

Алгоритмически неразрешимые проблемы

Алан Тьюринг высказал предположение (известное как Тезис Чёрча — Тьюринга), что любой алгоритм в интуитивном смысле этого слова может быть представлен эквивалентной машиной Тьюринга. Уточнение представления о вычислимости на основе понятия машины Тьюринга (и других эквивалентных ей понятий) открыло возможности для строгого доказательства алгоритмической неразрешимости различных массовых проблем (то есть проблем о нахождении единого метода решения некоторого класса задач, условия которых могут варьироваться в известных пределах). Простейшим примером алгоритмически неразрешимой массовой проблемы является так называемая проблема применимости алгоритма (называемая также проблемой остановки). Она состоит в следующем: требуется найти общий метод, который позволял бы для произвольной машины Тьюринга (заданной посредством своей программы) и произвольного начального состояния ленты этой машины определить, завершится ли работа машины за конечное число шагов, или же будет продолжаться неограниченно долго.

В течение первого десятилетия истории теории алгоритмов неразрешимые массовые проблемы были обнаружены лишь внутри самой этой теории (сюда относится описанная выше проблема применимости), а также внутри математической логики (проблема выводимости в классическом исчислении предикатов). Поэтому считалось, что теория алгоритмов представляет собой обочину математики, не имеющую значения для таких её классических разделов, как алгебра или анализ. Положение изменилось после того, как А. А. Марков и Э. Л. Пост в 1947 году установили алгоритмическую неразрешимость известной в алгебре проблемы равенства для конечнопорождённых и конечноопределённых подгрупп (т. н. проблемы Туэ). Впоследствии была установлена алгоритмическая неразрешимость и многих других «чисто математических» массовых проблем. Одним из наиболее известных результатов в этой области является доказанная Ю. В. Матиясевичем алгоритмическая неразрешимость десятой проблемы Гильберта.

2. Алгоритмически неразрешимые проблемы

За время своего существования человечество придумало множество алгоритмов для решения разнообразных практических и научных проблем. Зададимся вопросом — а существуют ли какие-нибудь проблемы, для которых невозможно придумать алгоритмы их решения?

Утверждение о существовании алгоритмически неразрешимых проблем является весьма сильным — мы констатируем, что мы не только сейчас не знаем соответствующего алгоритма, но мы не можем принципиально никогда его найти.

Успехи математики к концу XIX века привели к формированию мнения, которое выразил Д. Гильберт — «в математике не может быть неразрешимых проблем», в связи с этим формулировка проблем Гильбертом на конгрессе 1900 года в Париже была руководством к действию, констатацией отсутствия решений в данный момент.

Первой фундаментальной теоретической работой, связанной с доказательством алгоритмической неразрешимости, была работа Курта Гёделя — его известная теорема о неполноте символических логик. Это была строго сформулированная математическая проблема, для которой не существует решающего её алгоритма. Усилиями различных исследователей список алгоритмически неразрешимых проблем был значительно расширен. Сегодня принято при доказательстве алгоритмической неразрешимости некоторой задачи сводить её к ставшей классической задаче — «задаче останова».

Имеет место быть следующая теорема:

Теорема 3.1. Не существует алгоритма (машины Тьюринга), позволяющего по описанию произвольного алгоритма и его исходных данных (и алгоритм и данные заданы

символами на ленте машины Тьюринга) определить, останавливается ли этот алгоритм на этих данных или работает бесконечно.

Таким образом, фундаментально алгоритмическая неразрешимость связана с бесконечностью выполняемых алгоритмом действий, т.е. невозможностью предсказать, что для любых исходных данных решение будет получено за конечное количество шагов.

Тем не менее, можно попытаться сформулировать причины, ведущие к алгоритмической неразрешимости, эти причины достаточно условны, так как все они сводимы к проблеме останова, однако такой подход позволяет более глубоко понять природу алгоритмической неразрешимости:

а) Отсутствие общего метода решения задачи

Проблема 1 : Распределение девяток в записи числа π ;

Определим функцию $f(n) = i$, где n – количество девяток подряд в десятичной записи числа π , а i – номер самой левой девятки из n девяток подряд: $\pi=3,141592\dots f(1) = 5$.

Задача состоит в вычислении функции $f(n)$ для произвольно заданного n .

Поскольку число π является иррациональным и трансцендентным, то мы не знаем никакой информации о распределении девяток (равно как и любых других цифр) в десятичной записи числа π . Вычисление $f(n)$ связано с вычислением последующих цифр в разложении π , до тех пор, пока мы не обнаружим n девяток подряд, однако у нас нет общего метода вычисления $f(n)$, поэтому для некоторых n вычисления могут продолжаться бесконечно – мы даже не знаем в принципе (по природе числа π) существует ли решение для всех n .

Проблема 2: Вычисление совершенных чисел;

Совершенные числа – это числа, которые равны сумме своих делителей, например: $28 = 1+2+4+7+14$.

Определим функцию $S(n) = n$ -ое по счёту совершенное число и поставим задачу вычисления $S(n)$ по произвольно заданному n . Нет общего метода вычисления совершенных чисел, мы даже не знаем, множество совершенных чисел конечно или счетно, поэтому наш алгоритм должен перебирать все числа подряд, проверяя их на совершенность. Отсутствие общего метода решения не позволяет ответить на вопрос о останове алгоритма. Если мы проверили M чисел при поиске n -ого совершенного числа – означает ли это, что его вообще не существует?

Проблема 3: Десятая проблема Гильберта;

Пусть задан многочлен n -ой степени с целыми коэффициентами – P , существует ли алгоритм, который определяет, имеет ли уравнение $P=0$ решение в целых числах?

Ю.В. Матиясевич показал, что такого алгоритма не существует, т.е. отсутствует общий метод определения целых корней уравнения $P=0$ по его целочисленным коэффициентам.

б) Информационная неопределенность задачи

Проблема 4: Позиционирование машины Поста на последний помеченный ящик;

Пусть на ленте машины Поста заданы наборы помеченных ящиков (кортежи) произвольной длины с произвольными расстояниями между кортежами и головка находится у самого левого помеченного ящика. Задача состоит установке головки на самый правый помеченный ящик последнего кортежа.

Попытка построения алгоритма, решающего эту задачу приводит к необходимости ответа на вопрос – когда после обнаружения конца кортежа мы сдвинулись вправо по пустым ящикам на M позиций и не обнаружили начало следующего кортежа – больше на ленте кортежей нет или они есть где-то правее? Информационная неопределенность задачи состоит в отсутствии информации либо о количестве кортежей на ленте, либо о максимальном расстоянии между кортежами – при наличии такой информации (при разрешении информационной неопределенности) задача становится алгоритмически разрешимой.

в) Логическая неразрешимость (в смысле теоремы Гёделя о неполноте)

Проблема 5: Проблема «останова» (см. теорема 3.1);

Проблема 6: Проблема эквивалентности алгоритмов;

По двум произвольным заданным алгоритмам (например, по двум машинам Тьюринга) определить, будут ли они выдавать одинаковые выходные результаты на любых исходных данных.

Проблема 7: Проблема тотальности;

По произвольному заданному алгоритму определить, будет ли он останавливаться на всех возможных наборах исходных данных. Другая формулировка этой задачи – является ли частичный алгоритм P всюду определённым?

3. Проблема соответствий Поста над алфавитом Σ

В качестве более подробного примера алгоритмически неразрешимой задачи рассмотрим проблему соответствий Поста (Э. Пост, 1943 г.). Мы выделили эту задачу, поскольку на первый взгляд она выглядит достаточно «алгоритмизуемой», однако она сводима к проблеме останова и является алгоритмически неразрешимой.

Постановка задачи:

Пусть дан алфавит $\Sigma : |\Sigma| \geq 2$ (для односимвольного алфавита задача имеет решение) и дано конечное множество пар из $\Sigma^+ \times \Sigma^+$, т.е. пары непустых цепочек произвольного языка над алфавитом $\Sigma : (x_1, y_1), \dots, (x_m, y_m)$.

Проблема: Выяснить существует ли конечная последовательность этих пар, не обязательно различных, такая что цепочка, составленная из левых подцепочек, совпадает с последовательностью правых подцепочек – такая последовательность называется решающей.

В качестве примера рассмотрим $\Sigma = \{a, b\}$

1. Входные цепочки: $(abbb, b), (a, aab), (ba, b)$

Решающая последовательность для этой задачи имеет вид:

$(a, aab) (a, aab) (ba, b) (abbb, b)$, так как: $a a ba abbb \equiv aab aab b b$

2. Входные цепочки: $(ab, aba), (aba, baa), (baa, aa)$

Данная задача вообще не имеет решения, так как нельзя начинать с пары (aba, baa) или (baa, aa) , поскольку не совпадают начальные символы подцепочек, но если начинать с цепочки (ab, aba) , то в последующем не будет совпадать общее количество символов «a», т.к. в других двух парах количество символов «a» одинаково.

В общем случае мы можем построить частичный алгоритм, основанный на идее упорядоченной генерации возможных последовательностей цепочек (отметим, что мы имеем счетное множество таких последовательностей) с проверкой выполнения условий задачи. Если последовательность является решающей – то мы получаем результативный ответ за конечное количество шагов. Поскольку общий метод определения отсутствия решающей последовательности не может быть указан, т.к. задача сводима к проблеме «останова» и, следовательно, является алгоритмически неразрешимой, то при отсутствии решающей последовательности алгоритм порождает бесконечный цикл.

В теории алгоритмов такого рода проблемы, для которых может быть предложен частичный алгоритм их решения, частичный в том смысле, что он возможно, но не обязательно, за конечное количество шагов находит решение проблемы, называются *частично разрешимыми проблемами*.

В частности, проблема останова так же является частично разрешимой проблемой, а проблемы эквивалентности и тотальности не являются таковыми.