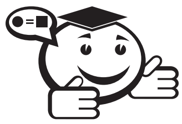
Модуль 7.1. Неформальная вычислимость и машины Тьюринга

Под ***алгоритмом*** понимается способ преобразования представления информации. Слово «*algorithm*» — произошло от имени аль-Хорезми — автора известного арабского учебника по математике (от его имени произошли также слова «алгебра» и «логарифм»).

Интуитивно говоря, алгоритм — некоторое формальное предписание, действуя согласно которому, можно получить решение задачи.

Алгоритмы типичным образом решают не только частные задачи, но и классы задач. Подлежащие решению частные задачи, выделяемые по мере надобности из рассматриваемого класса, определяются с помощью параметров. Параметры играют роль исходных данных для алгоритма.

Основные особенности алгоритма



Определенность. *Алгоритм разбивается на отдельные шаги (этапы), каждый из которых должен быть простым и локальным.*

Ввод. *Алгоритм имеет некоторое (быть может, равное нулю) число входных данных, т. е. величин, заданных ему до начала работы.*

Вывод. *Алгоритм имеет одну или несколько выходных величин, т. е. величин, имеющих вполне определенное отношение к входным данным.*

Детерминированность. *После выполнения очередного шага алгоритма однозначно определено, что делать на следующем шаге.*

Обратите внимание, что мы не требуем, чтобы алгоритм заканчивал свою работу для любых входных данных.

Примеры алгоритмов широко известны: изучаемые в школе правила сложения и умножения десятичных чисел или, скажем, алгоритмы сортировки массивов. Для алгоритмически разрешимой задачи всегда имеется много различных способов ее решения, т. е. различных алгоритмов.

Данное здесь определение алгоритма не является, конечно, строгим, но оно интуитивно кажется вполне определенным. К сожалению, для решения некоторых задач не существует алгоритма. Установление таких фактов требует введения строгого понятия алгоритма.

Мы будем рассматривать алгоритмы, имеющие дело только с натуральными числами. Можно доказать, что это не является потерей общности, так как объекты другой природы можно закодировать натуральными числами. Для пользователей компьютеров такое утверждение должно быть очевидным.

*Пусть **N** обозначает множество натуральных чисел* $\{0, 1, 2, \ldots\}$. *Объекты, которые мы будем рассматривать, будут функциями с областью определения* $D_f \subseteq \boldsymbol{N}^k$ *(k — целое положительное число) и с областью значений* $R_f \subseteq \boldsymbol{N}$. *Такие функции будем называть **k-местными частичными**. Слово «частичная» должно напомнить о том, что функция определена на подмножестве* $\boldsymbol{N}^k$ *(конечно, в частном случае может быть* $D_f = \boldsymbol{N}^k$*, тогда функция называется **всюду определенной**).*

Назовем k-местную функцию $f\colon \boldsymbol{N}^k \to \boldsymbol{N}$ ***вычислимой**, если существует алгоритм A, её вычисляющий, т. е. такой алгоритм A, что:*

1. *Если на вход алгоритма A поступил вектор* $\boldsymbol{x} = <x_1, x_2, \ldots, x_k>$ *из* D_f*, то вычисление должно закончиться после конечного числа шагов и выдать* $f(\boldsymbol{x})$.
2. *Если на вход алгоритма A поступил вектор* $\boldsymbol{x}$*, не принадлежащий области определения* D_f*, то алгоритм A никогда не заканчивается.*

Несколько замечаний по поводу этого определения.

1. Понятие вычислимости определяется здесь для частичных функций (областью определения которых является некоторое подмножество натурального ряда). Например, нигде не определенная функция вычислима, в качестве A надо взять программу, которая всегда зацикливается.
2. Можно было бы изменить определение, сказав так: «если $f(\boldsymbol{x})$ не определено, то либо алгоритм A не останавливается, либо останавливается, но ничего не печатает на выходе». На самом деле от этого ничего бы не изменилось (вместо того, чтобы останавливаться, ничего не напечатав, алгоритм может зацикливаться).
3. Входами и выходами алгоритмов могут быть не только натуральные числа, но и двоичные строки (слова в алфавите $\{0, 1\}$), конечные последовательности слов и вообще любые, как говорят, «конструктивные объекты».
4. Множество вычисляемых функций мы не отождествляем с множеством «практически вычисляемых» функций, так как не накладываем на первое множество никаких ограничений, связанных с современными вычислительными машинами. Хотя каждое входное натуральное число должно быть конечным, тем не менее не предполагается верхняя граница размера этого числа, так, например, количество цифр числа может быть больше числа электронов во Вселенной. Точно так же нет никакой верхней границы на число шагов, которые может сделать алгоритм для конкретных $\boldsymbol{x}$ из области определения.

Рассматривая теорию алгоритмов, мы можем ссылаться на программистский опыт, говоря об алгоритмах, программах, интерпретаторах и т. д. Это позволяет нам игнорировать детали построения тех или иных алгоритмов под тем предлогом, что читатель их легко восстановит (или хотя бы поверит). Но в некоторых случаях этого недостаточно, поэтому мы собираемся дать строгое определение нового множества функций, которое в некотором смысле будет совпадать с множеством вычислимых функций. Мы дадим две формализации понятия вычислимой функции.

Машины Тьюринга

Рассмотрим еще один способ определения вычислимых функций, следуя в изложении [1, с. 12–14]. Формулировка, выраженная в терминах воображаемой вычислительной машины, была дана английским математиком Аланом Тьюрингом в 1936 г. Главная трудность при нахождении этого определения была в том, что Тьюринг искал его до создания реальных цифровых вычислительных машин. Познание шло от абстрактного к конкретному: фон Нейман был знаком с работой Тьюринга, и сам Тьюринг позднее сыграл вдохновляющую роль в развитии вычислительных машин.

На неформальном уровне мы можем описывать машину Тьюринга как некий черный ящик с лентой. Лента разбита на ячейки, и каждая ячейка может содержать пустой символ 0 либо непустой символ 1. Лента потенциально бесконечна в обе стороны в том смысле, что мы никогда не придем к ее концу, но в любое время лишь конечное число ячеек может быть непустым. В начале лента содержит числа входа, в конце — число-выход. В промежуточное время лента используется как пространство памяти для вычисления.

Если мы откроем черный ящик, то обнаружим, что он устроен очень просто. В любой момент времени он может обозревать лишь одну ячейку памяти. Устройство содержит конечный список инструкций (или *состояний*) q_0, $q_1, \ldots, q_n$. Каждая инструкция может указать два возможных направления действий; одного нужно придерживаться, если на обозреваемой ячейке ленты находится 0, а другого — если там находится 1. В любом случае следующее действие может состоять из таких трех типов элементарных шагов:

- символ (возможно, такой же, как старый) пишется на обозреваемой ячейке ленты, при этом предыдущий символ стирается;
- лента сдвигается на одну ячейку влево или вправо;
- указывается следующая инструкция.

Таким образом, список инструкций определяет некоторую функцию перехода, которая по данной инструкции и обозреваемому символу указывает три компоненты того, что нужно делать. Мы можем формализовать эти идеи, взяв в качестве машины Тьюринга эту функцию перехода.

Машина Тьюринга — *это функция* M*, такая, что для некоторого натурального числа* n *область определения этой функции есть подмножество множества* $\{0, 1, \ldots, n\} \times \{0, 1\}$*, а область значений есть подмножество множества* $\{0, 1\} \times \{Л, П\} \times \{0, 1, \ldots, n\}$.

Например, пусть $M(3,1) = <0, \text{Л}, 2>$. Подразумеваемый смысл этого состоит в том, что как только машина дойдет до инструкции q_3, а на обозреваемой ячейке написан символ 1, она должна стереть 1 (оставляя на ячейке 0), передвинуть ленту так, чтобы обозреваемой ячейкой стала левая соседняя ячейка от той, которая обозревалась, и перейти к следующей инструкции q_2. Если $M(3,1)$ не определено, тогда как только машина дойдет до инструкции q_3, а на обозреваемой ячейке написан символ 1, то машина останавливается. (Это единственный путь остановки вычисления.)

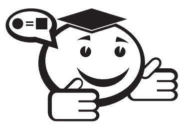
Такая подразумеваемая интерпретация не включена в формальное определение машины Тьюринга, но она мотивирует и подсказывает формулировки всех следующих определений. В частности, можно определить, что означает для машины M передвижение (за один шаг) от одной конфигурации до другой. Нам не нужно здесь давать формальных определений, так как они являются простыми переводами наших неформальных идей.

Входные и выходные данные — это строчки из 1, разделенные 0. Пусть $<n>$ будет строчкой из 1 длины $n+1$. Тогда

$$<n_1>\ 0\ <n_2>\ 0 \ldots 0\ <n_k>$$

получено комбинацией k строчек из 1, каждая отделена от другой 0.

Наконец, мы можем определить вычислимость.



Пусть $D_f \subseteq \boldsymbol{N}^k$ — область определения k-местной функции f: $D_f \to \boldsymbol{N}$. Функция f называется ***вычислимой по Тьюрингу****, если существует машина Тьюринга M, такая, что как только M начинает с инструкции q_0, обозревая самой левый символ строки*

$$<n_1>\ 0\ <n_2>\ 0 \ldots 0\ <n_k>,$$

(вся остальная часть ленты пуста), тогда:

- *если $f(n_1, n_2, \ldots, n_k)$ определено, то M, в конце концов, остановится, обозревая самый левый символ строки $<f(n_1, n_2, \ldots, n_k)>$, при этом часть, находящаяся справа от этой строчки, пустая;*
- *если $f(n_1, n_2, \ldots, n_k)$ не определено, то M никогда не останавливается.*

Заметим, что имеется бесконечное множество машин Тьюринга, для каждой вычислимой функции своя. Более того, для любой вычислимой функции имеется бесконечное множество машин Тьюринга, вычисляющих эту функцию.

Пример 1

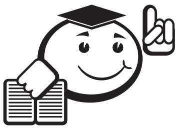
Построим машину Тьюринга, вычисляющую сумму n_1+n_2. Зададим функцию M следующим образом:

$M(0, 1)$ = <1, П, 0>;
$M(0, 0)$ = <1, П, 1>;
$M(1, 1)$ = <1, П, 1>;
$M(1, 0)$ = <0, Л, 2>;
$M(2, 1)$ = <0, Л, 3>;
$M(3, 1)$ = <0, Л, 4>;
$M(4, 1)$ = <1, Л. 4>;
$M(4, 0)$ = <0, П, 5>.

Посмотрим, как происходит сложение 1 + 1. В текущей строке символов обозреваемый символ выделен.

Номер инструкции	**Текущая строка символов**	**Комментарий**
0	0***1***10110	Прохождение через первое слагаемое
0	01***1***0110	
0	011***0***110	Заполнение промежутка
1	0111***1***10	Прохождение через второе слагаемое
1	01111***1***0	
1	011111***0***	Конец второго слагаемого
2	01111***1***0	Стирание 1
3	0111***1***00	Стирание второй 1
4	011***1***000	Движение назад
4	01***1***1000	
4	0***1***11000	
4	***0***111000	Остановка
5	0***1***11000	

Мы должны заметить, что многие детали нашего определения машины Тьюринга до некоторой степени произвольны. Если бы было более одной ленты, то класс вычислимых функций остался бы неизменным, хотя некоторые функции могли бы быть вычислены более быстро. Аналогично, мы могли бы допускать больше символов, чем 0 и 1, или же у нас могла бы быть лента, бесконечная только в одну сторону от начальной точки, вместо имеющейся бесконечной в обоих направлениях. Ни одно из этих изменений не затрагивает класса вычислимых функций. Что действительно существенно в этом определении — это разрешение произвольно большого количества материала для запоминающего устройства и произвольно длинных вычислений.



Список литературы по модулю

[1] Справочная книга по математической логике : в 4 ч. : пер. с англ. / под ред. Дж. Барвайса. — М. : Наука, 1982. – Ч. 3: Теория рекурсии. – 360 с.