюринга, или **"Принцип нормализации Маркова"**, если речь идет о нормальных алгорифмах Маркова.

Появление точного математического понятия **"алгоритм"** позволило установить неразрешимость ряда алгоритмических проблем сначала в самой теории алгоритмов, затем в математической логике, а позже и среди известных задач, поставленных в математике ранее, в частно-

сти, в алгебре и теории чисел.

В качестве конкретного примера математического уточнения понятия алгоритма ниже мы приведем определение машины Тьюринга.

Алфавиты и слова. В качестве исходных данных и искомым результатов алгоритмов употребляются конкретные конструктивные объекты, которые можно легко сравнивать друг с другом и преобразовывать друг в друга, как-то: числа, формулы, кортежи, матрицы и т. д. Очевидно, все конструктивные объекты, рассматриваемые в математике, можно достаточно естественным образом занумеровать натуральными числами, а каждое натуральное число можно записать в виде строки, составленной из обычных цифр или из соответствующего числа палочек, т. е. в виде слова, составленного из одной буквы "|".

Таким образом, наиболее естественными объектами, используемыми алгоритмами в качестве исходных данных и искомым результатов, являются слова в конечных алфавитах.

Алфавитом называется любое непустое множество *символов*, называемых *буквами* алфавита.

Конечная последовательность, составленная из записанных друг за другом букв алфавита \mathcal{A} , называется **словом в алфавите \mathcal{A}** . Мы рассматриваем также и **пустое слово**, которое считается словом в любом алфавите и обозначается через Λ или через ε .

Единственное требование, которое нужно наложить на элементы алфавита \mathcal{A} , заключается в том, чтобы каждое слово в этом алфавите однозначно разбивалось на составляющие его буквы, т. е. чтобы была исключена возможность "разночтения". В частности, это условие будет выполнено, если все буквы алфавита будут связными символами.

Множество всех слов в алфавите \mathcal{A} будем обозначать через \mathcal{A}^* или через $\Omega(\mathcal{A})$.

Длину слова X , т. е. число конкретных букв, из которых оно составлено, будем обозначать через $\partial(X)$ или через $|X|$.

Если X – некоторое слово в алфавите \mathcal{A} , а n – некоторое натуральное число, то через X^n обозначается результат приписывания друг к другу n экземпляров слова X .

Кроме того, для любого слова X полагаем $X^0 \equiv \Lambda$, где символ \equiv обозначает равенство по определению.

Можно ограничиться рассмотрением алгоритмов, перерабатывающих слова в алфавитах

$$\mathcal{A}_n \equiv \{a_1, a_2, \dots, a_n\} \quad (n \geq 1),$$

являющихся начальными отрезками фиксированной счетной последова-

тельности букв:

$$a_1, a_2, a_3, \dots, a_k, a_{k+1}, \dots$$

Рассмотрим однобуквенный алфавит $\{1\}$. Элементы множества $\Omega(\{1\})$ мы будем рассматривать как представления натуральных чисел – каждое натуральное число n однозначно представляется словом длины n из множества $\Omega(\{1\})$. В силу этого полагаем

$$N \equiv \Omega(\{1\}).$$

Отождествив символ "1" с a_1 , мы получим $\mathcal{A}_1 = \{1\}$ и, следовательно,

$$N \subseteq \Omega(\mathcal{A}_n)$$

при любом $n \geq 1$.

Графическое совпадение слов X и Y будем обозначать через $X \equiv Y$.

Говорят, что слово E *входит* в слово X , если можно указать такие слова R и Q , что $X \equiv REQ$. Если при этом слово R (слово Q) пусто, то говорят, что слово E есть *начало* (соответственно, *конец*) слова X .

Одним из первых уточнений интуитивного понятия алгоритма было понятие так называемой **машины Тьюринга**, введённое А. Тьюрингом и уточненное Э. Постом в терминах абстрактных вычислительных машин. **Машины Тьюринга** реализуют алгоритмы переработки слов в некоторых алфавитах. Они копируют работу человека, вычисляющего по заданной программе. Хотя машины Тьюринга – это довольно узкий класс "вычислительных машин", но на них оказалось возможным моделировать все известные в математике алгоритмические процессы.

Мы рассмотрим простейший вариант определения машины Тьюринга – машину Тьюринга с одной лентой.

Машина Тьюринга T задается

- 1) **нешним алфавитом** $A_T = \{a_0, a_1, \dots, a_k\}$,
- 2) **алфавитом внутренних состояний** $Q_T = \{q_0, q_1, \dots, q_l\}$,

Машина Тьюринга T имеет ленту, разбитую на конечное число ячеек, в которые можно записывать символы a_0, a_1, \dots, a_k , при этом по техническим причинам удобно считать, что среди этих символов имеется так называемый **пустой символ**, который мы будем обозначать a_0 . Это означает, что мы придерживаемся следующего соглашения:

высказывание: "в данной ячейке записан символ a_0 " означает, что эта ячейка пустая.

Ячейки ленты упорядочены слева направо.

В процессе работы машины Тьюринга, в случае необходимости, разрешается добавлять новые (пустые) ячейки как к правому, так и к левому концам ленты. В этом смысле иногда говорят, что лента *потенциально бесконечна в обе стороны*. В принципе можно было бы считать ленту бесконечной, но потребовать, чтобы в каждый момент времени лишь конечное число ее ячеек могло содержать символы, отличные от a_0 .

У машины Тьюринга T имеется также **головка**, которая в каждый момент времени обозревает некоторую ячейку и сама находится в одном из данного множества **внутренних состояний** $Q_T = \{q_0, q_1, \dots, q_l\}$.

Работа машины Тьюринга состоит из отдельных тактов, выполняемых согласно заданной программе.

Программа P_T машины Тьюринга T – это конечный набор **команд** следующих трех типов:

$$q_i a_j \rightarrow q_r a_t S, \quad q_i a_j \rightarrow q_r a_t R, \quad q_i a_j \rightarrow q_r a_t L.$$

При этом каждой паре чисел (i, j) ($1 \leq i \leq l$; $0 \leq j \leq k$) соответствует ровно одна команда, в которой перед \rightarrow стоит слово $q_i a_j$. Эту единственную команду будем обозначать через $P_T(i, j)$.

Если в рассматриваемый момент времени головка машины Тьюринга находится в состоянии q_i при $i \neq 0$, а в обозреваемой ею ячейке находится символ a_j , то машина выполняет команду $P_T(i, j)$ следующим образом:

1) если выполняемая команда имеет вид

$$q_i a_j \rightarrow q_r a_t S,$$

то головка заменяет символ a_j в обозреваемой ячейке на символ a_t и переходит в состояние q_r ;

2) если выполняемая команда имеет вид

$$q_i a_j \rightarrow q_r a_t R \quad \text{или} \quad q_i a_j \rightarrow q_r a_t L,$$

то головка заменяет символ a_j в обозреваемой ячейке на символ a_t , переходит в состояние q_r и сдвигается в соседнюю справа (соответственно слева) ячейку.

При этом, если считывающая головка находилась в самой правой (левой) ячейке ленты, и ей надо сдвинуться вправо (влево), то к ленте механически справа (слева) пристраивается новая пустая ячейка.

Если в результате выполнения некоторой команды головка машины приходит в состояние q_0 , называемое **заключительным состоянием**, то машина **останавливается**. Это достигается условием, что

в программе ни одна команда не начинается с так называемого **заключительного состояния** q_0 .

Среди внутренних состояний машины Тьюринга T кроме q_0 выделяют еще и внутреннее состояние q_1 , называемое **начальным внутренним состоянием**.

Для полного описания состояния машины Тьюринга в некоторый момент времени достаточно указать слово, которое записано на ее ленте, обозреваемую ячейку и состояние головки. Все это можно описать в виде слова: Aq_ia_jB , где q_i – состояние головки в данный момент, a_j – символ, записанный в обозреваемой ячейке, а A и B – слова, записанные левее и правее этой ячейки. Слово такого вида Aq_ia_jB называются **конфигурацией**, описывающей полное состояние машины Тьюринга в данный момент. Конфигурация Aq_1a_jB называется **начальной**, а конфигурация Aq_0a_jB – **заключительной**.

Говорят, что машина Тьюринга T , внешний алфавит которой содержит символы 1, 0, **правильно вычисляет** данную 1-местную числовую функцию f , если выполняются следующие 2 требования:

1. Если для данного натурального числа a значение функции f определено и равно числу b , то машина Тьюринга T , начав работать в состоянии, описываемом конфигурацией q_101^a0 , через конечное число тактов работы придет в состояние, описываемое конфигурацией вида $q_001^b0 \dots 0$.

2. Если для натурального числа a функция f не определена, то машина T , начав работу в конфигурации q_101^a0 , никогда не придет во внутреннее состояние q_0 , т. е. не остановится.

Числовая функция называется **правильно вычислимой по Тьюрингу**, если существует машина Тьюринга, правильно вычисляющая эту функцию.

Следующий аналог тезиса Черча также является общепринятым.

Тезис Черча в форме Тьюринга.

Если функция вычислима с помощью алгоритма в интуитивном смысле, то она правильно вычислима по Тьюрингу.

1.1. Алгоритмические проблемы для машин Тьюринга

Будем рассматривать машины Тьюринга, которые вычисляют одноместные функции натурального аргумента при обозначении натуральных чисел словами вида 1^n , и будем обозначать 1 через a_1 . Учитывая, что все остальные внешние символы машины Тьюринга будут иметь вспомогательный характер, мы можем считать, что все они берутся из данного

счетного алфавита

$$\{a_0, a_1, \dots, a_k, \dots\}. \quad (1)$$

Аналогично, внутренний алфавит $Q_T = \{q_0, q_1, \dots, q_l\}$ произвольной машины Тьюринга T можно считать конечным подмножеством данного счетного множества

$$\{q_0, q_1, \dots, q_l, \dots\}. \quad (2)$$

Пусть T – произвольная машина Тьюринга с внешним алфавитом $\{a_0, a_1, \dots, a_k\}$, алфавитом внутренних состояний $\{q_0, q_1, \dots, q_l\}$ и программой P_T .

Будем обозначать символ S через D_0 , символ L – через D_1 , символ R – через D_2 . Тогда можно считать, что каждая команда любой машины Тьюринга T имеет вид

$$q_i a_j \rightarrow q_r a_t D_\varepsilon.$$

Занумеруем все простые числа в порядке возрастания $p_1, p_2, \dots, p_n, \dots$

Теперь, если дана произвольная машина Тьюринга T , то, занумеровав некоторым образом все ее команды в виде:

$$\begin{aligned} q_{i_1} a_{j_1} &\rightarrow q_{r_1} a_{t_1} D_{\varepsilon_1}, \\ q_{i_2} a_{j_2} &\rightarrow q_{r_2} a_{t_2} D_{\varepsilon_2}, \\ &\dots, \\ q_{i_m} a_{j_m} &\rightarrow q_{r_m} a_{t_m} D_{\varepsilon_m}, \end{aligned}$$

закодируем всю программу машины Тьюринга T следующим натуральным числом

$$n(T) \Rightarrow p_1^{i_1} p_2^{j_1} p_3^{r_1} p_4^{t_1} p_5^{\varepsilon_1} p_6^{i_2} p_7^{j_2} p_8^{r_2} p_9^{t_2} p_{10}^{\varepsilon_2} \dots p_{5m-4}^{i_m} p_{5m-3}^{j_m} p_{5m-2}^{r_m} p_{5m-1}^{t_m} p_{5m}^{\varepsilon_m},$$

которое будем называть **номером машины Тьюринга T** .

Очевидно, данная машина Тьюринга может иметь несколько номеров, но не каждое натуральное число является номером какой-то машины Тьюринга. Используя алгоритм разложения натурального числа на простые множители, *легко построить алгоритм, позволяющий по произвольному натуральному числу определить, является ли оно номером некоторой машины Тьюринга, и в случае положительного ответа восстановить программу искомой машины, а также ее внешний и внутренний алфавиты.*

Для машин Тьюринга естественным образом формулируются следующие ниже алгоритмические проблемы.

Проблема остановки:

по произвольной машине Тьюринга T и произвольной ее начальной конфигурации $E_1q_1E_2$ определить, придет ли машина T в заключительное состояние q_0 , если начнет работу в этой конфигурации.

Мы говорим, что машина Тьюринга T **применима** к непустому слову E , если начав работать в конфигурации q_1E , она через конечное число шагов придет во внутреннее состояние q_0 , т. е. остановится.

Простейшим примером алгоритмически неразрешимой проблемы является **проблема распознавания самоприменимости**, которая представляет собой частный случай **проблемы остановки**.

Проблема распознавания самоприменимости:

по произвольной машине Тьюринга T определить, применима ли она к коду своего номера, т. е. к слову $0\overline{n(T)}0$.

Машина Тьюринга T_1 , определенная программой

$$q_ia_j \rightarrow q_00S, (1 \leq i \leq l; 0 \leq j \leq k)$$

является простым примером самоприменимой машины.

Машина Тьюринга T_2 , заданная программой

$$q_ia_j \rightarrow q_10S, (1 \leq i \leq l; 0 \leq j \leq k)$$

не применима ни к какому непустому слову в ее внешнем алфавите и, следовательно, не является самоприменимой.

Теорема 1.1.1. *Невозможен алгоритм, распознающий по любому натуральному числу n является ли оно номером самоприменимой машины Тьюринга.*

Доказательство. Предположим, что существует алгоритм \mathcal{A} , распознающий по любому натуральному числу n является ли оно номером самоприменимой машины Тьюринга.

Обозначим через M_0 множество всех номеров самоприменимых машин Тьюринга, а через $\chi_{M_0}(x)$ – его характеристическую функцию.

Напомним, что если $n \in M_0$, то $\chi_{M_0}(n) = 1$, а если $n \notin M_0$, то $\chi_{M_0}(n) = 0$.

Легко понять, что алгоритм \mathcal{A} позволяет вычислять значения функции $\chi_{M_0}(x)$, т. е. эта функция вычислима в интуитивном смысле. Значит в силу **тезиса Тьюринга** существует машина Тьюринга T_0 , правильно вычисляющая функцию $\chi_{M_0}(x)$. Последнее означает, что

если $n \in M_0$, то машина Тьюринга T_0 , начав работать в конфигурации q_101^n0 , через конечное число шагов остановится в конфигурации $q_0010\dots0$, если же $n \notin M_0$, то машина Тьюринга T_0 , начав работать в

конфигурации $q_1 0 1^{n_1} 0$, через конечное число шагов остановится в конфигурации $q_0 0 0 \dots 0$.

Предположим, что $Q_{T_0} = \{q_0, q_1, \dots, q_m\}$.

Для получения противоречия построим новую машину Тьюринга T_1 следующим образом.

Полагаем

$$A_{T_1} = A_{T_0}, \quad Q_{T_1} = \{q_0, q_1, \dots, q_m, q_{m+1}, q_{m+2}\}.$$

Программа P_{T_1} машины Тьюринга T_1 получается из программы P_{T_0} машины Тьюринга T_0 заменой во всех ее командах внутреннего состояния q_0 на внутреннее состояние q_{m+1} и добавлением к полученному таким образом множеству команд $(P_{T_0})_{q_0}[q_{m+1}]$ следующих трех команд

$$q_{m+1} 0 \rightarrow q_{m+2} 0 R, \quad q_{m+2} 1 \rightarrow q_{m+2} 1 S, \quad q_{m+2} 0 \rightarrow q_0 0 S.$$

Пусть n_1 — это номер машины T_1 .

Предположим, что $n_1 \in M_0$. Тогда по определению множества M_0 машина T_1 применима к слову $0 1^{n_1} 0$, т. е. машина Тьюринга T_1 , начав работать в конфигурации $q_1 0 1^{n_1} 0$, через конечное число шагов остановится, т. е. перейдет в заключительное состояние q_0 . Однако анализ работы машины Тьюринга T_1 показывает, что она в рассматриваемом случае, начав работать в конфигурации $q_1 0 1^{n_1} 0$, через конечное число шагов перейдет в конфигурацию $q_{m+1} 0 1 0 \dots 0$ (на этом начальном этапе машина Тьюринга T_1 работает как машина Тьюринга T_0), а затем перейдет в конфигурацию $0 q_{m+2} 1 0 \dots 0$, в которой "будет оставаться вечно". Значит машина Тьюринга T_1 , начав работать в конфигурации $q_1 0 1^{n_1} 0$, не перейдет через конечное число шагов в заключительное состояние q_0 , т. е. не остановится.

Из полученного противоречия получаем, что наше предположение $n_1 \in M_0$ не верно, поэтому $n_1 \notin M_0$.

Но тогда по определению множества M_0 машина T_1 не применима к слову $0 1^{n_1} 0$, т. е. машина Тьюринга T_1 , начав работать в конфигурации $q_1 0 1^{n_1} 0$, не остановится через конечное число шагов, т. е. не перейдет в заключительное состояние q_0 . Однако анализ работы машины Тьюринга T_1 показывает, что она в рассматриваемом случае, начав работать в конфигурации $q_1 0 1^{n_1} 0$, через конечное число шагов перейдет в конфигурацию $q_{m+1} 0 0 \dots 0$ (на этом начальном этапе машина Тьюринга T_1 работает как машина Тьюринга T_0), а затем перейдет в заключительную конфигурацию $0 q_0 0 \dots 0$. Значит машина Тьюринга T_1 , начав работать в конфигурации $q_1 0 1^{n_1} 0$, через конечное число шагов перейдет в заключительное состояние q_0 , т. е. остановится. Полученное противоречие завершает доказательство теоремы. \square

Замечание. В ходе доказательства теоремы нами установлена *невычислимость функции* χ_{M_0} – характеристической функции множества M_0 номеров самоприменимых машин Тьюринга. Отсюда сразу следует *алгоритмическая неразрешимость проблемы остановки для машин Тьюринга*.

2. Алгоритмические проблемы для конечно определенных полугрупп и групп

2.1. Задание полугрупп и групп образующими элементами и определяющими соотношениями

Универсальный способ задания полугрупп с помощью порождающих элементов и определяющих соотношений был предложен А. Тьюе в работе [40]. Там же он впервые сформулировал алгоритмическую проблему *распознавания равенства слов для конечно определенных полугрупп*, т. е. таких заданий полугрупп, в которых число порождающих и число соотношений конечно.

Пусть задан произвольный алфавит \mathcal{A} и некоторое множество \mathcal{S} упорядоченных пар слов в алфавите \mathcal{A} . Множество \mathcal{A}^* всех слов в алфавите \mathcal{A} образуют свободную полугруппу относительно бинарной операции приписывания одного слова к другому. При этом пустое слово играет роль единицы и обозначается через 1. По сложившейся традиции каждую упорядоченную пару слов $\langle A, B \rangle$ будем записывать в виде $A = B$. Эти равенства, называемые *определяющими соотношениями* используются для определения некоторого отношения эквивалентности, факторизация по которому переводит свободную полугруппу в алфавите \mathcal{A} в искомую полугруппу, заданную определяющими соотношениями множества \mathcal{S} и обозначаемую через

$$\Pi(\mathcal{A}, \mathcal{S}) = \langle \mathcal{A} \mid \mathcal{S} \rangle.$$

Приведем определение отношения равенства слов в таком образом определяемой полугруппе $\Pi(\mathcal{A}, \mathcal{S})$.

Левым элементарным преобразованием, отвечающим определяющему соотношению $A = B$, называется переход вида

$$UAV \rightarrow UB V,$$

где U и V – произвольные слова.

Правым элементарным преобразованием, отвечающим определяющему соотношению $A = B$, называется любой переход вида

$$UBV \rightarrow UAV.$$

В дальнейшем запись $P \rightarrow Q$ будет обозначать, что слово Q может быть получено из слова P с помощью одного элементарного преобразования.

Два слова P и Q в алфавите \mathcal{A} называются *эквивалентными в полугруппе* $\Pi(\mathcal{S})$, если либо $P = Q$, либо существует цепочка элементарных преобразований вида

$$P = P_0 \rightarrow P_1 \rightarrow \dots \rightarrow P_k = Q$$

(записываем $P = Q$ в $\Pi(\mathcal{S})$).

Очевидно, определенное таким образом отношение эквивалентности согласовано с операцией умножения слов, т. е. если $P_1 \equiv Q_1$ и $P_2 \equiv Q_2$, то $P_1 P_2 \equiv Q_1 Q_2$.

Элементами полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ являются классы $[P]$ эквивалентных слов в алфавите \mathcal{A} , которые задаются своими представителями P . Операция умножения классов задается естественным образом

$$[P] \cdot [Q] = [PQ].$$

Если множества \mathcal{A} и \mathcal{S} конечны, то они могут задаваться перечислением их элементов. В таком случае задание полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ может быть записано в виде

$$\langle a_1, \dots, a_n \mid A_1 = B_1, \dots, A_m = B_m \rangle.$$

Для того, чтобы полугруппа с единицей была группой, достаточно чтобы в ней имелись обратные элементы для всех порождающих. При задании групп с помощью образующих и определяющих соотношений это достигается следующим образом.

Для получения группового алфавита к данному полугрупповому алфавиту

$$\mathcal{A} = \{a_1, a_2, \dots, a_i, \dots\}$$

добавляется алфавит "двойников", называемых обратными степенями образующих,

$$\mathcal{A}^{-1} = \{a_1^{-1}, a_2^{-1}, \dots, a_i^{-1}, \dots\}.$$

Пусть теперь \mathcal{S} произвольное множество упорядоченных пар слов в групповом алфавите, записанных в виде равенств $A = B$ и называемых

определяющими соотношениями. Для обеспечения того, чтобы буквы a_i^{-1} были обратными для букв a_i мы добавляем к множеству \mathcal{S} так называемые *тривиальные определяющие соотношения*

$$a_i^{-1}a_i = 1, \quad a_i a_i^{-1} = 1. \quad (3)$$

Очевидно, полугруппа заданная расширенной таким образом системой определяющих соотношений, является группой с единичным элементом 1.

Так как добавление двойников и тривиальных соотношений происходит автоматически каждый раз, то в задании группы обычно они явно не указываются, так что построенная группа называется *группой, заданной образующими \mathcal{A} и множеством определяющих соотношений \mathcal{S}* и обозначается через

$$G(\mathcal{A}, \mathcal{S}) = \langle\langle \mathcal{A} \mid \mathcal{S} \rangle\rangle.$$

Соответствующие тривиальным соотношениям (3) элементарные преобразования

$$U a_i^{-1} a_i V \rightarrow UV \quad \text{и} \quad U a_i a_i^{-1} V \rightarrow UV \quad (4)$$

называются *сокращениями* букв a_i и a_i^{-1} , а обратные им преобразования называются *вставками* этих букв.

Слово X называется *несократимым*, если оно не содержит вхождений подслов вида $a_i a_i^{-1}$ или вида $a_i^{-1} a_i$. Очевидно, используя преобразования (4), произвольное слово X в любой группе G можно преобразовать в равное ему в G несократимое слово.

Напомним, что элементами группы $G(\mathcal{A}, \mathcal{S})$ являются классы $[P]$ эквивалентных слов в групповом алфавите, которые задаются своими представителями P .

Обратным к данному слову вида

$$W = a_{i_1}^{\sigma_1} a_{i_2}^{\sigma_2} \dots a_{i_{k-1}}^{\sigma_{k-1}} a_{i_k}^{\sigma_k},$$

где $\sigma_j = \pm 1$ ($j = 1, 2, \dots, k$), называется слово

$$W^{-1} = a_{i_k}^{-\sigma_k} a_{i_{k-1}}^{-\sigma_{k-1}} \dots a_{i_2}^{-\sigma_2} a_{i_1}^{-\sigma_1}.$$

Очевидно, классы эквивалентности $[W]$ и $[W^{-1}]$ определяют взаимно обратные элементы группы.

Если множество определяющих соотношений \mathcal{S} пусто, то мы получаем *свободную группу с порождающими \mathcal{A}* , которая задана определяющими соотношениями (7). Она обозначается через $\langle\langle \mathcal{A} \mid \emptyset \rangle\rangle$ или просто $F(\mathcal{A})$. Мощность множества \mathcal{A} называется *рангом* свободной группы $F(\mathcal{A})$.

2.2. Проблема равенства для полугрупп и групп.

Так как элементами полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ являются классы $[P]$ эквивалентных слов в алфавите \mathcal{A} , которые задаются своими представителями P , то естественно возникает вопрос о возможности различать элементы этой полугруппы по их представителям. Это приводит к постановке следующих проблем.

Проблема равенства для полугрупп.

Для произвольной полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ и любых двух слов U и V в алфавите \mathcal{A} ее образующих определить, представляют ли они один и тот же элемент этой полугруппы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$ [40].

Проблема равенства для полугрупп рассматривается нами как **алгоритмическая проблема** и понимается как задача построения единого алгоритма, дающего возможность получить ответ на каждый вопрос из бесконечной серии однотипных вопросов вида

дано задание произвольной полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ и два слова U и V в алфавите \mathcal{A} ее образующих; вопрос состоит в следующем: представляют ли они один и тот же элемент этой полугруппы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$

В рассматриваемом случае в качестве параметров конкретных вопросов выступают задание полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ и два слова U и V в алфавите \mathcal{A} ее образующих.

Положительным решением **проблемы равенства для полугрупп** считается соответствующий алгоритм (построение соответствующего алгоритма).

Отрицательным решением **проблемы равенства для полугрупп** считается доказательство невозможности построения соответствующего алгоритма.

Таким образом, **проблема равенства для полугрупп** может быть решена либо в положительном, либо в отрицательном смысле. Нахождение любого из этих решений **проблемы равенства для полугрупп** делает ее решенной (снимает, закрывает эту проблему).

Если в предыдущей формулировке зафиксировать полугруппу $\Pi(\mathcal{A}, \mathcal{S})$, то мы получим более узкую проблему – **проблему равенства для фиксированной полугруппы**.

Проблема равенства для фиксированной полугруппы $\Pi(\mathcal{A}, \mathcal{S})$.

Для произвольных двух слов U и V в алфавите \mathcal{A} образующих полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ определить, представляют ли они один и тот же элемент этой полугруппы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$.

Проблема равенства для фиксированной полугруппы $\Pi(\mathcal{A}, \mathcal{S})$

рассматривается нами как **алгоритмическая проблема** и понимается как задача построения единого алгоритма, дающего возможность получить ответ на каждый вопрос из бесконечной серии однотипных вопросов вида

даны два слова U и V в алфавите \mathcal{A} образующих полу группы $\Pi(\mathcal{A}, \mathcal{S})$; вопрос состоит в следующем: представляют ли они один и тот же элемент этой полу группы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$

В рассматриваемом случае в качестве параметров конкретных вопросов выступают два слова U и V в алфавите \mathcal{A} образующих полу группы $\Pi(\mathcal{A}, \mathcal{S})$.

Положительным решением **проблемы равенства для фиксированной полу группы** $\Pi(\mathcal{A}, \mathcal{S})$ считается соответствующий алгоритм (построение соответствующего алгоритма).

Отрицательным решением **проблемы равенства для фиксированной полу группы** $\Pi(\mathcal{A}, \mathcal{S})$ считается доказательство невозможности построения соответствующего алгоритма.

Таким образом, **проблема равенства для фиксированной полу группы** $\Pi(\mathcal{A}, \mathcal{S})$ может быть решена либо в положительном, либо в отрицательном смысле. Нахождение любого из этих решений **проблемы равенства для фиксированной полу группы** $\Pi(\mathcal{A}, \mathcal{S})$ делает ее решенной (снимает, закрывает эту проблему).

В 1947 году независимо А.А. Марков [16] и Э. Пост [36] опубликовали отрицательное решение **проблемы равенства для фиксированной полу группы**: каждый их них

построил конечно определенную полу группу $\Pi(\mathcal{A}, \mathcal{S})$ и доказал, что невозможно создать алгоритм, позволяющий для любых двух слов U и V в алфавите \mathcal{A} образующих полу группы $\Pi(\mathcal{A}, \mathcal{S})$ определить, представляют ли они один и тот же элемент этой полу группы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$.

Из этого результата А.А. Маркова и Э. Поста, конечно, следует отрицательное решение **проблемы равенства для полу групп**, т. е.

невозможно создать алгоритм, позволяющий по произвольному конечному заданию полу группы $\Pi(\mathcal{A}, \mathcal{S})$ и любым двум словам U и V в алфавите \mathcal{A} ее образующих определить, представляют ли они один и тот же элемент этой полу группы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$.

Таким образом, **проблема равенства для полу групп** решена в отрицательном смысле и это решение можно считать закрывающим (снимающим) эту проблему.

В тоже время **проблема равенства для фиксированной полу группы** для некоторых полу групп решена в отрицательном смысле (до-

казана невозможность построения соответствующего алгоритма), для некоторых полугрупп решена в положительном смысле (построен соответствующий алгоритм), а для некоторых полугрупп остается открытой.

Проблема равенства для конечно определенных групп впервые сформулирована в 1912 году М. Дэном в работе [32]. Заметим, что **проблема равенства для конечно определенных групп** является частным случаем **проблемы равенства для полугрупп**. Приведем ее формулировку.

Проблема равенства для групп.

Для произвольной группы $\Pi(\mathcal{A}, \mathcal{S})$ и любых двух слов U и V в алфавите \mathcal{A} ее образующих определить, представляют ли они один и тот же элемент этой группы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$ [32].

Проблема равенства для групп рассматривается нами как **алгоритмическая проблема** и понимается как задача построения единого алгоритма, дающего возможность получить ответ на каждый вопрос из бесконечной серии однотипных вопросов вида

дано задание произвольной группы $\Pi(\mathcal{A}, \mathcal{S})$ и два слова U и V в алфавите \mathcal{A} ее образующих; вопрос состоит в следующем: представляют ли они один и тот же элемент этой полугруппы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$

В рассматриваемом случае в качестве параметров конкретных вопросов выступают задание группы $\Pi(\mathcal{A}, \mathcal{S})$ и два слова U и V в алфавите \mathcal{A} ее образующих.

Положительным решением **проблемы равенства для групп** считается соответствующий алгоритм (построение соответствующего алгоритма).

Отрицательным решением **проблемы равенства для групп** считается доказательство невозможности построения соответствующего алгоритма.

Таким образом, **проблема равенства для групп** может быть решена либо в положительном, либо в отрицательном смысле. Нахождение любого из этих решений **проблемы равенства для групп** делает ее решенной (снимает, закрывает эту проблему).

Если в предыдущей формулировке зафиксировать группу $\Pi(\mathcal{A}, \mathcal{S})$, то мы получим более узкую проблему – **проблему равенства для фиксированной группы**.

Проблема равенства для фиксированной группы $\Pi(\mathcal{A}, \mathcal{S})$.

Для произвольных двух слов U и V в алфавите \mathcal{A} образующих группы $\Pi(\mathcal{A}, \mathcal{S})$ определить, представляют ли они один и тот же элемент этой группы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$.

Проблема равенства для фиксированной группы $\Pi(\mathcal{A}, \mathcal{S})$ рассматривается нами как **алгоритмическая проблема** и понимается как задача построения единого алгоритма, дающего возможность получить ответ на каждый вопрос из бесконечной серии однотипных вопросов вида

даны два слова U и V в алфавите \mathcal{A} образующих группы $\Pi(\mathcal{A}, \mathcal{S})$; вопрос состоит в следующем: представляют ли они один и тот же элемент этой группы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$

В рассматриваемом случае в качестве параметров конкретных вопросов выступают два слова U и V в алфавите \mathcal{A} образующих группы $\Pi(\mathcal{A}, \mathcal{S})$.

Положительным решением **проблемы равенства для фиксированной группы** $\Pi(\mathcal{A}, \mathcal{S})$ считается соответствующий алгоритм (построение соответствующего алгоритма).

Отрицательным решением **проблемы равенства для фиксированной группы** $\Pi(\mathcal{A}, \mathcal{S})$ считается доказательство невозможности построения соответствующего алгоритма.

Таким образом, **проблема равенства для фиксированной группы** $\Pi(\mathcal{A}, \mathcal{S})$ может быть решена либо в положительном, либо в отрицательном смысле. Нахождение любого из этих решений **проблемы равенства для фиксированной группы** $\Pi(\mathcal{A}, \mathcal{S})$ делает ее решенной (снимает, закрывает эту проблему).

В 1955 году П.С. Новиков в работе [24] получил отрицательное решение **проблемы равенства для фиксированной группы**: он

построил конечно определенную группу $\Pi(\mathcal{A}, \mathcal{S})$ и доказал, что невозможно создать алгоритм, позволяющий для любых двух слов U и V в алфавите \mathcal{A} образующих группы $\Pi(\mathcal{A}, \mathcal{S})$ определить, представляют ли они один и тот же элемент этой группы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$.

Из этого результата П.С. Новикова, конечно, следует отрицательное решение **проблемы равенства для групп**, т. е.

невозможно создать алгоритм, позволяющий по произвольному конечному заданию группы $\Pi(\mathcal{A}, \mathcal{S})$ и любым двум словам U и V в алфавите \mathcal{A} ее образующих определить, представляют ли они один и тот же элемент этой группы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$.

Таким образом, **проблема равенства для групп** решена в отрицательном смысле и это решение можно считать закрывающим (снимающим) эту проблему.

В тоже время **проблема равенства для фиксированной группы** для некоторых групп решена в отрицательном смысле (доказана невозможность построения соответствующего алгоритма), для некоторых

групп решена в положительном смысле (построен соответствующий алгоритм), а для некоторых групп остается открытой.

Очевидно, в группе G два слова X и Y равны, тогда и только тогда, когда слово XY^{-1} равно пустому слову 1 в G . Поэтому все определяющие соотношения группы $G(\mathcal{A}, \mathcal{S})$ вида $A = B$ из множества \mathcal{S} можно записать в виде $AB^{-1} = 1$, а сама проблема равенства для данной группы G , заданной образующими и определяющими соотношениями, может быть сформулирована так:

по любому слову W в алфавите группы G определить, равно W пустому слову в G или нет.

Проблема равенства для групп и полугрупп называется также **проблемой тождества**.

Для свободной группы $F(\mathcal{A})$ проблема тождества имеет весьма простое решение, так как легко доказывается, что *данное слово W равно 1 в $F(\mathcal{A})$ тогда и только тогда, когда оно превращается в пустое слово в результате полного сокращения*.

На протяжении ряда лет усилия многих математиков были направлены на поиски алгоритмов, решающих проблему тождества для конкретных конечно определенных групп. В частности, еще в 1932 г. В. Магнус установил разрешимость проблемы тождества для групп с одним определяющим соотношением. Позже появились исследования многих авторов, относящиеся к так называемым группам с малым сокращением (В.А. Тартаковский и др.) Большой интерес математиков к этой проблеме как до появления точного понятия алгоритма, так и после объясняется тем, что она является центральной проблемой комбинаторной теории групп.

Трудности, стоявшие на пути решения этой проблемы, были связаны с тем, что в начале XX века группы, заданные образующими элементами и определяющими соотношениями, были мало исследованы. Более того, была некоторая надежда, что алгоритм, решающий проблему тождества, существенно облегчит изучение таких групп.

Как уже было сказано выше в 1955 году П.С. Новиков [24] доказал *алгоритмическую неразрешимость проблемы тождества теории групп*, построив пример конечно определенной группы, для которой требуемый алгоритм невозможен.

Эти исследования П.С. Новикова относятся как к алгебре, так и к математической логике. Дело не столько в том, что уточнение понятия алгоритма возникло в недрах математической логики, сколько в том, что группы с определяющими соотношениями задаются посредством так называемых групповых исчислений, которые родственны логическим фор-

мальным системам. Успех был достигнут благодаря созданному П.С. Новиковым методу исследования преобразований слов в групповых исчислениях.

2.3. Теорема А.А. Маркова – Э. Поста

В 1947 году А.А. Марков [16] и Э. Пост [36] независимо указали способ построения по произвольной машине Тьюринга T с внешним алфавитом $A_T = \{a_0, a_1, \dots, a_n\}$, алфавитом внутренних состояний $Q_T = \{q_0, q_1, \dots, q_m\}$ и программой P_T конечно определенной полугруппы $S(T)$, заданной множеством образующих элементов

$$\{a_0, a_1, \dots, a_n, q_0, q_1, \dots, q_m, h\}$$

и множеством определяющих соотношений $\mathcal{R}(T)$, которое строится по программе P_T машины Тьюринга T следующим образом.

Рассмотрим произвольную команду $P_T(i, j)$ ($1 \leq i \leq m; 0 \leq j \leq n$).

I. Если команда $P_T(i, j)$ имеет вид

$$q_i a_j \rightarrow q_r a_t S,$$

то включаем во множество определяющих соотношений $\mathcal{R}(T)$ одно соотношение $q_i a_j = q_r a_t$.

II. Если команда $P_T(i, j)$ имеет вид

$$q_i a_j \rightarrow q_r a_t L,$$

то для каждого числа s ($s = 0, \dots, n$) включаем во множество определяющих соотношений $\mathcal{R}(T)$ одно соотношение $a_s q_i a_j = q_r a_s a_t$.

Кроме того в этом случае включаем одно дополнительное соотношение $h q_i a_j = h q_r a_0 a_t$.

III. Если команда $P_T(i, j)$ имеет вид

$$q_i a_j \rightarrow q_r a_t R,$$

то для каждого числа s ($s = 0, \dots, n$) включаем во множество определяющих соотношений $\mathcal{R}(T)$ одно соотношение $q_i a_j a_s = a_t q_r a_s$.

Кроме того в этом случае включаем одно дополнительное соотношение $q_i a_j h = a_t q_r a_0 h$.

IV. Для каждого j ($j = 0, \dots, n$) включаем во множество $\mathcal{R}(T)$ два соотношения

$$q_0 a_j = q_0, \quad a_j q_0 = q_0.$$

Кроме того, включаем во множество определяющих соотношений $\mathcal{R}(T)$ еще одно соотношение $hq_0h = q_0$.

Словом Поста будем называть любое слово вида $hE_1q_iE_2h$, где $E_1q_iE_2$ – конфигурация, т. е. E_1, E_2 – слова во внешнем алфавите A_T машины Тьюринга T , причем слово E_2 не пусто.

Для произвольной машины Тьюринга T и произвольных ее конфигураций X и Y запись $X \models_T Y$ будет означать, что машина Тьюринга T за один такт работы переводит конфигурацию X в конфигурацию Y , а запись $X \vdash_T Y$ будет означать, что машина Тьюринга T за конечное число тактов работы переводит конфигурацию X в конфигурацию Y .

Запись $X \Vdash_T$ мы будем использовать для сокращенной записи утверждения "машина Тьюринга T , начав работать в конфигурации X , через конечное число тактов работы остановится."

Теорема 2.3.1. *Машина Тьюринга T , начав работать в конфигурации X , через конечное число тактов работы остановится тогда и только тогда, когда в полугруппе $S(T)$ выполняется равенство $hXh = q_0$.*

Напомним, что для конечно определенной полугруппы

$$S = \langle a_1, \dots, a_n \mid A_1 = B_1, \dots, A_m = B_m \rangle$$

переход

$$XA_iY \rightarrow XB_iY$$

называется *левым элементарным преобразованием*, а переход

$$XB_iY \rightarrow XA_iY$$

называется *правым элементарным преобразованием*.

Доказательство теоремы основывается на следующих двух очевидных леммах.

Лемма 2.3.1. *Если машина Тьюринга T за один такт работы переводит конфигурацию X в конфигурацию X_1 , то для слов Поста hXh и hX_1h , соответствующих этим конфигурациям, переход $hXh \rightarrow hX_1h$ является левым элементарным преобразованием полугруппы $S(T)$, отвечающим определяющему соотношению из групп I – III.*

Лемма 2.3.2. *Если hXh – слово Поста, соответствующее конфигурации X машины Тьюринга T , а переход $hXh \rightarrow Y_1$ является левым элементарным преобразованием полугруппы $S(T)$, отвечающим определяющему соотношению из групп I – III, то найдется такая конфигурация X_1 , что $Y_1 = hX_1h$ – слово Поста, соответствующее конфигурации X_1 , и машина Тьюринга T за один такт работы переводит конфигурацию X в конфигурацию X_1 .*

Доказательство. теоремы 2.3.1. Если машина Тьюринга T , начав работать в конфигурации X , через конечное число тактов работы остановится, то для некоторого натурального числа k имеем

$$X = X_0 \models_T X_1 \models_T X_2 \models_T \dots \models_T X_{k-1} \models_T X_k = Z_1 q_0 Z_2,$$

где $X_0, X_1, X_2, \dots, X_{k-1}, X_k$ – последовательные конфигурации, которые проходит машина Тьюринга T , начав работать в конфигурации X , до своей остановки.

Тогда по лемме 2.3.1 найдутся такие слова Поста $Y_0, Y_1, Y_2, \dots, Y_{k-1}, Y_k$, что в полугруппе $S(T)$ имеет место цепочка элементарных преобразований

$$hXh = Y_0 \rightarrow Y_1 \rightarrow Y_2 \rightarrow \dots \rightarrow Y_{k-1} \rightarrow Y_k = hZ_1 q_0 Z_2 h.$$

Используя определяющие соотношения из IV-ой группы полугруппы $S(T)$, получаем в полугруппе $S(T)$ равенство $Y_k = q_0$, а значит в полугруппе $S(T)$ выполняется и равенство $hXh = q_0$.

Предположим, что в полугруппе $S(T)$ выполняется равенство $hXh = q_0$. Тогда существует цепочка элементарных преобразований

$$hXh = Y_0 \rightarrow Y_1 \rightarrow Y_2 \rightarrow \dots \rightarrow Y_{k-1} \rightarrow Y_k = q_0. \quad (5)$$

Очевидно, в каждое слово Y_j из последовательности (5) входит только одна q -буква.

Можно считать, что символ q_0 не входит в слово X .

Рассмотрим кратчайшую цепочку элементарных преобразований (5), заканчивающуюся словом, содержащим символ q_0 :

$$hXh = Y_0 \rightarrow Y_1 \rightarrow Y_2 \rightarrow \dots \rightarrow Y_{r-1} \rightarrow Y_r = hZ_1 q_0 Z_2 h. \quad (6)$$

Очевидно, что в цепочке элементарных преобразований (6) не используются определяющие соотношения полугруппы $S(T)$ из IV-ой группы и последнее в этой цепочке преобразование

$$Y_{r-1} \rightarrow Y_r$$

не может быть правым преобразованием.

Покажем, что в цепочке (6) вообще нет правых преобразований.

Рассмотрим самое последнее правое элементарное преобразование в (6), если они вообще есть,

$$Y_t \rightarrow Y_{t+1}. \quad (7)$$

Тогда $t + 1 < r$ и легко видеть, что в переходе (7) используется определяющее соотношение из групп I, II и III, а поэтому в переходе

$$Y_{t+1} \rightarrow Y_{t+2} \quad (8)$$

может использоваться лишь то же самое определяющее соотношение, но в противоположную сторону (переход (7) был правым элементарным преобразованием, соответствующим некоторому определяющему соотношению, а переход (8) оказался левым элементарным преобразованием, соответствующим тому же определяющему соотношению). Значит $Y_{t+2} = Y_t$. А это противоречит условию минимальности последовательности (6).

Для завершения доказательства теоремы 2.3.1 остается воспользоваться леммой 2.3.2. \square

Так как **проблема останковки** для машин Тьюринга алгоритмически неразрешима, то из теоремы 2.3.1 получаем следующую теорему А.А. Маркова – Э. Поста.

Теорема 2.3.2 (А.А. Марков – Э. Пост). *Проблема равенства для конечно определенных полугрупп алгоритмически неразрешима, т. е. невозможно построить алгоритм, позволяющий для произвольной конечно определенной полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ и любых двух слов U и V в алфавите \mathcal{A} определить, представляют ли они один и тот же элемент этой полугруппы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$.*

Используя конструкцию М. Холлом [39] получим некоторое усиление предыдущей теоремы.

Теорема 2.3.3. *Алгоритмически неразрешима проблема равенства для конечно определенных полугрупп с 2 образующими элементами, т. е. невозможно построить алгоритм, позволяющий для произвольной конечно определенной полугруппы $\Pi(\mathcal{A}, \mathcal{S})$ с 2 образующими элементами (множество \mathcal{A} состоит из 2 элементов) и любых двух слов U и V в алфавите \mathcal{A} определить, представляют ли они один и тот же элемент этой полугруппы, т. е. эквивалентны ли они в $\Pi(\mathcal{A}, \mathcal{S})$.*

Доказательство. Рассмотрим произвольную конечно определенную полугруппу

$$\Pi = \langle a_1, \dots, a_n \mid A_1 = B_1, \dots, A_m = B_m \rangle.$$

Пусть a и b – две новые буквы. Для произвольного слова P в алфавите $\{a_1, \dots, a_n\}$ обозначим через P^* слово в алфавите $\{a, b\}$, полученное из

слова P заменой каждого вхождения каждой буквы a_i ($i = 1, \dots, n$) на $aba^{i+1}b^{i+1}$.

Обозначим через Π^* конечно определенную полугруппу

$$\langle a, b \mid A_1^* = B_1^*, \dots, A_m^* = B_m^* \rangle,$$

тогда отображение $P \rightarrow P^*$ является изоморфным вложением полугруппы Π в полугруппу Π^* , т. е. для любых слов P и Q в алфавите $\{a_1, \dots, a_n\}$ имеет место эквивалентность

$$P = Q \text{ в полугруппе } \Pi \iff P^* = Q^* \text{ в полугруппе } \Pi^*.$$

□

Существенно более сложное рассуждение показывает, что *существует конкретная машина Тьюринга T_u , для которой алгоритмически неразрешима проблема остановки*. Это дает возможность доказать следующую теорему.

Теорема 2.3.4. *Существует конечно определенная полугруппа с двумя образующими элементами*

$$S = \langle a, b \mid A_1 = B_1, \dots, A_m = B_m \rangle,$$

для которой алгоритмически неразрешима проблема равенства, т. е. невозможно построить алгоритм, позволяющий для любых двух слов U и V в алфавите $\{a, b\}$ определить, представляют ли они один и тот же элемент этой полугруппы, т. е. эквивалентны ли они в S .

Особый интерес представляет вопрос о существовании конечно определенных полугрупп $\Pi(\mathcal{A}, \mathcal{S})$ с неразрешимой проблемой равенства и с "простым" множеством определяющих соотношений \mathcal{S} .

Наиболее простые системы определяющих соотношений \mathcal{S} , задающие полугруппы с неразрешимой проблемой равенства были построены в 1957 году независимо Г.С. Цейтиным [28] и Д. Скоттом [43]. Оба эти задания содержат 7 определяющих соотношений. Г.С. Маканин [12] построил конечно определенную полугруппу с неразрешимой проблемой равенства, заданную 5 определяющими соотношениями, а Ю.В. Матиясевич [21] построил соответствующий пример полугруппы с 3 определяющими соотношениями.

Из результатов А.А. Маркова и Э. Поста следует неразрешимость еще одной алгоритмической проблемы для полугрупп.

Общая проблема изоморфизма для полугрупп.

По произвольным двум заданиям конечно определенных полугрупп определить, изоморфны ли задаваемые ими полугруппы.

Аналогичным образом формулируется **Общая проблема изоморфизма для групп**, которая впервые рассматривалась Тие еще в 1908 году в работе [41].

Общая проблема изоморфизма для групп.

По произвольным двум заданиям конечно определенных групп определить, изоморфны ли задаваемые ими группы.

С каждой фиксированной конечно определенной полугруппой Π_0 связывается частная проблема изоморфизма.

Частная проблема изоморфизма для полугруппы Π_0 .

По произвольной конечно определенной полугруппе Π определить, изоморфна она полугруппе Π_0 или нет.

Неразрешимость **частной проблемы изоморфизма** для каждой конечно определенной полугруппы с n образующими в классе всех конечно определенных полугрупп с $n+3$ образующими установлена А.А. Марковым.

Г.С. Цейтин [27] усилил этот результат, показав, что число $n+3$ можно заменить на число $n+2$. И при этом отметил, что заменить на число $n+1$, вообще говоря, нельзя: свободная полугруппа с n образующими имеет разрешимую частную проблему изоморфизма в классе всех конечно определенных полугрупп с $n+1$ образующим. Это следует из следующих простых рассуждений.

Пусть $\Pi_n = \langle a_1, \dots, a_n \rangle$ – свободная полугруппа с n образующими, а

$$\Pi = \langle b_1, \dots, b_n, b_{n+1} \mid A_1 = B_1, \dots, A_m = B_m \rangle -$$

произвольная конечно определенная полугруппа с $n+1$ образующим.

Можно считать, что ни одно определяющее соотношение не имеет вид $b_i = W(b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_m)$ или вид $W(b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_m) = b_i$.

Предположим, что полугруппа Π изоморфна свободной полугруппе Π_n , а φ – соответствующий изоморфизм.

Если хотя бы одно из определяющих слов A_i, B_i ($i = 1, \dots, m$) пусто, то без ограничения общности можем считать, что пусто некоторое слово B_{i_0} . Тогда для любого b_j , входящего в запись соответствующего слова A_{i_0} , выполняется условие: $\varphi(b_j)$ – пустое слово. Но свободная полугруппа Π_n не может порождаться менее чем n элементами. Значит найдется такое j , что все определяющие соотношения $A_i = B_i$ с пустой правой частью имеют вид $b_j^{k_i} = 1$. Без ограничения общности можем считать, что $j = n+1$. Все определяющие соотношения вида $b_{n+1}^{k_j} = 1$ можем

заменить одним, им равносильным, определяющим соотношением вида $b_{n+1}^d = 1$.

Нетрудно понять, что если одна из частей определяющего соотношения является степенью b_{n+1} , то и другая часть является степенью b_{n+1} . Все определяющие соотношения такого вида, а в их число входит и определяющее соотношение $b_{n+1}^d = 1$, заменим одним, им равносильным соотношением вида $b_{n+1}^q = 1$. Все остальные определяющие соотношения удовлетворяют условию: в каждую их часть входит буква, отличная от b_{n+1} . Так как $\varphi(b_{n+1})$ – пустое слово, то $b_{n+1} = 1$ в полугруппе Π . Нетрудно понять, что в силу выше сказанного это выполнено тогда и только тогда, когда $q = 1$. В итоге мы можем получить задание полугруппы Π с помощью n образующих элементов и некоторого конечного множества определяющих соотношений, причем существование изоморфизма φ влечет, что все эти соотношения тривиальны, т. е. имеют вид $A = A$.

Если ни одно из определяющих слов A_i, B_i ($i = 1, \dots, m$) не является пустым, то полугруппа Π не может быть изоморфна свободной полугруппе Π_n (напомним, мы предположили, что ни одно определяющее соотношение не имеет вид $b_i = W(b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_m)$ или вид $W(b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_m) = b_i$).

Итак в силу результата Г.С. Цейтина для любой конечно определенной полугруппы с n образующими элементами неразрешима частная проблема изоморфизма в классе всех конечно определенных полугрупп с $n + 2$ образующими элементами. Существуют конечно определенные полугруппы с n образующими элементами, например, свободная полугруппа ранга n , имеющие разрешимую частную проблему изоморфизма в классе всех конечно определенных полугрупп с $n + 1$ образующим элементом.

Покажем, что при любом $n \geq 2$ существуют конечно определенные полугруппы с n образующими элементами и m определяющими соотношениями, имеющие неразрешимую частную проблему изоморфизма в классе всех конечно определенных полугрупп с n образующими элементами и $m + 1$ определяющим соотношением.

Для произвольной конечно определенной полугруппы Π и слов X и Y в алфавите ее образующих обозначим через $\Pi_{X,Y}$ полугруппу, задание которой получается из задания полугруппы Π добавлением к множеству определяющих соотношений нового соотношения $X = Y$.

Пусть Π – произвольная конечно определенная полугруппа без пустых определяющих слов, имеющая неразрешимую проблему равенства. В качестве такой полугруппы можно взять, например, построенную в ходе доказательства теоремы 2.3.2 полугруппу А.А. Маркова – Э. Поста

$S(T)$. Применив к полугруппе Π описанную выше конструкцию М. Холла, получим полугруппу Π^* с 2 образующими элементами, имеющую неразрешимую проблему равенства и такую, что каждое определяющее слово из ее задания имеет длину ≥ 2 . Очевидно, для любой образующей a полугруппы Π^* класс $[a]$ состоит из одного слова a , так как к слову a длины 1 неприменимо ни одно определяющее соотношение этой полугруппы. Поэтому любой гомоморфизм этой полугруппы на себя индуцирует подстановку на множестве образующих, а так как некоторая степень этой подстановки дает тождественную подстановку, то данный произвольный гомоморфизм является автоморфизмом. Отсюда следует, что

полугруппы Π^ и $\Pi_{X,Y}^*$ изоморфны тогда и только тогда, когда $X = Y$ в полугруппе Π .*

3. Неразрешимость исчисления предикатов

Первым примером алгоритмически неразрешимой проблемы, возникшей в математике вне теории алгоритмов и даже до ее создания, была проблема проверки является ли данная формула логики предикатов первого порядка тождественно истинной (выводимой). Эта проблема была сформулирована в начале XX века Д. Гильбертом после создания логики предикатов первого порядка и оставалась нерешенной в течение нескольких лет. Она получила название *проблемы разрешимости* (Entscheidungsproblem) логики (исчисления) предикатов. Невозможность соответствующего алгоритма была доказана А. Черчем в 1936 году [31].

Теорема 3.0.5. *Невозможен алгоритм, позволяющий по произвольной замкнутой формуле логики предикатов первого порядка с равенством, алфавит которой содержит 2-местный функциональный символ, определить, является ли эта формула тождественно истинной.*

Доказательство. Обозначим через \cdot двуместный функциональный символ, входящий по предположению в алфавит рассматриваемого исчисления предикатов первого порядка с равенством. Пусть Π – конечно определенная полугруппа с двумя образующими элементами, имеющая алгоритмически неразрешимую проблему равенства слов. Можно считать, что полугруппа Π имеет задание

$$\langle a, b \mid A_1(a, b) = B_1(a, b), A_2(a, b) = B_2(a, b), \dots, A_m(a, b) = B_m(a, b) \rangle.$$

Каждое слово $W = w_1 w_2 w_3 \dots w_n$ мы рассматриваем как терм

$$(\dots((w_1 \cdot w_2) \cdot w_3) \cdot \dots) \cdot w_n,$$

который, чтобы не усложнять обозначений, будем по-прежнему обозначать через W .

Для произвольных двух слов $U(a, b)$ и $V(a, b)$ в алфавите $\{a, b\}$ обозначим через $\Phi_{U,V}$ следующую формулу

$$\begin{aligned} (\forall x)(\forall y)(\forall z)(x \cdot (y \cdot z) = (x \cdot y) \cdot z) \rightarrow \\ \rightarrow (\forall u)(\forall v)((\&_{i=1}^m A_i(u, v) = B_i(u, v)) \rightarrow U(u, v) = V(u, v)). \end{aligned}$$

Нетрудно понять, что имеет место следующая эквивалентность

$$X = Y \text{ в полугруппе } \Pi \iff \text{формула } \Phi_{U,V} \text{ тождественно истинна.}$$

Это завершает доказательство теоремы 3.0.5. \square

Замечание. Аналогичную теорему можно доказать и для *чистого* исчисления предикатов первого порядка с равенством, т. е. исчисления, алфавит которого не содержит функциональных символов.

Предположим, что алфавит чистого исчисления предикатов первого порядка содержит 3-местный предикатный символ p . Обозначим через $F_{U,V}$ следующую формулу

$$\begin{aligned} (\forall x_1)(\forall x_2)(\exists y)(\forall y_1)(p(x_1, x_2, y) \& (p(x_1, x_2, y_1) \rightarrow y_1 = y)) \rightarrow \\ \rightarrow (\Phi_{U,V})^*, \end{aligned}$$

где для произвольной формулы Φ через Φ^* обозначена формула, полученная заменой функционального символа \cdot на предикатный символ p по следующему правилу:

1. по формуле Φ строим равносильную ей формулу Φ_1 заменой в формуле Φ каждой подформулы вида $t_1 = t_2$, где t_1, t_2 - термы, на равносильную ей формулу вида

$$(\exists v_1)(\exists v_2) \dots (\exists v_m) \Psi, \text{ где } \Psi = \&_{i=1}^k \varphi_i$$

и каждая из формул φ_i имеет вид $x \cdot y = z$;

2. по формуле Φ_1 строим равносильную ей формулу Φ^* заменой в формуле Φ_1 каждой подформулы вида $x \cdot y = z$ на формулу $p(x, y, z)$.

Нетрудно понять, что имеет место следующая эквивалентность

$$X = Y \text{ в полугруппе } \Pi \iff \text{формула } F_{U,V} \text{ тождественно истинна.}$$

Аналогичное утверждение можно доказать и для чистого исчисления предикатов первого порядка, алфавит которого содержит по крайней мере один 2-местный предикатный символ, но сделать это достаточно трудно. В то же время алгоритмически разрешим вопрос о тождественной истинности формул для исчисления одноместных предикатов.

4. Другие подходы к построению невычислимых функций

Во многих книгах, статьях и учебниках по теории алгоритмов, например, в [2], [6], [10], [14], [23] и [29] строятся примеры невычислимых функций. Выше в качестве примера невычислимой функции была указана характеристическая функция χ_{M_0} множества M_0 номеров самоприменимых машин Тьюринга. Следует однако заметить, что эти построения, как правило базируются на канторовском диагональном методе, использующем нумерацию всех машин Тьюринга, всех вычислимых по Тьюрингу функций и т. д. Оказывается существуют и другие, в определенном смысле, более прямые способы построения невычислимых функций.

В качестве примера рассмотрим изученную Тибором Радо задачу, получившую название *Проблема усердного бобра* [37], [4].

Ограничимся рассмотрением лишь машин Тьюринга T , внешний алфавит которых \mathcal{A}_T содержит только два символа 0 и 1. Конфигурации вида $q_i 0 1^m 0 \dots 0$ будем называть *стандартными* конфигурациями.

Напомним, что одноместная функция $f(x)$ **вычисляется** машиной Тьюринга T , если выполнены следующие два условия:

1) если натуральное число n принадлежит области определения $D(f)$ функции f и $f(n) = m$, то начав работать в стандартной конфигурации $q_1 0 1^n 0$, машина Тьюринга T остановится в конфигурации вида $C q_0 D$ и $m = |C q_0 D|_1$, где через W_1 мы обозначаем число 1 в слове W ;

2) если натуральное число n не принадлежит области определения $D(f)$ функции f , то начав работать в стандартной конфигурации $q_1 0 1^n 0$, машина Тьюринга T никогда не остановится.

Одноместная функция $f(x)$ **правильно вычисляется** машиной Тьюринга T , если выполнены следующие два условия:

1) если натуральное число n принадлежит области определения $D(f)$ функции f и $f(n) = m$, то начав работать в стандартной конфигурации $q_1 0 1^n 0$, машина Тьюринга T остановится в стандартной конфигурации $q_0 0 1^m 0 \dots 0$;

2) если натуральное число n не принадлежит области определения

$D(f)$ функции f , то начав работать в стандартной конфигурации $q_1 0 1^n 0$, машина Тьюринга T никогда не остановится.

Введем понятие **продуктивность машины Тьюринга T** в смысле Тибора Радо.

*Если машина Тьюринга T , начав работу с пустой лентой, т. е. в стандартной конфигурации $q_1 0$, не остановится, то ее **продуктивность** не определена.*

*Если машина Тьюринга T , начав работу с пустой лентой, остановится в конфигурации вида $Cq_0 D$, то ее **продуктивность** $pr(T)$ – это $|Cq_0 D|_1$. Т. е. **продуктивность** $pr(T)$ машины Тьюринга T – это число единиц, которое она напечатает на пустой ленте до остановки.*

Напомним, что каждая машина Тьюринга T вычисляет для любого k единственную k -местную функцию $f_T^{(k)}$. В частности, любая машина Тьюринга T вычисляет единственную одноместную функцию $f_T^{(1)}$.

В этих терминах, **продуктивность** $pr(T)$ машины Тьюринга T в смысле Т. Радо – это просто значение $f_T^{(1)}(0)$, если оно определено, т. е. если $0 \in D(f_T^{(1)})$. Если же значение $f_T^{(1)}(0)$ не определено, т. е. если $0 \notin D(f_T^{(1)})$, то и продуктивность $pr(T)$ машины Тьюринга T не определена.

Через $p(n)$ Т. Радо обозначает наибольшее значение $pr(T)$ для всех машин Тьюринга с n внутренними состояниями, отличными от заключительного состояния q_0 , и с внешним алфавитом, содержащим лишь два символа 0 и 1, т. е. $p(n)$ – это продуктивность самой продуктивной машины Тьюринга с n внутренними состояниями и с внешним алфавитом, содержащим лишь два символа 0 и 1. Легко понять, что $p(n)$ – всюду определенная функция.

Мы считаем, что утверждение

” T – машина Тьюринга с n внутренними состояниями, отличными от заключительного состояния q_0 , и с внешним алфавитом, содержащим лишь два символа 0 и 1”,

означает, что

$$A_T = \{0, 1\} \text{ и } Q_T = \{q_0, q_1, \dots, q_n\}.$$

Установим некоторые необходимые для доказательства свойства функции $p(n)$.

Прежде всего покажем, что функция $p(n)$ строго возрастает, т. е. для любого n выполняется неравенство $p(n+1) > p(n)$.

Обозначим через T_n самую продуктивную машину Тьюринга с n внутренними незаключительными состояниями q_1, \dots, q_n и заключительным состоянием q_0 .

Через M обозначим машину Тьюринга с $n+1$ внутренним незаключительным состоянием q_1, \dots, q_n, q_{n+1} и заключительным состоянием q_0 , программа P_M которой получается из программы P_{T_n} машины T_n заменой в ее командах заключительного состояния q_0 на новое состояние q_{n+1} и добавлением двух новых команд

$$q_{n+1}1 \rightarrow q_{n+1}1L, \quad q_{n+1}0 \rightarrow q_01S.$$

Нетрудно понять, что построенная машина Тьюринга M с $n+1$ внутренним незаключительным состоянием переводит стандартную конфигурацию q_10 в конфигурацию вида Cq_0D , где $|Cq_0D|_1 = p(n) + 1$, поэтому

$$p(n+1) \geq p(n) + 1 > p(n).$$

Покажем, что для любого n выполняется неравенство

$$p(n+16) \geq 2n.$$

Число 16 здесь не играет ключевой роли. Для дальнейшего достаточно бы было доказать, что существует такое число n_0 , что для любого n выполняется неравенство

$$p(n+n_0) \geq 2n.$$

В качестве n_0 можно взять число незаключительных внутренних состояний любой "Удваивающей" машины Тьюринга, т. е. такой машины D , которая переводит стандартную конфигурацию q_101^n0 в стандартную конфигурацию $q_001^{2n}0\dots0$. Приведем пример такой "Удваивающей" машины с 16 незаключительными состояниями.

Удвоение. Построим машину Тьюринга D , выполняющую преобразование

$$q_101^a0 \xRightarrow{D} q_001^a01^a0.$$

Построим машину Тьюринга D , выполняющую при $a > 0$ преобразование

$$q_101^a01^b01^c0 \xRightarrow{D} q_101^{a-1}01^{b+1}01^{c+1}0,$$

а при $a = 0$ – преобразование

$$q_1001^b01^c0 \xRightarrow{D} q_001^b01^c0.$$

$$\begin{array}{ll}
q_1 0 \longrightarrow q_2 0 R, & q_2 0 \longrightarrow q_3 1 S, \\
q_3 1 \longrightarrow q_3 1 R, & q_3 0 \longrightarrow q_4 0 R, \\
q_4 1 \longrightarrow q_4 1 R, & q_4 0 \longrightarrow q_5 0 L, \\
q_5 1 \longrightarrow q_6 0 L, & q_5 0 \longrightarrow q_7 0 L, \\
q_6 1 \longrightarrow q_6 1 L, & q_6 0 \longrightarrow q_7 1 L, \\
q_7 1 \longrightarrow q_8 0 L, & q_8 1 \longrightarrow q_8 1 L, \\
q_8 0 \longrightarrow q_0 0 S, & q_2 1 \longrightarrow q_9 1 S, \\
q_9 1 \longrightarrow q_9 1 R, & q_9 0 \longrightarrow q_{10} 0 L, \\
q_{10} 1 \longrightarrow q_{11} 0 R, & q_{11} 0 \longrightarrow q_{12} 1 S, \\
q_{12} 1 \longrightarrow q_{12} 1 R, & q_{12} 0 \longrightarrow q_{13} 0 R, \\
q_{13} 1 \longrightarrow q_{13} 1 R, & q_{13} 0 \longrightarrow q_{14} 1 S, \\
q_{14} 1 \longrightarrow q_{14} 1 L, & q_{14} 0 \longrightarrow q_{15} 0 L, \\
q_{15} 1 \longrightarrow q_{15} 1 L, & q_{15} 0 \longrightarrow q_{16} 0 L, \\
q_{16} 1 \longrightarrow q_{16} 1 L, & q_{16} 0 \longrightarrow q_1 0 S.
\end{array}$$

Если применить машину D к начальной конфигурации $q_1 0 1^a 0 = q_1 0 1^a 0 1^0 0 1^0$, то через конечное число шагов получим заключительную конфигурацию $q_0 0 1^a 0 1^a 0$.

Обозначим через P_n машину Тьюринга с n внутренними состояниями, которая печатает на пустой ленте подряд n единиц и останавливается в требуемой позиции, т. е. переводит стандартную конфигурацию $q_1 0$ в стандартную конфигурацию $q_0 0 1^n 0$. Приведем пример программы такой машины.

$$\begin{array}{l}
q_1 0 \longrightarrow q_2 1 R, \\
q_2 0 \longrightarrow q_3 1 R, \\
\ldots \quad \ldots \quad \ldots \\
q_{n-1} 0 \longrightarrow q_n 1 S, \\
q_n 1 \longrightarrow q_n 1 L, \\
q_n 0 \longrightarrow q_0 1 L.
\end{array}$$

Определим две операции над машинами Тьюринга, которые существенно облегчат нам построение новых машин Тьюринга по ранее построенным.

Напомним понятие **композиция машин Тьюринга**.

Пусть T_1 и T_2 – машины Тьюринга с одним и тем же внешним алфавитом \mathcal{A} и внутренними алфавитами $\mathcal{Q}_1 = \{q_0, q_1, \dots, q_m\}$ и $\mathcal{Q}_2 = \{q_0, q_1, \dots, q_p\}$ соответственно.

Тогда **композицией** машин Тьюринга T_1 и T_2 называется машина Тьюринга T с тем же внешним алфавитом \mathcal{A} , внутренним алфавитом $\mathcal{Q} = \{q_0, q_1, \dots, q_m, q_{m+1}, \dots, q_{m+p}\}$ и программой P_T , где

$$P_T = (P_{T_1})_{q_0}[q_{m+1}] \cup (P_{T_2})_{q_1, \dots, q_p}[q_{m+1}, \dots, q_{m+p}].$$

Через $(P_{T_2})_{q_1, \dots, q_p}[q_{m+1}, \dots, q_{m+p}]$ обозначен результат одновременной замены во всех командах программы P_{T_2} q_1 на q_{m+1} , q_2 на q_{m+2} , \dots , q_p на q_{m+p} , а через $(P_{T_1})_{q_0}[q_{m+1}]$ обозначен результат замены во всех командах программы P_{T_1} q_0 на q_{m+1} .

Построенную машину Тьюринга T , являющуюся *композицией* машин T_1 и T_2 , будем обозначать через $T_1 \cdot T_2$, а иногда и просто $T_1 T_2$.

Если машина Тьюринга T начинает работать в конфигурации $Zq_i W$, где Z и W – слова в ее входном алфавите \mathcal{A} , а $1 \leq i \leq m$, то она "сначала работает как машина Тьюринга T_1 " – если машина Тьюринга T_1 , начав работать в конфигурации $Zq_i W$, не остановится, то не остановится и машина Тьюринга T , если же машина Тьюринга T_1 , начав работать в конфигурации $Zq_i W$, остановится в конфигурации $Uq_0 V$, то дальше машина Тьюринга T "работает как машина Тьюринга T_2 , начав работать в конфигурации $Uq_1 V$ ". Более точно, если машина Тьюринга T_2 , начав работать в конфигурации $Uq_1 V$, не остановится, то не остановится и машина Тьюринга T , если же машина Тьюринга T_2 , начав работать в конфигурации $Uq_1 V$, остановится в конфигурации $Xq_0 Y$, то машина Тьюринга T остановится в этой же конфигурации. Если же $m+1 \leq i \leq m+p$, то машина Тьюринга T "работает как машина Тьюринга T_2 ".

Композиция $P_n \cdot D$ – это машина с $n + 16$ внутренними неактивными состояниями, переводящая конфигурацию $q_1 0$ в стандартную конфигурацию $q_0 01^{2n} 0$, поэтому

$$p(n + 16) \geq 2n.$$

Приступает к формулировке и доказательству основной теоремы этого параграфа.

Теорема 4.0.6. *Невозможно построить машину Тьюринга, алфавит которой состоит лишь из символов 0 и 1, правильно вычисляющую функцию $p(n)$.*

Доказательство. Предположим противное. Пусть машина Тьюринга ZB с k неактивными внутренними состояниями, алфавит которой состоит лишь из символов 0 и 1, правильно вычисляет функцию $p(n)$, т. е. для каждого натурального числа n переводит стандартную конфигурацию $q_1 01^n 0$ в стандартную конфигурацию $q_1 01^{p(n)} 0 \dots 0$.

Для произвольного натурального числа n композиция $P_n \cdot ZB \cdot ZB$ машин Тьюринга является машиной Тьюринга с $n + 2k$ внутренними незаключительными состояниями и с продуктивностью $p(p(n))$. Поэтому для любого натурального числа n выполняется неравенство

$$p(n + 2k) \geq p(p(n)).$$

Так как для любого натурального числа n выполняется неравенство $p(n + 1) > p(n)$, то $p(n)$ – строго возрастающая функция, т. е. неравенство $t < s$ влечет неравенство $p(t) < p(s)$. Значит из неравенства $p(n + 2k) \geq p(p(n))$ следует неравенство $n + 2k \geq p(n)$.

Последнее неравенство дает, очевидно, неравенство

$$n + 16 + 2k \geq p(n + 16).$$

Это вместе с установленным выше неравенством

$$p(n + 16) \geq 2n$$

дает неравенства

$$n + 16 + 2k \geq 2n, \quad 16 + 2k \geq n,$$

что невозможно, так как k фиксировано, а n – произвольное натуральное число.

Полученное противоречие завершает доказательство теоремы. □

Программа дисциплины
"Теория алгоритмов и сложность вычислений"

Введение. Алгоритмы в интуитивном смысле. Примеры алгоритмов из различных разделов математики: алгебры, теории чисел, математической логики, математического анализа, теории обыкновенных дифференциальных уравнений и т. д.

Основные свойства алгоритмов. Дискретность, детерминированность, элементарность шагов и массовость алгоритмов.

Необходимость математического уточнения интуитивного понятия алгоритма, примеры математических проблем, сформулированных в конце XIX – начале XX в.в., приведших к уточнению понятия алгоритма.

Неразрешимые алгоритмические пробелы в теории алгоритмов, алгебре, математической логике, теории чисел, математическом анализе, топологии.

Машина Тьюринга. Внешний и внутренний алфавиты, команды и программа машины Тьюринга. Различные варианты машин Тьюринга: многоленточные и одноленточные, с одномерной и многомерной лентой, с потенциально бесконечной в обе стороны лентой, с непродолжаемой влево лентой и т. д.

Словарные алгоритмы, реализуемые машинами Тьюринга.

Вычислимые по Тьюрингу функции. Правильная вычислимость по Тьюрингу. Вычислимость по Тьюрингу элементарных теоретико-числовых функций.

Разрешимые и перечислимые множества слов.

Операции над машинами Тьюринга. Композиция машин Тьюринга. Разветвление. Диаграммы машин Тьюринга.

Циклический сдвиг, копирование.

Тезис Тьюринга. Замкнутость класса правильно вычислимых по Тьюрингу функций относительно операций суперпозиции, примитивной рекурсии и минимизации.

Тезис А. Тьюринга.

Примитивно рекурсивные, частично рекурсивные и рекурсивные функции. Простейшие (исходные) функции. Операции суперпозиции, примитивной рекурсии и минимизации.

Примитивно рекурсивные функции. Примеры примитивно рекурсивных теоретико-числовых функций.

Частично рекурсивные и рекурсивные функции, примеры.

Операции над примитивными, рекурсивными и частично рекурсивными функциями.

Тезис А. Черча.

Нумерация пар и n -ок натуральных чисел. Нумерационные функции.
 Рекурсивные и рекурсивно перечислимые множества и предикаты.
 Теорема Э. Поста.
 Теорема о графике функции.
 Правильная вычислимость по Тьюрингу любой частично рекурсивной функции.
Арифметизация теории машин Тьюринга. Геделева нумерация слов в конечных и счетных алфавитах.
 Нумерация команд и программ машин Тьюринга. Нумерация конфигураций.
 Построение примитивно рекурсивных функций, описывающих работу машин Тьюринга.
 Частичная рекурсивность любой вычислимой по Тьюрингу функции.
 Универсальные частично рекурсивные функции.
 Неразрешимость проблем останова, самоприменимости и бессмертия для машин Тьюринга.
 Нормальная форма С. Клини.
 Универсальные машины Тьюринга.
Неразрешимые алгоритмические проблемы. Неразрешимость проблемы выводимости для полусистем Туэ.
 Неразрешимость проблемы равенства для полугрупп и групп.
 Теоремы А.А. Маркова и С.И. Адяна об алгоритмической неразрешимости проблем распознавания полугрупповых и групповых свойств.
 Неразрешимые проблемы в математической логике.
Сложность алгоритмов.
 Многоленточные машины Тьюринга: внешний и внутренний алфавиты, программы.
 Сложностные характеристики работы машины Тьюринга: временная (число шагов) и емкостная (объем памяти), связь между ними.
 Сложностные характеристики работы машины Тьюринга в худшем случае: временная и емкостная сигнализирующие функции (сложности, характеристики алгоритма), связь между ними. Сложностные классы.
 Другие сложностные характеристики.
Недетерминированные машины Тьюринга.
 Недетерминированные многоленточные машины Тьюринга: внешний и внутренний алфавиты, программы.
 Классы \mathcal{P} и \mathcal{NP} .
 \mathcal{NP} -трудные и \mathcal{NP} -полные задачи.
 Теорема Кука об \mathcal{NP} -полноте проблемы выполнимости для логики высказываний.

Примеры \mathcal{NP} -полных проблем из различных разделов математики: дискретной математики, теории булевых функций, математической логики, теории графов, алгебры, теории чисел, теории автоматов и языков и т. д.

Трудно разрешимые задачи. Нижние оценки. Задачи, требующие экспоненциального времени и памяти (из теории автоматов и языков, из теории разрешимых элементарных теорий, например, теорема Рабина – Фишера о сложности разрешения арифметики Пресбургера).

Неэлементарные задачи. Определение класса элементарных функций. Задачи, требующие неэлементарного времени для решения: из теории регулярных выражений, из математической логики (неэлементарность элементарной сингулярной (слабой) теории функции следования, неэлементарность элементарной теории свободной группы и т. д.)

Нормальные алгоритмы А.А. Маркова. Схема нормального алгоритма А.А. Маркова. Простые и заключительные правила подстановки. Выполнение нормального алгоритма. Примеры нормальных алгоритмов.

Вычислимые по А.А. Маркову функции. Принцип нормализации.

Сочетание нормальных алгоритмов.

Универсальный алгоритм. Основные теоремы о невозможности алгоритмов.

Сложность описания алгоритма. Длина записи нормального алгоритма. Использование сложности описания (задания) алгоритма для доказательства существования алгоритмически неразрешимых проблем.

Диофантовы множества и функции. 10-ая проблема Д. Гильберта.

Теорема М. Девиса – Х. Путнам – Дж. Робинсон – Ю.В. Матиясевича.

Неразрешимые алгоритмические проблемы для систем линейных диофантовых уравнений.

Неразрешимые алгоритмические проблемы для систем уравнений в свободных группах и полугруппах.

Неразрешимые алгоритмические проблемы в топологии, математическом анализе, теории дифференциальных уравнений.

Значение существования алгоритмически неразрешимых проблем и трудно разрешимых проблем для математики и ее приложений.

Список литературы

- [1] Адян С.И. Алгоритмическая неразрешимость проблем распознавания некоторых свойств групп // Докл. АН СССР. 1955. Т. 103. N 4. С.533-535.
- [2] Адян С.И., Дурнев В.Г. *Алгоритмические проблемы для групп и полугрупп* // Успехи матем. наук. 2000. Том 55. № 2. С. 3 – 94.
- [3] Ахо А., Хопкрофт Дж., Ульман Дж. *Построение и анализ вычислительных алгоритмов*. М.: Мир, 1983.
- [4] Булос Дж., Джеффри Р. *Вычислимость и логика*. М.: Мир, 1994.
- [5] Гэри М., Джонсон Д. *Вычислительные машины и труднорешаемые задачи*. М.: Мир, 1979.
- [6] Дурнев В.Г. *Элементы теории алгоритмов*. Ярославль: ЯрГУ, 2008.
- [7] Дурнев В.Г. О позитивных формулах на свободных полугруппах // Сиб. матем. журн. 1974. Т. 25. N 5. С.1131-1137.
- [8] Дурнев В.Г. Неразрешимость позитивной $\forall\exists^3$ -теории свободной полугруппы // Сиб. матем. журн. 1995. Т. 36. N 5. С.1067-1080.
- [9] Дурнев В.Г. Об уравнениях на свободных полугруппах и группах // Матем. заметки. 1974. Т. 16. N 5. С.717-724.
- [10] Катленд Н. *Вычислимость. Введение в теорию рекурсивных функций*. М.: Мир, 1983.
- [11] Колмогоров А.Н., Успенский В.А. К определению понятия алгоритма // Успехи мат. наук. 1958. Т. 13. Вып. 4. С.3-28.
- [12] Маканин Г.С. К проблеме тождества в конечно-определенных полугруппах // Докл. АН СССР. 1966. Т. 171. N 2. С.285-287.
- [13] Маканин Г.С. Проблема разрешимости уравнений в свободной полугруппе // Мат. сб. 1977. Т. 103(145). N 2(6). С.147-236.
- [14] Мальцев А.И. *Алгоритмы и рекурсивные функции*. М.: Наука, 1986.
- [15] Манин Ю.И. *Вычислимое и невычислимое*. М.: Советское радио, 1979.

- [16] Марков А.А. *Невозможность некоторых алгоритмов в теории ассоциативных систем* // ДАН СССР. 1947. Том 55. № 7. С. 587 – 590.
- [17] Марков А.А., Нагорный Н.М. *Теория алгоритмов*. М.: Наука, 1984.
- [18] Марков А.А. Неразрешимость проблемы гомеоморфии // Докл. АН СССР. 1958. Т. 121. N 2. С.218-220.
- [19] Марков А.А. К проблеме представимости матриц // Z. Math. Log. und Grundle. Math. 1958. Т. 4. N 2. С.157-168.
- [20] Марченков С.С. Неразрешимость позитивной $\forall\exists$ -теории свободной полугруппы // Сиб. мат. журн. 1982. Т. 32. N 1. С.196-198.
- [21] Матиясевич Ю.В. Простые примеры неразрешимых ассоциативных исчислений // Докл. АН СССР. 1967. Т. 173. N 6. С.1264-1266.
- [22] Матиясевич Ю.В. Диофантовость перечислимых множеств // Докл. АН СССР. 1970. Т. 130. N 3. С.495-498.
- [23] Мендельсон Э. *Введение в математическую логику*. М.: Наука, 1976.
- [24] Новиков П.С. Об алгоритмической неразрешимости проблемы тождества теории групп // Докл. АН СССР. 1952. Т. 85. N 4. С.709-712.
- [25] Семенов А.Л. *Интерпретация свободных алгебр в свободных группах* // ДАН СССР. 1980. Том 252. № 6. С. 1326 – 1332.
- [26] Трахтенброт Б.А. *Алгоритмы и вычислительные автоматы*. М.: Советское радио, 1974.
- [27] Цейтин Г.С. Относительно проблемы распознавания свойств ассоциативных исчислений // Докл. АН СССР. 1956. Т. 107. N 2. С.209-212.
- [28] Цейтин Г.С. Ассоциативное исчисление с неразрешимой проблемой эквивалентности // Труды матем. ин-та. АН СССР. 1958. Т. 52. С.172-189.
- [29] Эббинхауз Г.Д., Якобс К., Ман Ф.К., Хермес Г. *Машины Тьюринга и рекурсивные функции*. М.: Мир, 1972.
- [30] Church A. *An unsolvable problem of elementary number theory* // Amer. J. Math. 1936. Vol. 58. № 2. P. 345 – 363.

- [31] Churh A. *A note on the Entscheidungsproblem* // J. Symbolic Logic. 1936. Vol. 1. № 1. P. 40 – 41.
- [32] Dehn M. *Über unendliche diskontinuerliche Gruppen* // Math. Ann. 1911. Bd. 71. S.116-144.
- [33] Post, E. *Intoduction to a general theory of elementary propositions* / E. Post // Amer. J. Math. 1921. Vol. 43.
- [34] Post E.L. *Finite combinatory processes - formulation 1* // Journal of Symbolic Logic. 1936. Vol. 1. № 3. P. 103 – 105.
- [35] Post E.L. *A variant of a recursively unsolvable problem* // Bull. Amer. Math. Soc. 1946. Vol. 52. P. 264 – 268.
- [36] Post E.L. *Recursive unsolvability of a problem of Thue* // J. Symbol Log. 1947. Vol. 12. № 1. P. 1 – 11.
- [37] Rado T. *On non-computable functions* // Bell System Technical Journal. 1962. № 41. P. 877 – 884.
- [38] Quine W. *Concatenation as a basis for arithmetic* // J. Symbol Log. 1946. Vol. 11. P. 105 – 114.
- [39] Hall M.Jr. *The word problem for semi-groups with two generators* // J.Symbolic Logic, 1949. V.14. P.115-118.
- [40] A. Thue. *Problem üder Veränderungen von Zeichenreihen nach gegebenen Regeln* // Vid. Skr. Math.-natur. Kl. 1914. № 10.
- [41] Tietze H. *Über topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten* // Monatsh. Math. Phys. 1908. Vol. 19. P. 1 – 118.
- [42] Turing A.M. *On computable numbers, with an application to the Entscheidungsproblem* // Proceedings of London Mathematical Society. Ser. 2. 1936. Vol. 42. № 3, 4. P. 230 – 265.
- [43] Scott D. *A short recursively unsolvable problem (abstract)* // J. Symbol. Log. 1956. Vol. 21. N 1. P.11-112.

МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ “ТЕОРИЯ АЛГОРИТМОВ И СЛОЖНОСТЬ ВЫЧИСЛЕНИЙ”

Составитель: **Дурнев** Валерий Георгиевич

Корректор И.В. Бунакова
Компьютерный набор, верстка В.Г. Дурнева, М.А. Башкина

Подписано в печать 10.12.2010. Формат $60 \times 84^1/_{16}$.
Бумага тип. Усл. печ. л. 2,32. Уч.-изд. л. 2,2.
Тираж 100 экз. Заказ .

Оригинал-макет подготовлен в редакционно-издательском отделе
Ярославского государственного университета

Отпечатано на ризографе

Ярославский государственный университет
им. П.Г. Демидова.
150000 Ярославль, ул. Советская, 14