**Чарльз Петцольд. Читаем Тьюринга**

Все, кто изучали историю, технологию или теорию вычислительных машин, вероятно, сталкивались с понятием машины Тьюринга. Машина Тьюринга – это воображаемый компьютер, изобретенный в 1936 году английским математиком Аланом Тьюрингом (1912–1954) для того, чтобы помочь решить проблему математической логики (биографию Тьюринга см. [Эндрю Ходжес. Игра в имитацию](http://baguzin.ru/wp/?p=11970)). Однако оригинальная статья Алана Тьюринга, описывающая его изобретение, читается редко. Может ли читатель в наши дни взяться за статью, опубликованную 70 лет назад в Трудах Лондонского математического общества, и остаться на плаву достаточно долго, чтобы в полной мере проникнуться ею и, возможно, даже получить от нее удовольствие? Обо всем этом в предлагаемой книге.

Чарльз Петцольд. Читаем Тьюринга. Путешествие по исторической статье Тьюринга о вычислимости и машинах Тьюринга. – М.: [ДМК Пресс](http://dmkpress.com/catalog/computer/software_development/978-5-97060-010-8/), 2014 г. – 440 с.



Первоначальным толчком к написанию Тьюрингом этой статьи было намерение решить задачу, сформулированную немецким математиком [Давидом Гильбертом](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BB%D1%8C%D0%B1%D0%B5%D1%80%D1%82%2C_%D0%94%D0%B0%D0%B2%D0%B8%D0%B4) (1862–1943). Гильберта интересовал общий процесс определения доказуемости произвольных утверждений в математической логике. Нахождение этого «общего процесса» было известно, как [Entscheidungsproblem](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BC%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F) (с нем. – «решаемость задачи» или «проблема разрешимости»). Хотя побуждением к написанию статьи для Тьюринга была, конечно же, Entscheidungsproblem, на деле большая часть статьи – о вычислимых числах. По определению Тьюринга, это числа, которые могут быть вычислены машиной. Исследование вычислимых чисел составляет первые 60% статьи Тьюринга, которые могут быть прочитаны и поняты без знания работ Гильберта по математической логике.

**Глава 1. Прах Диофанта покоится в этой могиле**

Много веков назад в древней Александрии старик должен был хоронить своего сына. Убитый горем, он утешал себя составлением большого сборника алгебраических задач с решениями в книге, названной им Арифметика. Вот, пожалуй, и все, что известно о Диофанте из Александрии, и большая часть этого исходит от загадки, которая, как полагают, была написана его близким другом вскоре после его смерти:

*Прах Диофанта покоится в этой могиле. И она, о чудо, искусно поведает нам, сколь долог был его век. Шестую часть жизни Бог одарил его детством; когда минула еще одна двенадцатая часть, пушком покрылись его щеки; спустя седьмую долю жизни, Он зажег его брачную свечу, а на пятом году его брака Он послал ему сына. Увы, поздний и слабый ребенок, достигнув половины жизни отца своего, был забран холодной могилой. Еще четыре года горе свое утешал он наукой о числах, и тут конца жизни своей он достиг.*

Пусть *х* – это общее количество лет, прожитых Диофантом. Каждый отрезок жизни Диофанта – это либо доля его полной жизни (например, *х*/6 – это его детство), либо целое число лет (например, до рождения сына он был женат 5 лет).

Сумма всех этих периодов жизни Диофанта равна *х*, поэтому загадку можно записать просто алгебраически:

*х*/6 + *х*/12 + *х*/7 + 5 + *х*/2 + 4 = *х*

Наименьшее общее кратное знаменателей этих дробей – 84, поэтому умножим на него все члены уравнения слева и справа:

14*х* + 7*х* + 12*х* + 420 + 42*х* + 336 = 84*х*

Собрав множители *х* в правой части, а константы – в левой, получим:

420 + 336 = 84*х* – 14*х* – 7*х* – 12*х* – 42х

Или:

9*х* = 756

А само решение:

*х* = 84.

*Рациональными* называют числа, которые можно представить, как отношение (*ratio*) двух целых чисел. Например, 3/6 – рациональное число.

**Глава 2. Иррациональные и трансцендентные числа**

Любое число с повторяющимся набором цифр где-то после десятичной запятой – это всё еще рациональное число. Число

0,234562345623456.

– рациональное, если набор цифр 23456 повторяется бесконечно. Чтобы показать, что оно – рациональное, давайте приравняем его к *x*:

*x* = 0,234562345623456...

Теперь умножим обе части равенства на 100 000:

100 000 *x* = 23 456,23456234562346.

Хорошо известно, что если вычесть одну и ту же величину из обеих частей равенства, то равенство сохранится. Это значит, что можно вычесть из второго равенства первое: вычитаем *x* из 100 000 *x* и 0,23456… из 23 456,23456…, и десятичная часть исчезает:

99 999 *x* = 23 456.

Таким образом:

*х* = 23456/99999

Это – отношение, а значит, рациональное число.

Можно было бы предположить (как многие годы считали люди), что все числа – рациональные, но рассмотрим гипотенузу прямоугольного треугольника (рис. 1).



Рис. 1. Гипотенуза прямоугольного треугольника – иррациональное число

Согласно Теореме Пифагора, х2 = 12 + 12, или х2 = 2, или х = √2.

Докажем, что не существует отношения двух целых чисел, которое, будучи умноженным на себя, дает в результате 2. Воспользуемся методом от противного. Давайте начнем с предположения, что квадратный корень из 2 – рациональное число. А раз так, то существуют такие целые числа *a* и *b*, что: *a*/*b* = √2. Являются ли *a* и *b* оба четными? Если да, разделим их оба на 2 и заменим их половинами. Если и они – все еще четные, делим их тоже на 2 и продолжим так до тех пор, пока или *a*, или *b* (или они оба) не станет нечетным. Возведем обе части уравнения в квадрат: *a2*/*b2* = 2. Или: *a2* = *2b2*.

Заметим, что квадрат *a* вдвое больше квадрата *b*. Это означает, что квадрат *a* четен, а это возможно лишь тогда, когда само число *a* – четное. Ранее мы установили, что *a* и *b* не могут быть оба четными, таким образом, теперь мы знаем, что нечетно число *b*.

Если *a* – четное, то оно вдвое больше некоторого целого числа, которое мы обозначим *c*: (2*с*)2 = 2*b*2. Или: 4*с*2 = 2*b*2. Или: 2*с*2 = *b*2. Это значит, что квадрат *b* – четное число, а раз так, то и *b* – тоже четное, как и *а*, что противоречит первоначальному предположению о том, что, *а* и *b* не могут быть оба четными. Следовательно, исходное предположение о том, что квадратный корень 2 – рациональное число, неверно. Квадратный корень 2 – *иррациональное* число.

К алгебраическим уравнениям сводятся многие реальные задачи, поэтому они считаются весьма важными. Общий вид алгебраического уравнения:

$$\sum\_{i=0}^{N}a\_{i}x^{i}=0$$

где *N* – положительное целое число и *ai* – целые числа.

Решения алгебраического уравнения называются *алгебраическими* числами.

В 1740-х годах швейцарский математик Леонард Эйлер (1707–1783) предположил, что существуют неалгебраические числа, и назвал их *трансцендентными* (от лат. *transcendentis* – запредельный) числами, потому что они выходят за пределы алгебраических. Доказать существование трансцендентных чисел было трудно. Как доказать, что конкретное число не является решением некоторого довольно длинного и замысловатого алгебраического уравнения?

Вопрос о существовании трансцендентных чисел оставался открытым вплоть до 1844 года, когда французский математик Жозеф Лиувилль (1809–1882) придумал число и смог доказать, что оно – неалгебраическое. Вот первые 30 десятичных разрядов числа Лиувилля:

0,110001000000000000000001000000.

Лиувилль строит это число с помощью факториалов. Факториал числа – это произведение числа и всех положительных целых чисел, не превышающих его самого, которое обозначается восклицательным знаком:

1! = 1

2! = 1 х 2 = 2

3! = 1 х 2 х 3 = 6

4! = 1 х 2 х 3 х 4 = 24

5! = 1 х 2 х 3 х 4 х 5 = 120

и т.д. Число Лиувилля (как его иногда называют) содержит 1 в разрядах с номерами 1, 2, 6, 24, 120 и т.д. Лиувилль создал это число именно для того, чтобы доказать, что оно не является решением какого-либо алгебраического уравнения. Нарастающее расстояние между ненулевыми цифрами числа – ключ к доказательству.

В 1882 году немецкий математик Фердинанд Линдеман (1852–1939) доказал, что одно из самых известных иррациональных чисел всех времен – тоже трансцендентное. Это число *π* – отношение длины окружности к ее диаметру: *π* = 3,1415926535897932384626433832795…

Статья Тьюринга (и эта книга) ограничивается вещественными (не мнимыми) числами (рис. 2). Все эти классы чисел бесконечны.



Рис. 2. Вещественные числа

Последующее обсуждение будет несколько проще, если мы примем на вооружение некоторые элементы теории множеств. Множество – это совокупность объектов, которые называются элементами множества. Множество обычно заключается в пару фигурных скобок. Например, {1, 2, 3, 4}. Количество элементов множества называется *мощностью* *множества*. Мощность конечного множества, приведенного выше, равна 4. Но Георг Кантор (1845–1918) показал, что существует по крайней мере два вида бесконечности – перечислимая и неперечислимая. Бесконечные множества натуральных, рациональных и даже алгебраических чисел – перечислимые. Бесконечное множество трансцендентных чисел – неперечислимое. У этих типов множеств разная мощность.

Различные собственные подмножества вещественных чисел равносильны друг другу. Рассмотрим множество вещественных чисел от 0 до 1. Можно установить взаимно-однозначное соответствие между ним и множеством вещественных чисел, больших 1. Стоит только поделить 1 на каждое число из множества. Например, 0,5 соответствует 2; 0,25 – 4; 0,1 – 10, а 0,0001 – 10 000. Этот простой факт оказывается очень полезным: он означает, что можно изучать свойства вещественных чисел, ограничившись диапазоном от 0 до 1, и всё, что мы обнаружим, будет применимо ко всем вещественным числам. (Тьюринг, как и Кантор, тоже пользуется этим свойством в своей статье.)

В 1891 году Кантор опубликовал доказательство неперечислимости вещественных чисел. Оно получило название *диагонального доказательства*. Давайте ограничимся вещественными числами от 0 до 1. Предположим, что мы изобрели некий способ перечисления всех вещественных чисел (это еще одно доказательство от противного). Допустим, список начинается как-то так:

0,1234567890…

0,2500000000…

0,3333333333…

0,3141592653…

0,0010110111…

0,4857290283…

0,0000000000…

0,9999999999…

0,7788778812…

0,2718281828…

…

Хотя этот список бесконечен, мы можем убедиться, что в нем кое- чего нет. Посмотрим на цифры, образующие диагональ списка от левого верхнего угла к нижнему правому. Эти цифры выделены жирным шрифтом:

0,**1**234567890…

0,2**5**00000000…

0,33**3**3333333…

0,314**1**592653…

0,0010**1**10111…

0,48572**9**0283…

0,000000**0**000…

0,9999999**9**99…

0,77887788**1**2…

0,271828182**8**...

…

Теперь составим число из этих выделенных цифр:

0,**1531190918**…

Поскольку список вещественных чисел бесконечен и количество цифр в каждом из них тоже бесконечно, у этого числа бесконечное число цифр. Теперь увеличим каждую отдельную цифру этого числа на 1 (цифра 9 меняется на 0):

0,**26422010259**…

Есть ли это новое число в исходном списке? Давайте действовать последовательно: является ли это новое число первым в списке? Нет, это не так, потому что первая цифра первого числа в списке – 1, а первая цифра нового числа – 2. Является ли оно вторым числом в списке? И снова нет, потому что вторая цифра второго числа в списке – 5, а вторая цифра нового числа – 6. Является ли оно третьим числом в списке? Нет, потому что третья цифра третьего числа в списке – 3, а третья цифра нового числа – 4. И так далее. Новое число не может быть *N*-ым числом в списке, потому что *N*-ая цифра *N*-го числа в списке не равна *N*-ой цифре нового числа. Таким образом, список неполон, и наше исходное предположение неверно. Невозможно перечислить вещественные числа от 0 до 1. Мы еще раз убедились, что вещественные числа неперечислимы.

Для обозначения мощности множества натуральных чисел (и, следовательно, всякого перечислимого бесконечного множества) Кантор в 1895 году выбрал первую букву еврейского алфавита с нулевым нижним индексом – ℵ0 (произносится как «алеф нуль»). Кантор назвал его первым трансфинитным числом.

Кантор выяснил, что мощность континуума – это следующее за ℵ0 трансфинитное число, которое он назвал ℵ1. Это предположение называют континуум-гипотезой Кантора, и оно может быть выражено математически так:

$$ℵ\_{1}=2^{ℵ\_{0}}$$

Глубокий смысл всего этого состоит в том, что мощность перечислимых множеств не просто меньше мощности континуума:

$$ℵ\_{0}<2^{ℵ\_{0}},$$

а много, много, много, много, много меньше:

$$ℵ\_{0}< < < < < < < < < < < < < < <2^{ℵ\_{0}}.$$

Хотя в реальности бесконечность найти трудно, тем не менее в ее математической идее есть огромная польза. Оказывается, что некоторые математические доказательства, которые имеют реальные применения, включая изложенные в статье Тьюринга, основаны на различии между перечислимыми и неперечислимыми множествами (рис. 3). Видите проблему?



Рис. 3. Математические инструменты не годятся для выполнения работы

**Глава 3. Столетия прогресса**

Давид Гильберт (1862–1943) был одним из математиков, интересовавшихся в конце XIX в. созданием основ математики. С этими основами, заложенными незадолго до смены веков, математика, казалось, была на правильном пути, и Давида Гильберта пригласили для вступительной речи на Втором Международном математическом конгрессе в Париже в августе 1900 года. Гильберт обратился за советом к своему хорошему другу по университету Кёнигсберга математику литовского происхождения Герману Минковскому (1864–1909). Минковский предложил сделать речь Гильберта взглядом вперед, а не назад:

*Было бы очень заманчиво попытаться заглянуть в будущее и перечислить задачи, на которых математики должны испытать себя в наступающем веке. Такой подход заставит людей говорить о вашей лекции десятилетия спустя.*

Гильберт призвал своих коллег к решению 23 нерешенных проблем из нескольких областей математики. В частности, пункт № 1 касался «Проблемы Кантора о мощности континуума»: представляет ли собой мощность континуума следующее трансфинитное число вслед за мощностью множества натуральных чисел или между ними есть другие трансфинитные числа, которые должны быть приняты во внимание. Проблема № 2 была связана с «Непротиворечивостью аксиом арифметики». При обосновании непротиворечивости геометрии Гильберт опирался на непротиворечивость системы вещественных чисел и ее арифметики. Теперь сама система вещественных чисел нуждалась в аксиоматизации и в «доказательстве того, что она непротиворечива в том смысле, что конечное число логических шагов, опирающихся на нее, никогда не приведет к противоречивому результату». Проблема № 10. Определение разрешимости диофантова уравнения. Дано диофантово уравнение с произвольным числом неизвестных и рациональными целочисленными коэффициентами. Разработать процесс, согласно которому за конечное число шагов можно установить, разрешимо ли это уравнение в рациональных числах.

11 сентября 1917 года Гильберт выступив в Швейцарском математическом обществе в Цюрихе с докладом на тему «Аксиоматическое мышление». Это выступление положило начало тому, что в начале 1920-х годов стало известно, как Программа Гильберта. Этот подход в математике известен как формализм. В понимании Гильберта построение формальной математической системы начинается с определений, аксиом и правил построения теорем из аксиом. В идеале построенная система должна обладать четырьмя взаимосвязанными свойствами: независимость, непротиворечивость, полнота и разрешимость.

*Независимость* означает, что нет никаких лишних аксиом – ни одна аксиома не может быть выведена из других аксиом. Математики подозревали, что пять постулатов Евклида не обладают свойством независимости. Именно поэтому они пытались вывести пятый постулат из остальных четырех. Позже было установлено, что постулаты Евклида действительно независимы. *Непротиворечивость* – это, безусловно, самая важная характеристика любой аксиоматической системы. Не должно быть возможности вывести две теоремы, противоречащие друг другу! *Полнота* – возможность вывести из аксиом все истинные формулы. *Разрешимость*, или Entscheidung. Разрешающая процедура – общий метод определения доказуемости любой заданной правильной формулы.

Возможно, Гильберт был первым, кто соединил слова Entscheidung и Problem, но из Гёттингена (где преподавал Гильберт) в большой математический мир оно вышло фактически только в 1928 г. В этом году ассистент Гильберта Вильгельм Аккерман (1896–1962) помог собрать учебные лекции Гильберта в тонкую книжку, которая известна теперь как Hilbert&Ackermann. Одним из первых читателей Hilbert&Ackermann был австрийский студент-математик из Вены по имени Курт Гёдель (1906–1978). 7 сентября 1930 года Гёдель заявил, что ему удалось доказать, что добавление к логике предикатов первого порядка аксиом, позволяющих вывести арифметику (включая сложение и умножение), делает систему неполной. В такой системе он вывел и формулу, и ее отрицание. Если арифметика непротиворечива, то одно из этих утверждений должно быть истинным. Однако ни одно из них не могло быть доказано.

С помощью метода, названного позже гёделевой нумерацией, Гёдель воспользовался разработанной в рамках системы арифметикой, чтобы поставить в соответствие каждой формуле и каждому доказательству номер. После чего ему удалось построить формулу, которая утверждала собственную недоказуемость. Это созвучно математической формулировке Парадокса лжеца («Все, что я говорю, включая и это высказывание, есть ложь»). Статья Гёделя была опубликована в следующем году под названием «О формально неразрешимых предложениях Principia Mathematica и родственных систем I».

Одной из решающих предпосылок для теоремы о неполноте было то, что арифметика непротиворечива. Как следствие Гёдель показал также, что доказательство непротиворечивости арифметики в рамках самой системы было невозможно. Поскольку определенные формулы не могли быть ни доказаны, ни опровергнуты, оказалось, что эти формулы противоречивы. (Значит ли это, что арифметика и элементарная теория чисел противоречивы? Едва ли, и никто не верит, что это так. Проблема в том, что непротиворечивость не может быть доказана в рамках самой системы.)

После этого математики просто потеряли интерес к математической логике. Венгерский математик Джон фон Нейман (1903–1957), который бывал в Гёттингене в середине 1920-х годов, после Гёделя тоже забросил логику, но позже способствовал применению математической логики при разработке электронных цифровых машин.

Из теоремы Гёделя о неполноте не следовало, что разрешающего процесса не существует, но смысл ее был в том, что такой разрешающий процесс не мог определить истинность какой-либо формулы. В лучшем случае он мог определить доказуемость формулы. Девять страниц книги Hilbert&Ackermann были посвящены Entscheidungsproblem. Для нескольких стандартных (и общих) типов формул в математической логике уже были разработаны разрешающие процессы. И казалось не таким уж маловероятным, что общий разрешающий процесс тоже возможен.

Но этого не случилось. В 1936 году американский математик Алонзо Чёрч (1903–1995) пришел к выводу о том, что общий случай Entscheidungsproblem неразрешим. Работая независимо от Чёрча и применяя совершенно другую методологию, Алан Тьюринг пришел к тому же выводу.

**Глава 4. Годы учебы**

Алан Тьюринг опубликовал свою самую значительную работу в 1936 г., когда ему было 24 года.

О ВЫЧИСЛИМЫХ ЧИСЛАХ ПРИМЕНЕНИТЕЛЬНО К ENTSCHEIDUNGSPROBLEM
[Получено 28 мая 1936 г.][[1]](#footnote-1)

«Вычислимые» числа кратко можно описать как вещественные числа, чье десятичное представление можно вычислить конечным способом.

Тьюринг предполагает, что вычислимые числа – подмножество вещественных, что означает существование некоторых вещественных чисел, которые невычислимы.

Согласно моему определению, число вычислимо, если машина может записать его десятичное представление.

Те «числа, которые могли бы считаться естественно вычислимыми», – это числа, которые люди фактически уже вычислили и для которых существуют алгоритмы. Тьюринг даже не утруждает себя упоминанием о том, что все рациональные числа вычислимы. При этом Тьюринг не утверждает, что вычислимы все трансцендентные числа. Вычислимые числа перечислимы. Перечислимость вычислимых чисел означает, что они не то же самое, что вещественные числа, поскольку вещественные числа не перечислимы.

1. Вычислительные машины.

Мы можем уподобить человека, вычисляющего вещественное число, машине, которая может находиться только в конечном числе состояний q1, q2, ..., qR, которые назовем «*m*-конфигурациями».

Машина снабжена «лентой» (подобием бумаги), по которой она движется и которая разделена на части (называемые «клетками»), каждая из которых может содержать «символ».

В любой момент только одна клетка, скажем, *r*-я, содержащая символ S(r), находится «в машине».

Возможное поведение машины в каждый момент определяется m-конфигурацией qn и текущим символом S(r). Эту пару qn, S(r) будем называть «конфигурацией»: таким образом, конфигурация определяет возможное поведение машины.

2. Определения.

Если на каждом шаге поведение машины полностью определяется конфигурацией, мы назовем машину «автоматической машиной» (или a-машиной).

Сегодня наша работа с вычислительной техникой настолько интерактивна, что мы забываем о множестве компьютерных программ, которые работают постоянно, не отвечая на каждое нажатие клавиши и щелчок мыши пользователя.

Вычислительные машины. Если a-машина печатает два вида символов, первые из которых (называемые цифрами) состоят только из 0 и 1 (остальные называются символами второго вида), то машина будет называться вычислительной машиной.

Клод Э. Шеннон (1916–2001), чья магистерская диссертация *Символьный анализ реле и переключающих цепей*, написанная в 1937 году в MIT, доказала эквивалентность между цепями и булевой алгеброй, безусловно, оценил бы такой выбор. Слово «бит» («bit» – сокращение от «binary digit») не встречалось в печати вплоть до более поздней статьи Шеннона 1948 г.

«Вычисляемое машиной число – это двоичная дробь, полученная приписыванием к этой последовательности двоичной запятой». Например, если некая вычислительная машина Тьюринга печатает только 0 и 1, то «вычисляемая машиной последовательность» будет: 01. «Вычисляемое машиной число» получается приписыванием к этой последовательности двоичной запятой и цифры 0: 0,01. Это – двоичный эквивалент 1/4.

Тьюринг употребляет слово «конфигурация» в трех значениях:

* m-конфигурация – это одно из состояний машины;
* конфигурация – это сочетание m-конфигурации и текущего символа;
* полная конфигурация – это, по сути, «снимок» всей ленты в некоторый момент времени плюс текущая m-конфигурация и положение головки.

Изменения машины и ленты между последовательными полными конфигурациями назовем ходами машины. Если вычислительная машина никогда не записывает более чем конечное число символов первого вида, назовем eё циклической. В противном случае говорят, что она ациклическая.

Машина, которая печатает 0 и 1 и больше ничего подпадает под определение циклической машины Тьюринга. Машина где-то застревает и не может больше печатать цифры. Это плохо. Тьюрингу нужно, чтобы его машины продолжали печатать цифры всегда. Если машине действительно нужно вычислить двоичный эквивалент 1/4, она должна напечатать 0 и 1, а затем печать цифру 0 без конца.

Говорят, что последовательность вычислима, если она может быть вычислена ациклической машиной. Число вычислимо, если оно отличается на целое число от числа, вычисляемого ациклической машиной.

Итак, вычислимая последовательность:

010000...

Соответствующее вычислимое число:

0,010000...

Число

1,010000...

тоже считается вычислимым, потому что оно отличается на целое число от числа, вычисляемого машиной.

**Глава 5. Машины в работе**

3. Примеры вычислительных машин.

Можно создать машину для вычисления последовательности 10101… Машина должна иметь четыре m-конфигурации «b», «с», «f», «e» и уметь печатать «0» и «1». Поведение машины описывается в таблице (рис. 4), в которой «R» означает «машина перемещается так, чтобы просмотреть клетку непосредственно справа от текущей». Аналогично – для «L». «E» означает «текущий символ стирается», а «P» – «печать».

Обычно во втором столбце явно указывается конкретный текущий символ, например, 0 или 1, но Тьюринг использует еще слово «Any», которым обозначается любой символ, и слово «None», чтобы обозначить отсутствие символа, то есть пустую клетку. Машины Тьюринга всегда начинают с m-конфигурации b (от begin – начать).



Рис. 4. Таблица, описывающая поведение машины

Эти строки могут читаться примерно так: «В m-конфигурации b, когда текущая клетка – пустая (None), напечатать 0, переместить головку вправо и сменить m-конфигурацию на с». Давайте запустим эту машину и понаблюдаем за ее работой (рис. 5).



Рис. 5. Работа простейшей машины Тьюринга

Теперь машина вернулась в исходную m-конфигурацию b, и цикл начинается снова.[[2]](#footnote-2)

Если мы позволим символам L, R появляться в столбце операций несколько раз, мы можем значительно упростить таблицу (рис. 6).



Рис. 6. Упрощенная запись таблицы

В качестве несколько более сложного примера мы можем создать машину для вычисления последовательности 001011011101111011111… (рис. 7).



Рис. 7. Таблица, описывающая поведение более сложной машины

Попробуйте проследить за работой машины (рис. 8; слева от ленты указан выполненная m-конфигурация).



Рис. 8. Работа более сложной машины Тьюринга

Соглашение о записи символов только через одну клетку очень полезно: я всегда буду делать это. Я буду называть одну последовательность чередующихся клеток F-клетками, а другую последовательность E-клетками. Символы в E-клетках будут подлежать стиранию. Символы в F-клетках образуют непрерывную последовательность. Без пропусков, пока не достигнет конца.

Если символ β находится в F-клетке S и символ α находится в E-клетке справа от S, то будем говорить, что S и β помечены символом α. Процесс печати этого α будет называться пометкой β (или S) символом α.

Говорят, что этот 0 (в F-клетке) *помечен х* (рис. 9).



Рис. 9. Говорят, что ноль помечен иксом

**Глава 6. Сложение и умножение**

Машины Тьюринга могут складывать и умножать. Первый пример – небольшая машина Тьюринга, которая последовательно вычисляет все положительные целые числа. Эта машина не соблюдает соглашения Тьюринга, потому что записывает очередное число поверх предыдущего. При печати результатов она не пропускает ни одной клетки и заменяет каждый результат следующим, еще большим числом. Кроме того, учитывая, что эти числа – целые, я строил машину для печати цифр на ленте так, как мы обычно записываем целые числа, – слева направо, начиная со старших цифр. Несмотря на несоблюдение соглашений Тьюринга, эта машина действительно показывает, как увеличивается число при добавлении к нему 1, что является по меньшей мере одним из базовых умений современного компьютера. В своих примерах готическим буквам я предпочел содержательные слова, выделяемые жирным шрифтом. Эта конкретная машина начинает работу в конфигурации **begin** и имеет всего лишь три m-конфигурации (рис. 10).



Рис. 10. Машина для печати последовательных целых чисел в двоичном исчислении

m-конфигурация **begin** просто печатает единственный 0 и затем переключается на **increment**. m-конфигурация **increment** читает цифру. Если она – 0, **increment** меняет ее на 1 и заканчивает увеличение всего числа. Если она – 1, то **increment** меняет ее на 0 и сдвигает головку влево для переноса 1 в старший разряд числа и увеличения его на 1. m-конфигурация **rewind** перемещает головку вправо до самого младшего разряда числа для подготовки его к следующему увеличению.

**Глава 7. Они же – подпрограммы**

4. Сокращенные таблицы.

Существуют определенные типы операций, используемые почти всеми машинами. Там, где появляются такие операции, за счет применения «скелетных таблиц» мы можем значительно сократить таблицы m-конфигураций.

Для Тьюринга скелетные таблицы существуют исключительно для того, чтобы (с его точки зрения) проще создавать большие машины и (с нашей точки зрения) проще читать и понимать их. Далее Тьюринг приводит многочисленные примеры использования скелетных таблиц.

**Глава 8. Всё есть число**

Даже машины Тьюринга – это числа.

5. Перечисление вычислимых последовательностей.

Например, последовательность 001011011101111... определяется таблицей (см. второй пример главы 5), и фактически любая вычислимая последовательность может быть описана такой таблицей.

Давайте теперь дадим номера m-конфигурациям, назвав их q1, ... , qR. Начальной m-конфигурацией всегда должна быть q1. Мы также дадим номера символам S1, ..., Sm и, в частности, пусто = S0, 0 = S1, 1 = S2. Строки таблицы теперь имеют вид, как на рис. 11.



Рис. 11. Таблица с тремя строками стандартного вида

Эти три стандартных вида Тьюринг пометил справа как N1, N2 и N3. Все три что-то печатают; разница только в том, что головка сдвигается влево (L), вправо (R) или стоит на месте.

Давайте из каждой строки вида (N1) образуем выражение вида qjSjSkLqm, из каждой строки вида (N2) образуем выражение вида qjSjSkRqm; а из каждой строки вида (N3) образуем выражение вида qjSjSkNqm.

Выпишем из таблицы машины все образованные таким образом выражения и разделим их точками с запятой. Таким образом, мы получаем полное описание машины.

В этом описании мы заменим qi буквой «D», за которой следует буква «A», повторенная i раз, а Sj - на «D», за которой следует «С», повторенная j раз.

Это новое описание машины можно назвать стандартным описанием (S.D). Оно составлено только из букв «A», «C», «D», «L», «R», «N» и «;».

Наконец, если мы меняем «A» на «1», «С» на «2», «D» на «3», «L» на «4», «R» на «5», «N» на «6» и «;» на «7», у нас будет описание машины в виде арабских цифр.

Целое число, представленное этими цифрами, можно назвать описательным номером (D.N). Машину, чей D.N равен *n*, можно обозначить как М(n).

Каждой вычислимой последовательности соответствует хотя бы один описательный номер. Следовательно, вычислимые последовательности и числа перечислимы.

Достигнутое здесь весьма интересно. Рассмотрим машину Тьюринга, которая вычисляет *π*. Обычно мы изображаем число *π* бесконечной последовательностью:

*π* = 3,1415926535897932384626433832795...

Теперь же мы можем представить *π* конечным целым числом – описателем машины Тьюринга, вычисляющей эти цифры. Какое из представлений *π* лучше? Первые 32 цифры с последующим многоточием? Или описатель машины Тьюринга, которая может сгенерировать столько цифр, сколько позволит нам наше терпение? В каком-то смысле описатель – более фундаментальное числовое представление *π*, поскольку описывает алгоритм вычисления числа.

Число, которое является описателем ациклической машины, будем назвать приемлемым числом.

Легко определить, является ли некоторое целое число правильным описателем, но Тьюринг теперь утверждает, что не существует общего процесса для определения того, что данный описатель представляет ациклическую машину и печатает предполагаемую бесконечную последовательность нулей и единиц. Не существует общего процесса для определения того, что машина может встретить символ, которого не ждет, зациклится, печатая пробелы, сломается, сгорит, переполнится или попадает «пальцем в небо».

**Глава 9. Универсальная машина**

6. Универсальная вычислительная машина.

Можно создать одну-единственную машину, которую можно использовать для вычисления любой вычислимой последовательности. Если этой машине V предоставить ленту, в начале которой записано S.D некоторой вычислительной машины M, то V вычислит ту же самую последовательность, что и M.

Хорошо известно, что описание Универсальной машины Тьюринга содержит несколько ошибок. (Весьма удивительно, как мало ошибок оно содержит, учитывая, что у Тьюринга не было возможности моделировать ее на настоящем компьютере.) Поскольку Универсальная машина очень важна для доводов Тьюринга в оставшейся части его статьи, он доказывает существование такой машины, создавая ее фактически полностью со всеми подробностями. Но стоит только понять ее основной механизм, как можно обнаружить, что эти подробности довольно скучны. Никто вас не накажет, если вы не усвоите всех символов и функций в описании Тьюринга.

Несмотря на небольшие опечатки и «ляпы», Тьюринг сделал нечто совершенно необыкновенное. Он обобщил вычисления, показав, что одна универсальная машина, запрограммированная должным образом, может выполнить работу любой вычислительной машины. В одной широко известной книге по вычислимости говорится: «теорема Тьюринга о существовании универсальной машины Тьюринга – одна из интеллектуальных вех прошлого столетия».

Все это вызывает вопрос: Компьютер изобретен Аланом Тьюрингом?

**Глава 10. Вычислительные машины и вычислимость**

Представив вычислительную машину, которая не делает почти ничего, Тьюринг фактически придумал универсальный компьютер «общего назначения». Это было революционной идеей. Тогда считалось общепринятым, что компьютеры следует проектировать специально для решения конкретных задач. Даже люди, серьезно занимавшиеся разработкой цифровых машин, часто не осознавали универсальности цифровой логики. Говард Эйкен, например, был одним из настоящих первопроходцев вычислительной техники и занимался ею с 1937 года. Тем не менее в 1956 году он заявил: «Если бы оказалось, что внутренняя логика машины, разработанной для численного решения дифференциальных уравнений, совпадает с логикой машины, предназначенной для обработки складских накладных, я бы расценил это как самое удивительное совпадение, с которым я когда-либо сталкивался».

Сравните заявление Эйкена 1956 года с тем, что писал Тьюринг в 1950 году: «Это особое качество цифровых компьютеров имитировать любую машину с дискретными состояниями, позволяет сказать, что они являются универсальными машинами. Важным следствием существования такого рода машин, без учета быстродействия, является то, что не нужно разрабатывать разнообразные новые машины для выполнения разнообразных вычислительных процессов. Все они могут быть выполнены одним цифровым компьютером, программируемым для каждого случая должным образом. Можно видеть, что вследствие этого все цифровые компьютеры в некотором смысле эквивалентны».

Место Алана Тьюринга в общей истории вычислительной техники никогда не было вполне ясным. В одной устоявшейся истории он заслуживает лишь упоминания, но, когда историю пишет видный математик, который рассматривает компьютер как физическое воплощение математических идей, Тьюринг становится основным действующим лицом.

Тьюринг показал, что не может быть общего процесса определения того, что машина – ациклическая. Отсюда следует, что не может быть компьютерной программы, которая определит окончательную судьбу других компьютерных программ.

Тьюринг также разрешил парадокс диагонального процесса: сначала он установил, что вычислимые числа перечислимы, хотя диагональный процесс должен указывать, что можно создать вычислимое число не из списка. Тьюринг показал, что диагональ не может быть вычислена конечным образом и, следовательно, невычислима. Вычислимые числа могут быть перечислимыми, но в действительности их нельзя перечислить за конечное число шагов.

**Глава 11. О машинах и людях**

Увлечение Тьюринга связью человеческого мозга и машин продолжалось и после его статьи 1936 г. о вычислимых числах. Другая известная работа Тьюринга «Вычислительные машины и интеллект» была опубликована в октябрьском 1950 года выпуске философского журнала Mind. «Могут ли машины мыслить?» – спрашивает Тьюринг. Теперь он придумывает тест с человеком, сидящим за телетайпом. (Современным эквивалентом ему может быть средство передачи сообщений или что-то еще, что не позволяет людям видеть или слышать того, с кем они общаются.) Человеку позволено задавать вопросы и получать ответы. Если на другом конце - действительно компьютер и человек не может сказать, что это – компьютер, мы вынуждены признать, что компьютер обладает разумом. Этот тест получил известность как тест Тьюринга, и он до сих пор остается спорным.

15 марта 1951 года Алан Тьюринг в знак признания его работы по вычислимым числам был избран членом Королевского общества. Его поручителями были Макс Ньюмэн и Бертран Рассел.

**Глава 12. Логика и вычислимость**

Следующая часть статьи Тьюринга требует некоторого знакомства с математической логикой. Начну я с того, что раньше называли исчислением высказываний, но более известное сегодня как пропозициональное исчисление или пропозициональная логика.[[3]](#footnote-3) Пропозициональная логика (логика высказываний) имеет дело со всеми декларативными утверждениями (или высказываниями), которые имеют истинностное значение – то есть их можно считать либо истинными, либо ложными. Например: сегодня среда, семь – простое число, идет дождь.

Некоторые из этих высказываний истинны, некоторые ложны, а некоторые могут быть истинными для меня, но ложными для вас. (Без драки!) В пропозициональной логике высказывания имеют единственное непротиворечивое, недвусмысленное истинное значение, и чем меньше мы делаем вид, что успешно разбираем что-то, кроме математических высказываний, тем меньше мы будем путаться.

В пропозициональной логике высказывания часто представляются прописными буквами. Первыми буквами алфавита (A, B и C) часто обозначаются высказывания с фиксированным значением истинности, тогда как последние буквы алфавита (X, Y и Z) используются для высказываний-переменных. Мы можем соединять отдельные высказывания с помощью определенных связок (соединителей), чтобы получать еще большие составные высказывания.

Первой из этих связок является строчная буква v, от латинского слова *vel* со значением *или*, а точнее *включающее или* как противоположность латинскому *aut*, *исключающему или*. Высказывание X v Y истинно, если истинно либо X, либо Y, либо они оба. Для представления возможных комбинаций X и Y полезна небольшая таблица истинности (рис. 12).



Рис. 12. Связка v со значением *или*, а точнее *включающее или*

Разрешается опускать символ v, если это не приводит к путанице. Формула XY равносильна формуле X v Y. Заметьте, что я не записываю равносильность этих двух формул одной строкой, связав их знаком равенства. Знак равенства не включен в язык исчисления высказываний. Когда мы говорим, что одно высказывание равносильно другому, мы подразумеваем, что они имеют одни и те же значения истинности для соответствующих значений истинности составляющих их высказываний. Мы выражаем эту равносильность на естественном языке, также известном как метаязык. Строгое разделение языка логики и метаязыка необходимо, чтобы избежать путаницы. Для обозначения равносильности используется метаязыковое сокращение eq. (от англ. equivalent): X v Y eq. XY.

Запомните, что это сокращение не является частью языка пропозициональной логики и необходимо исключительно для удобства. Понятие «и» представляется знаком амперсанда &. Формула X & Y истинна тогда и только тогда, когда истинны и X, и Y (рис. 13).



Рис. 13. Связка & со значением *и*

Операция «и» часто называется конъюнкцией от известного из грамматики понятия «союз» (conjunction – союз); в свою очередь, операция «или» часто называется менее знакомым словом дизъюнкция.

Из таблиц истинности очевидно, что: X v X eq. X, X & X eq. X.

Если в составном высказывании используются обе связки, то v выполняется раньше &. Если же нужен иной порядок выполнения операций, его можно изменить правильной расстановкой круглых скобок или использовать их всегда для большей строгости.

X & Y v Z eq. X & (Y v Z).

Эти высказывания не равносильны:

(X & Y) v Z.

Например, если X ложно, Y истинно и Z истинно, то первые два высказывания ложны, а последнее истинно.

Не буду забивать вам голову различными правилами, которые обеспечивают строгую парность круглых скобок и появление логических связок только в нужных местах. Эти правила связаны с понятием правильной формулы. Слова «истина» и «ложь» не входят в словарь пропозициональной логики так же, как и символы «T» и «F», но для удобства их можно использовать вместо пропозициональных символов. Можно считать, что T – это всегда истинное высказывание, а F – всегда ложное высказывание. Впредь в таблицах истинности я буду использовать T и F.

Следующие тождества очевидны из таблиц истинности: X v T eq. T, X v F eq. X, X & T eq. X, X & F eq. F. Из таблиц истинности видно, что обе операции коммутативны: X v Y eq. Y v X, X & Y eq. Y & X. Обе операции также ассоциативны: X v (Y v Z) eq. (X v Y) v Z, X & (Y & Z) eq. (X & Y) & Z. Обе операции дистрибутивны по отношению друг к другу: X v (Y & Z) eq. (X v Y) & (X v Z), X & (Y v Z) eq. (X & Y) v (X & Z).

Если в таблицах истинности заменить F на 0, а T на 1, то можно видеть, что конъюнкция в точности равносильна умножению двух однозначных двоичных чисел, а дизъюнкция чем-то напоминает сложение. Поэтому конъюнкцию иногда называют «логическим умножением», а дизъюнкцию – «логическим сложением». Однако в использовании этих терминов есть некоторая натянутость, поэтому они не поощряются.

Конъюнкция и дизъюнкция – это бинарные операции. Единственная унарная (одноместная) операция называется «не», или «отрицание», и обозначается черточкой, очень похожей на знак минус (рис. 14).



Рис. 14. Операция отрицания

Отрицание всегда вычисляется первым: знак отрицания применяется только к идущему вслед за ним символу. Двойное отрицание взаимоуничтожается: – –X eq. X.

Эти два отношения - самые простые: X v -X eq. T, X & -X eq. F. Два самых основных – но и самых интересных – логических отношения объединяют дизъюнкцию, конъюнкцию и отрицание. Их называют законами Де Моргана в честь математика XIX века [Огастеса Де Моргана](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%2C_%D0%9E%D0%B3%D0%B0%D1%81%D1%82%D0%B5%D1%81_%D0%B4%D0%B5) (1806–1874), хотя это основополагающее понятие было известно еще Аристотелю:

–(X v Y) eq. –X & –Y

–(X & Y) eq. –X v –Y.

Эти тождества понятны в обычной речи. Например, «нет дождя или снега» или –(X v Y), это то же самое, что «нет дождя и нет снега» или -X & - Y. Когда мне говорят «Увы, ты явно не богат и красив» или –(X & Y), я могу лишь сделать вывод «надо полагать, я или беден, или уродлив, или все это вместе» или –X v –Y.

Заметьте, что законы Де Моргана можно записать так, чтобы все знаки отрицания были сгруппированы в одной части тождества: X v Y eq. –(–X & –Y), X & Y eq. –(–X v –Y).

В таблицах истинности для операций v и & можно заменить все F на T и все T на F, и получится таблица истинности для противоположной операции. Это называется «принципом двойственности», который применим также и к составным высказываниям. Вот одно из них:

X & –Y v Z.

Применим отрицание к каждой переменной и поменяем местами v и & (не забыв вставить скобки, чтобы не исказить смысл высказывания) – и получим отрицание первоначального высказывания:

–X v (Y & –Z).

Если нужно проверить, что эти два высказывания действительно отрицают друг друга, можно создать небольшую таблицу истинности для проверки всех значений (рис. 15).



Рис. 15. Таблица проверка истинности высказываний X & –Y v Z и –X v (Y & –Z)

Последние два столбца содержат противоположные значения истинности, значит:

X & –Y v Z eq. –(–X v (Y & –Z)).

У операции исключающего «или» нет собственного обозначения, но она вычисляется по следующей формуле (рис. 16):

(X v Y) & –(X & Y).



Рис. 16. Таблица истинности для операции *исключающего или*

Исключающее «или» очень похоже на обычную дизъюнкцию, за исключением случая, когда и X, и Y истинны. Если мы применим законы Де Моргана ко второй половине высказывания для исключающего или, то получим: (X v Y) & (–X v –Y). Симметрия такого вида гораздо более симпатична. В компьютерах исключающее «или» используется для вычисления суммы двух двоичных цифр, а конъюнкция – для бита переноса.

Третья бинарная операция – сложная. Она называется «импликация». X –> Y читается как «X влечет Y» или «если X, то Y». Следует предостеречь, что многие, впервые столкнувшись с ее таблицей истинности, недоумевают: «Здесь что-то не так» (рис. 17).



Рис. 17. Таблица истинности для операции импликации

Первые две строки таблицы могут показаться странными. Если X ложно, то почему X –> Y должно быть истинно независимо от значения Y? Давайте посмотрим на это с другой стороны, начав с предположения, что X –> Y истинно. Если X –> Y истинно и X истинно, то Y должно быть истинно. Однако если X не истинно, то что можно сказать о Y? Ничего. Y может быть чем угодно. Именно поэтому X –> Y может быть истинным, если X ложно независимо от Y.

Рассмотрим высказывание «Если идет дождь, то есть осадки». Это высказывание истинно, если идет дождь. Но оно также истинно, если дождь не идет (скажем, идет снег или град). Единственный случай, когда высказывание ложно, – это когда идет дождь, но осадков нет.

Импликация очень часто используется в математической логике. Довольно нередко слева от знака импликации – формула, о которой мы знаем, что она истинна. Тогда если мы можем показать, что само высказывание истинно, то мы можем заключить, что формула справа истинна. Импликация не коммутативна и не ассоциативна. Однако X –> Y eq. –Y –> –X. Второе высказывание называется контрапозицией. Если идет дождь, я беру зонт. У меня нет зонта, значит, дождя не должно быть. У импликации есть очень простая связь с дизъюнкцией: X –> Y eq. –X v Y. Иными словами, X –> Y истинно, если либо X ложно, либо Y истинно. Импликацию можно также выразить через конъюнкцию: X –> Y eq. –(X & –Y).

Если слева от знака импликации – конъюнкция произвольного числа членов, то любой из этих членов может быть справа от знака импликации:

X & Y –> X

X & Y –> Y

Еще один оператор – двунаправленное условие (или «тогда и только тогда»), обозначаемое тильдой (рис. 18).



Рис. 18. Таблица истинности двунаправленного условия

X ~ Y истинно только тогда, когда X и Y имеют одинаковые значения истинности. Тьюринг в своей статье вообще не использует двунаправленное условие. Тем не менее стоит заметить, что оно равносильно конъюнкции импликаций противоположных направлений:

X ~ Y eq. (X –> Y) & (Y –> X).

Если допустить, что это тождество имеет смысл, то оно верно только тогда, когда T –> T истинно (что несомненно) и F –> F тоже истинно (что сомнительно). Кроме того, T –> F и F –> T должны иметь противоположные значения истинности. Несомненно, что T –> F должно быть ложно, что означает, что F –> T должно быть истинно. Это подтверждает правильность таблицы истинности для импликации.

Допустим, дано следующее высказывание: X v Y v (–X & –Y). Что вы можете о нем сказать? Истинно оно или ложно? Вы можете возразить и заявить, что этого сказать нельзя, покуда неизвестны значения X и Y. Тогда вам предлагается составить таблицу истинности, дабы проверить все комбинации X и Y (рис. 19). Независимо от конкретных значений X и Y это высказывание всегда истинно. Мы говорим, что такое высказывание – *тавтология*, или что оно всюду справедливо. Всюду справедливые высказывания очень любимы в математической логике, потому что они истинны независимо от значений истинности их составляющих.



Рис. 19. Таблица истинности тавтологического высказывания

Давайте рассмотрим другой случай (рис. 20). Это высказывание всюду ложно. Это – опровержение (или противоречие). Отрицание тавтологии – это противоречие, а отрицание противоречия – тавтология.



Рис. 20. Таблица истинности опровержения

И третий случай (рис. 21). Это высказывание иногда истинно, а иногда ложно в зависимости от значений X и Y. Это высказывание, как говорят, выполнимо, то есть обладает способностью быть истинным при определенной комбинации логических значений.



Рис. 21. Таблица истинности выполнимого высказывания

Всюду справедливое высказывание (тавтология) считается также выполнимым. Высказывание всюду справедливо тогда и только тогда, когда отрицание этого высказывания невыполнимо. Для любого высказывания мы можем использовать таблицу истинности, чтобы определить, является ли это высказывание всюду справедливым, опровержимым (противоречивым) или просто выполнимым. Процесс построения таблицы истинности – чисто механический. Он не требует никакого особого вдохновения, понимания или интуиции. Программист может легко представить себе программу, которая читает высказывание пропозициональной логики, проверяет его и печатает слова «справедливо», «противоречиво» или «выполнимо».

Поэтому мы говорим, что высказывания в исчислении высказываний разрешимы. В пропозициональной логике существует разрешающая процедура определения справедливости или выполнимости любого высказывания. Иными словами, Entscheidungsproblem для исчисления высказываний решена.

**Глава 13. Вычислимые функции**

**Глава 14. Главное доказательство**

В определенном смысле заключительный раздел статьи Тьюринга – самый важный, потому что именно здесь он показывает, что «Entscheidungsproblem не может быть решена». Для того времени это было важным выводом, но построенная Тьюрингом для получения этого результата конструкция – воображаемое устройство, известное теперь как машина Тьюринга, – в конечном счете станет интересней и плодотворней, нежели само доказательство.[[4]](#footnote-4)

**Глава 15. Лямбда-исчисление**

**Глава 16. Постижение континуума**

История науки, математики и техники особенно подвержена вольным истолкованиям. Унаследовав «правильные» научные теории и «точные» технологии, мы можем вытянуть из недр истории цепочку причинно-следственных связей, которые привели к этому непреложному результату. Допущенные неверные шаги скрадываются, а если обсуждаются исторические разногласия или вражда, то они всенепременно заканчиваются победой над теми, кто пытался препятствовать движению вперед к прекрасному мгновению, которое все мы теперь должны восхвалять.

Философия математики – обширная и сложная область, но, наверное, самый основной ее вопрос – и простой, и тревожащий: *Насколько существование математических объектов независимо от людей, их изучающих?* Математики просто открывают математические модели, которые уже существуют внутри самой материи вселенной, почти так же, как астрономы открывают звезды и другие астрономические тела? Или же математики выдумывают математику, как инженер конструирует новый пылесос или композитор сочиняет оперу?

По одну сторону этих споров – реалисты, или платоники (последователи учения Платона), которые верят в слова [Роджера Пенроуза](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D0%BD%D1%80%D0%BE%D1%83%D0%B7%2C_%D0%A0%D0%BE%D0%B4%D0%B6%D0%B5%D1%80) об «объективности математической истины. Платоническое бытие, как я его вижу, определено наличием объективных внешних законов, которые не зависят ни от нашего частного мнения, ни от особенностей нашей культуры». По другую сторону этих споров – конструктивисты, которые видят математику как исключительно человеческое изобретение. Для конструктивистов кажущаяся перманентность и трансцендентность математики – это просто иллюзия, усиленная способностью человека к образному мышлению – способностью, развившейся в нашем мозгу на протяжении миллионов лет эволюции.

**Глава 17. Весь мир – машина Тьюринга?**

Независимо от того, насколько хорошо вы понимаете идею и принцип действия машины Тьюринга, это на самом деле не поможет вам построить вычислительную машину. У Тьюринга не было намерений предложить свою машину в качестве проекта настоящего компьютера. Его машина – упрощенная абстрактная модель вычислений, выполняемых человеком или машиной. Первоначальная и вполне определенная цель создания Тьюрингом своей машины – доказать, что не существует общей разрешающей процедуры для логики первого порядка. И только позже эти воображаемые устройства начали способствовать нашему пониманию теории вычислений. Это длилось около 20 лет, после чего машина Тьюринга стала предметом изучения дисциплины, известной нам теперь как информатика (computer science).

К 1936 году существовали три различные формулировки интуитивного понятия эффективной вычислимости:

* машины Тьюринга;
* рекурсивные функции в определении Курта Гёделя, данном в 1934 году на основе предположения Жака Эрбрана, а затем исследованном Клини; и
* определимые функции, разработанные Чёрчем и его студентами, а более других – Клини.

Равносильность этих трех формулировок была установлена Тьюрингом частично в приложении к его статье о вычислимых числах и более строго в его статье 1937 года «Вычислимость и λ-определимость», а также Стивеном Клини в его статье 1936 года «λ-определимость и рекурсивность». С тех пор термины «рекурсивная функция» и «вычислимая функция» стали почти синонимами.

Машина Тьюринга не только определила основные требования к эффективной вычислимости, но и установила ее пределы: ни один из известных сегодня компьютеров или языков программирования не мощнее машины Тьюринга; ни один компьютер или язык программирования не может решить проблему останова; ни один компьютер или язык программирования не может предсказать судьбу другой компьютерной программы. За эти пределы не поможет вырваться ни «наилучший» язык программирования, ни какая-то другая машина. В лучшем случае можно лишь быстрее выполнять работу.

По этой причине универсальность вычисления – и по возможностям, и по ограничениям – видимо, лежит в основе любого типа деятельности, связанной с обработкой данных. Эти ограничения, скорее всего, присущи материальному миру природы так же, как законы термодинамики.

**Глава 18. Долгий сон Диофанта**

1. Оригинальный текст статьи Тьюринга набран с отступом. [↑](#footnote-ref-1)
2. В статье Тьюринга нет ни одного рисунка ленты с клетками. [↑](#footnote-ref-2)
3. См. также заметку [Булева логика: критерии И, ИЛИ](http://baguzin.ru/wp/?p=7645) применительно к формулам и функциям Excel. [↑](#footnote-ref-3)
4. Содержание глав 12–16 весьма специальное. [↑](#footnote-ref-4)