

Неразрешимые алгоритмические проблемы

1. Введение

Определение. *Алгоритмическая проблема* — это проблема, в которой требуется найти единый метод (алгоритм) для решения бесконечной серии однотипных единичных задач.

Такие проблемы называют так же *массовыми проблемами*. Они возникали и решались в различных областях математики на протяжении всей ее истории.

В качестве понятия, уточняющего понятие алгоритма, будем использовать понятие машины Тьюринга. Докажем, что существуют функции невычислимые по Тьюрингу.

2. Нумерация машин Тьюринга

Покажем, что множество всех машин Тьюринга, различающихся по существу своей работы, счетно. Для этого перенумеруем все машины Тьюринга с помощью специальных шифров.

2.1. Коды символов

Будем считать, что для обозначения внутренних состояний машин Тьюринга используются буквы бесконечной последовательности: $q_0, q_1, q_2, \dots, q_i, \dots$. Для обозначения букв внешних алфавитов используются буквы последовательности: $a_0, a_1, a_2, \dots, a_i, \dots$. Для обозначения перемещений управляющей головки используются символы Л, Н, П.

Закодируем все эти символы словами алфавита $\{0, 1\}$ по следующим правилам.

№	Символ	Код	Кол-во «0»
1	Л	101	1
2	Н	1001	2
3	П	10001	3
4	a_0	100001	4
5	q_0	1000001	5
6	a_1	10000001	6
7	q_1	100000001	7
8	a_2	1000000001	8
9	q_2	10000000001	9
...	...	...	...

Таблица 1. Коды символов

2.2. Шифры команд

В записях команд $a_i q_k \rightarrow a_j q_\ell X$, где $X \in \{\text{Л}, \text{Н}, \text{П}\}$, опускается символ « $\rightarrow$ », а буквы a_i, q_k, a_j, q_ℓ, X заменяются соответствующими кодами таблицы 1.

Пример. Для команды $a_0 q_1 \rightarrow a_1 q_0 \text{Н}$ получим следующий шифр:

$$\underbrace{100001}_{a_0} \underbrace{100000001}_{q_1} \underbrace{10000001}_{a_1} \underbrace{1000001}_{q_0} \underbrace{1001}_{\text{Н}}$$

2.3. Шифры программ

Программы машин Тьюринга мы записывали в виде прямоугольных таблиц.

Пример. Программа вычисляющая функцию $f_{+1}(x) = x + 1$ в кодах задается следующей таблицей:

$A \setminus Q$	q_1
0	$1q_0H$
1	$1q_1\Pi$

Таблица 2. Программа вычисляющая $x + 1$ в кодах

Для кодирования программы машины Тьюринга вначале запишем эту программу в строчку, перебирая команды из таблицы слева направо и сверху вниз. А затем заменим каждую команду ее шифром. Полученную строчку из ноликов и единиц, составленную для программы машины Тьюринга, называют *шифром программы*.

Пример. Запишем шифр программы, вычисляющей функцию $f_{+1}(x) = x + 1$ в кодах. Сама программа задается таблицей 2. Полагая $a_0 = 0$, $a_1 = 1$ получим следующий шифр:

$$\overbrace{100001}^{a_0 q_1 \rightarrow a_1 q_0 H} \overbrace{100000001}^{a_1 q_1 \rightarrow a_1 q_1 \Pi} \overbrace{10000001}^{a_1} \overbrace{1000001}^{q_0} \overbrace{1001}^{H} \overbrace{10000001}^{a_1} \overbrace{100000001}^{q_1} \overbrace{10000001}^{a_1} \overbrace{100000001}^{q_1} \overbrace{10001}^{\Pi}$$

Лемма. Множество всех шифров программ счетно.

Доказательство. Шифр каждой программы можно рассматривать как натуральное число, записанное в десятичной системе исчисления. Поэтому множество всех шифров является подмножеством множества натуральных чисел. Значит множество всех шифров не более чем счетно (т. е. или конечно или счетно).

Кроме того понятно, что множество машин Тьюринга бесконечно. Действительно, рассмотрим например машины Тьюринга, вычисляющие функции $f_0(x) \equiv 0$, $f_1(x) \equiv 1$, $f_2(x) \equiv 2$, ... в кодах. Множество этих машин Тьюринга бесконечно. И бесконечно множество их шифров. Значит множество всех шифров программ бесконечно.

Поэтому множество всех шифров программ является бесконечным подмножеством множества натуральных чисел. Значит это множество счетно. $\square$

3. Невычислимые функции

Как и ранее, будем рассматривать функции, аргументы и значения которых принадлежат расширенному натуральному ряду $\mathbb{N} = \{0, 1, 2, \dots\}$ (множество, содержащее натуральные числа и нуль).

Определение. Функция называется *вычислимой по Тьюрингу*, если существует машина Тьюринга, вычисляющая эту функцию.

Замечание. Поскольку множество всех шифров программ счетно, то и всех мыслимых машин Тьюринга имеется счетное количество. Тогда и множество функций, вычислимых по Тьюрингу, так же счетно.

Докажем, что существуют функции невычислимые по Тьюрингу.

Пусть $\mathcal{F}$ — множество всех функций $f: \mathbb{N} \mapsto \mathbb{N}$. Докажем, что множество $\mathcal{F}$ несчетно.

Лемма. Множество $\mathcal{F}$ несчетно.

Доказательство проведем методом от противного. Предположим, что множество $\mathcal{F}$ счетно. Тогда его можно представить в виде $\mathcal{F} = \{f_0, f_1, f_2, \dots\}$. Определим функцию $g: \mathbb{N} \mapsto \mathbb{N}$ следующим образом:

$$g(i) = \begin{cases} 1, & \text{при } f_i(i) = 0, \\ 0, & \text{при } f_i(i) \neq 0. \end{cases}$$

Тогда для $i = 0$ выполнено $g(0) \neq f_0(0)$, и значит $g \neq f_0$. Для $i = 1$ выполнено $g(1) \neq f_1(1)$, и значит $g \neq f_1$. Таким образом, при любом $i \in \mathbb{N}$ выполнено $g(i) \neq f_i(i)$, и значит $g \neq f_i$. Получили, что $g \notin \mathcal{F}$. Но это противоречит определению $\mathcal{F}$ (ведь $\mathcal{F}$ — множество *всех* функций $f: \mathbb{N} \mapsto \mathbb{N}$). $\square$

Теорема. *Существуют функции невычислимые по Тьюрингу.*

Доказательство. Множество функций вычислимых по Тьюрингу счетно. А так как множество всех функций несчетно, то значит существуют функции, невычислимые по Тьюрингу. $\square$

4. Неразрешимые алгоритмические проблемы

Приведем примеры неразрешимых алгоритмических проблем.

4.1. 10-ая проблема Гильберта

10-ая проблема Гильберта алгоритмически неразрешима.

4.2. Проблема распознавания самоприменимости

Определение. Предположим, что на ленте машины Тьюринга записан шифр ее собственной программы. Если машина применима к своему шифру, то будем называть ее *самоприменимой*, в противном случае — *несамоприменимой*.

Проблема распознавания самоприменимости состоит в следующем. По заданному шифру программы нужно установить, является ли соответствующая машина Тьюринга самоприменимой или нет.

Теорема. *Проблема распознавания самоприменимости алгоритмически неразрешима.*

Доказательство. Предположим противное. Пусть такая машина A существует. Будем считать, что A всякий самоприменимый шифр перерабатывает в символ «С» (имеющий смысл утвердительного ответа на поставленный вопрос о самоприменимости), а всякий несамоприменимый шифр — в символ «Н» (имеющий смысл отрицательного ответа на поставленный вопрос), и останавливается над этими символами.

В таком случае можно построить и такую машину B , которая по-прежнему перерабатывает несамоприменимые шифры в «Н», в то время как к самоприменимым шифрам B уже не применима. Этого можно добиться путем такого изменения программы машины A , чтобы вместо выключения над символом «С», машина начала бы неограниченно повторять этот же символ (для этого достаточно исправить команды выключения машины A). Таким образом, полученная машина B применима ко всякому несамоприменимому шифру (останавливается над символом «Н») и не применима к самоприменимым шифрам (неограниченно перезаписывает символ «С»). Но это приводит к противоречию.

1. Предположим, что машина B самоприменима. Тогда (по построению), при применении к своему собственному шифру машина B будет неограниченно перезаписывать символ «С». Но это означает что B несамоприменима.

2. Пусть машина B несамоприменима. Тогда (по построению), при применении к своему собственному шифру машина B остановится над символом «Н». Но это означает, что B самоприменима.

Полученное противоречие доказывает теорему. $\square$

4.3. Проблема распознавания применимости

Проблема распознавания применимости состоит в следующем. Заданы программа машины Тьюринга и начальная конфигурация к ней. Нужно установить, применима ли машина Тьюринга к данной конфигурации или нет.

Теорема. *Проблема распознавания применимости алгоритмически неразрешима.*

Доказательство. Если бы существовала машина Тьюринга для решения этой проблемы, то с ее помощью можно было бы решить и проблему распознавания самоприменимости. А так как проблема распознавания самоприменимости алгоритмически неразрешима, то алгоритмически неразрешима и проблема распознавания применимости. $\square$

Список литературы

1. Лихтарников Л.М., Сукачева Т.Г. Математическая логика. Курс лекций. Задачник-практикум и решения. 4-е изд. СПб.: Издательство «Лань», 2009. 288 с.
2. Лихтарников Л.М., Сукачева Т.Г. Математическая логика. СПб.: Издательство «Лань», 1998. 288 с.
3. Игошин В.И. Математическая логика и теория алгоритмов. М.: Издательский центр «Академия», 2004. 448 с.
4. Гладкий А.В. Математическая логика. М.: Российск. гос. гуманит. ун-т, 1998. 479 с.