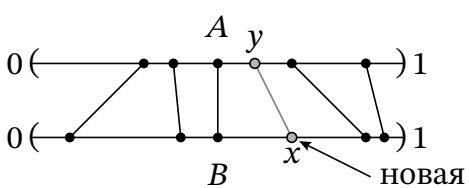
А. Шень

Диагональ Кантора и другие рассуждения



Montpellier (Grabels), France

А. Шень

Диагональ Кантора и другие рассуждения

Москва
Издательство МЦНМО

УДК 519.83
ББК 22.1
Ш47

Шень А.

Ш47 Диагональ Кантора и другие рассуждения. — М.: МЦНМО, 2025. — 24 с.: ил.

ISBN 978-5-4439-1948-5

Немецкий математик XIX века Георг Кантор (1845–1918) придумал то, что теперь называют «теория множеств» — ныне это стандартный язык математики. Начал он с того, что обнаружил: действительных чисел (бесконечных дробей) больше, чем целых, а потом ещё много чего доказал.

Мы разберём некоторые из рассуждений Кантора (начав с «диагонального аргумента») и примеры, где такой подход оказывается полезным.

ББК 22.1

Оригинал-макет предоставлен автором. Электронная версия книги является свободно распространяемой и доступна по адресу
`https://www.mccme.ru/shen/cantor.pdf`
`https://www.mccme.ru/shen/cantor.zip`

Научно-популярное издание

Александр Шень

Диагональ Кантора и другие рассуждения

В соответствии с Федеральным законом № 436-ФЗ
от 29 декабря 2010 года издание маркируется знаком 6+

Подписано в печать 10.06.2025 г. Формат 60 × 90 1/16. Бумага офсетная.
Печать офсетная. Печ. л. 1,5. Тираж 1000 экз. Заказ №

Издательство Московского центра непрерывного математического образования
119002, Москва, Большой Власьевский пер., 11. Тел. (499) 241-08-04.

Отпечатано в ООО «Типография „Миттель Пресс“».
Москва, ул. Руставели, д. 14, стр. 6.
Тел./факс +7 (495) 619-08-30, 647-01-89
E-mail: mittelpress@mail.ru

ISBN 978-5-4439-1948-5

1. Математический фокус

Наверное, проще всего объяснить «диагональное рассуждение Кантора[1]» на примере такого воображаемого[2] математического фокуса.

Фокусник рисует на доске квадратную таблицу 8×8 (как шахматная доска), отворачивается и предлагает зрителям заполнить все клетки таблицы нулями и единицами. В таблице 8 строк, и в каждой строке получается последовательность из восьми нулей и единиц («двоичное слово длины 8») — всего восемь таких последовательностей.

0	0	0	1	0	0	0	0
1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	1
1	0	0	1	0	0	1	0
1	0	0	0	1	0	0	0
0	0	0	0	1	1	1	1
1	0	0	1	1	1	0	1
1	1	0	0	1	1	1	0

Потом фокусник оборачивается и с одного взгляда на доску называет последовательность из 8 нулей и единиц, *которой в строках таблицы нет*.

С одной стороны, ничего особенного. Двоичных слов длины 8 (программисты называют их «байтами») всего $256 = 2^8$ штук: на первом месте может стоять любая из цифр 0 и 1, для каждого из этих вариантов есть два варианта цифры на втором месте, для каждого из четырёх получившихся есть два варианта на третьем месте — и так далее. А в таблице всего 8 последовательностей — ну вот фокусник и берёт любую из 248 ($= 256 - 8$) оставшихся.

С другой стороны, даже и прочесть все последовательности в таблице не так просто, и ещё надо прикинуть, какой не встретилось. Как же фокусник прямо сразу называет последовательность, которой нет?

[1] Георг Кантор (Georg Ferdinand Ludwig Philipp Cantor, 1845–1918), немецкий математик. Родился и прожил первые десять лет в Санкт-Петербурге. Петербургским математикам — уже в постсоветское время — удалось добиться установки мемориальной доски на доме, где он жил (11-я линия Васильевского острова, дом 24, во дворе).

[2] Вряд ли такой фокус поразит зрителей.

1 Придумайте простой способ показывать такой фокус.

Можно действовать так. Посмотрим на цифры, стоящие на диагонали квадрата, и заменим их на противоположные ($0 \leftrightarrow 1$):

0	0	0	1	0	0	0	0
1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	1
1	0	0	1	0	0	1	0
1	0	0	0	1	0	0	0
0	0	0	0	1	1	1	1
1	0	0	1	1	1	0	1
1	1	0	0	1	1	1	0

1 0 1 0 0 0 1 1

Получится последовательность из 8 цифр (на рисунке 10100011), которая отличается от первой строки в первой цифре, от второй строки во второй цифре, ..., от восьмой строки в восьмой цифре. И этого достаточно: две последовательности совпадают, если у них равны *все* цифры, и для несовпадения достаточно отличия *в одной* цифре.

Можно ещё сказать так. Нам нужно было выполнить восемь требований: отличаться от первой строки, от второй строки и так далее. Мы их выполнили по очереди, выделив для каждого требования одну из цифр.

2 Можно ли было вместо этой диагонали использовать горизонталь? вертикаль? другую диагональ?

(Горизонталь и вертикаль не помогут, а другую диагональ использовать, конечно, можно — получится последовательность, которая от первой отличается в последней цифре, ..., от последней — в первой.)

3 Докажите, что $2^n > n$ при всех целых положительных n.

В самом деле, если было бы $2^n \leqslant n$, то в таблице $n \times n$ хватило бы строк, чтобы записать все n-битовые слова, и фокус бы не удался. (Конечно, это неравенство можно объяснить и без фокусов: при $n = 1$ получается $2 > 1$, а затем при увеличении n на единицу левая часть увеличивается вдвое, а правая только на 1.)

2. Несчётность бесконечных дробей

Ровно то же самое рассуждение с диагональю можно провести и для бесконечной таблицы. Представим себе, что наша таблица бесконечна вправо и вниз (так что заполняет четверть бесконечной плоскости) и вся заполнена нулями и единицами. В каждой её строке, таким образом, записана бесконечная последовательность нулей и единиц.[3]

0	0	0	1	0	0	0	0	...
1	**1**	1	1	1	1	1	1	...
0	1	**0**	1	0	1	0	1	...
1	0	0	**1**	0	0	1	0	...
1	0	0	0	**1**	0	0	0	...
0	0	0	0	1	**1**	1	1	...
1	0	0	1	1	1	**0**	1	...
1	1	0	0	1	1	1	**0**	...
.....								⋱
1	0	1	0	0	0	1	1	...

Если прочесть теперь (слева направо и сверху вниз) последовательность цифр по диагонали, то эта последовательность пересекается с i-й строкой на i-м месте, поэтому совпадает с i-й строкой в i-м знаке. А если теперь все цифры с диагонали заменить на противоположные, то получится «обращённая диагональ» — последовательность, которая *отличается от i-й на i-м месте* и потому отсутствует в таблице.

Выходит, что *невозможно занумеровать все бесконечные последовательности нулей и единиц целыми положительными числами* (не пропустив ни одной). В самом деле, предположим, что мы такое сделали. Заполним таблицу, написав в первой строке первую последовательность, во второй строке вторую и так далее. Но обращённая диагональ в этой таблице пропущена (ведь она отличается от любой строки таблицы) — значит, занумеровали мы не все!

---

[3]Можно говорить о бесконечных двоичных дробях и соответствующих им действительных числах на отрезке $[0, 1]$ — считая, что последовательность написана после запятой и слева от запятой стоит нуль. Небольшая проблема, правда, в том, что одно и то же число, скажем 1/2, может быть записано и как 0,10000..., и как 0,01111..., так что проще иметь дело с последовательностями — и даже можно называть их бесконечными двоичными дробями, считая дроби вроде 0,10000... и 0,01111... различными.

Таким образом, мы (вслед за Кантором) доказали, что

множество всех бесконечных двоичных дробей несчётно

(это как раз и значит, что их нельзя перенумеровать, ничего не пропустив).

4 Как доказать, что множество всех бесконечных десятичных дробей несчётно?

К таблице с десятичными дробями можно применить то же рассуждение, взяв последовательность цифр по диагонали и заменив каждую цифру на какую-то другую (теперь есть целых 9 вариантов — все цифры, кроме той, что на диагонали). А можно заметить, что если мы пронумеровали все бесконечные десятичные дроби, то среди них есть ведь и все бесконечные двоичные дроби, и можно все остальные вычеркнуть и получить противоречие с уже доказанным.

3. Диагональный метод

Похожие рассуждения оказываются полезными в самых разных задачах. Схему рассуждения можно описать (пусть и несколько расплывчато) так.

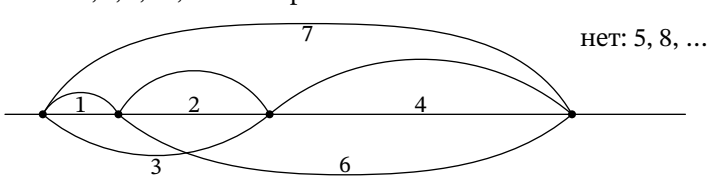
• Мы строим объект (бесконечную последовательность нулей и единиц), который удовлетворяет бесконечному числу условий: i-е условие (назовём его A_i) состоит в том, что строимая последовательность отличается от i-й строки таблицы.

• Объект строится не сразу, а постепенно (мы определяем цифры последовательности по очереди), и i-й шаг гарантирует, что условие A_i будет выполнено, независимо от дальнейшего.

Это, конечно, общая схема, но вот пример, где она помогает.

5 Докажите, что можно расставить на прямой точки таким образом, чтобы расстояние между любой парой (различных) точек было целым положительным числом, и чтобы каждое целое положительное число встречалось только один раз (только для одной пары).

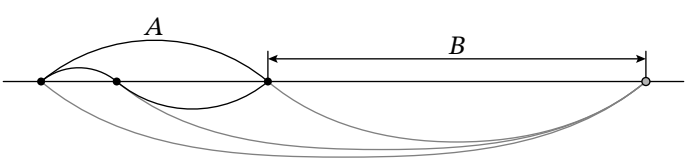
Скажем, в этом примере все попарные расстояния $(1, 2, 3, 4, 6, 7)$ различны — но $5, 8, 9, 10, \ldots$ не встречаются:



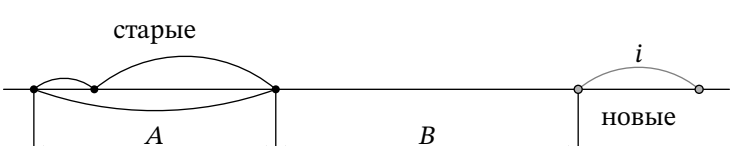
Легко сделать так, чтобы все попарные расстояния были *различными* целыми числами (можно, например, взять точки с координатами $1, 10, 100, 1000, \ldots$) — если не хотеть, чтобы среди расстояний встречались все положительные целые числа. Легко сделать и так, чтобы *все положительные целые числа встречались* среди расстояний (можно взять точки $1, 2, 3, 4, \ldots$) — если не требовать, чтобы все попарные расстояния были разными. Но как совместить оба требования?

Попробуем следовать описанной схеме и строить искомое множество, добавляя точки постепенно. Другими словами, в каждый момент есть конечное число точек, и новые добавляются одна за другой (а старые остаются). Ясно, что нельзя допустить появления двух пар на одном и том же расстоянии, потому что дальше они уже никуда не денутся. Так что мы будем рассматривать только «хорошие» наборы точек: все попарные расстояния целые и разные.

Если мы неудачно добавим точку к хорошему набору, он может стать плохим: может появиться расстояние, которое уже было в другом месте. Этого можно избежать, если выбрать новую точку достаточно далеко от старых — так, чтобы минимальное расстояние между новой точкой и старыми (на рисунке это B) было больше максимального расстояния между старыми (A).



Но нам нужно не просто добавить точку, избегая одинаковых расстояний. Надо, чтобы среди попарных расстояний появилось заданное число i («условие A_i»). Можем ли мы это сделать, добавляя точки (и по-прежнему избегая одинаковых расстояний)?



Можно добавить где-то далеко не одну, а две точки на расстоянии i. Если расстояние между ними и старыми точками взять больше максимального расстояния между старыми ($B > A$ на рисунке), то «новые»

расстояния (между новыми и старыми точками) будут больше всех старых, и появится расстояние i. Единственная проблема — не совпадут ли два «новых» расстояния?

Но это означало бы, что и между какими-то старыми точками расстояние было равно i, а в этом случае ничего добавлять не надо.

Итак, мы показали, что для любой допустимой (без равных расстояний) конфигурации точек и для любого i можно добавить точки так, чтобы гарантировать условие A_i (наличие пары точек на расстоянии i), сохранив допустимость. (Если условие A_i уже было выполнено, то мы «добавляем 0 точек» — ну или можно добавить одну, как мы уже обсуждали.) Теперь остаётся сослаться на «диагональный аргумент» — делая это по очереди для всех i, мы получим требуемую конфигурацию.[4]

4. Ещё несколько примеров

В нашем исходном примере, имея таблицу с бесконечными последовательностями α_i (в строках), мы построили последовательность α, которая отличается от каждой из α_i — отличается *хотя бы в одном месте*. Но можно захотеть большего:

6 Докажите, что найдётся последовательность α, которая отличается от любой из последовательностей α_i *хотя бы в двух местах*.

Теперь условие A_i означает, что (строимая) последовательность α отличается от α_i (при всех i) по крайней мере в двух местах. Построение будем тоже проводить по шагам: на каждом шаге мы дописываем к уже построенной конечной последовательности битов (началу будущей α) ещё несколько битов. А именно, достаточно дописать два бита, выбрав их так, чтобы они не совпадали с соответствующими битами в α_i.

[4] Педанты сказали бы, что ещё надо проверить, что если на каждом шаге нет одинаковых расстояний, то и в «предельной» конфигурации из всех добавленных точек нет одинаковых расстояний. Но это очевидно: нарушение условия включает четыре точки (две пары), и с какого-то момента все эти четыре точки уже присутствуют и нарушают условие.

На рисунке это построение можно изобразить так:

0	0	0	1	0	0	0	0	...
1	1	1	1	1	1	1	1	...
0	1	0	1	0	1	0	1	...
1	0	0	1	0	0	1	0	...
.....								
1	1	0	0	1	0	0	1	...

Таким же способом можно получить последовательность, отличающуюся от всех α_i минимум в трёх местах, или в тысяче мест, — но можно пойти ещё немного дальше.

7 Докажите, что найдётся последовательность α, которая отличается от любой из последовательностей α_i *в бесконечном числе мест*.

Тут явно нужна какая-то другая конструкция: никакой начальный отрезок последовательности α не может гарантировать, что она отличается от α_i в бесконечном числе мест. Надо действовать иначе: разбить все места на бесконечное число бесконечных групп[5] (первую, вторую, ...). На первом шаге мы фиксируем значения α на местах первой группы, ..., на i-м шаге — на местах i-й группы, и делаем это назло α, в том смысле, что α отличается от α_i во всех местах i-й группы. А группа эта бесконечна, так что мы построим требуемую последовательность.

Насколько можно усиливать это утверждение дальше? Можно, к примеру, поставить вопрос так. Будем говорить, что две бесконечные последовательности битов «существенно различны», если для всех достаточно больших N среди первых N мест есть по крайней мере $0{,}49N$ различий.[6]

Можно ли теперь построить последовательность, которая существенно отличается от всех α_i? На самом деле да, хотя непонятно, как тут можно использовать диагональный метод.[7] А именно, можно доказать, что если мы будем бросать честную монету, то вероятность того, что мы получим последовательность, несущественно отличающуюся от α_i, равна нулю для каждого i (это называется «усиленный закон больших чисел»), и потому с единичной вероятностью мы получим последовательность,

[5] Например, первая группа может содержать все нечётные места, вторая — все чётные, не делящиеся на 4, третья — делящиеся на 4, но не на 8, и так далее. Можно придумать и много других способов.

[6] Важно, что эта доля (49%) меньше половины: если мы потребуем, чтобы было больше 50%, то никакая последовательность не сможет существенно отличаться от 00000... и 11111... одновременно.

[7] Мы говорим «непонятно», а не «невозможно», так как наше описание диагонального метода было неформальным.

существенно отличающуюся от всех α_i («счётная аддитивность вероятности»). Но это тема совсем другого рассказа.

8 Докажите, что всякая функция $f : \mathbb{Z} \to \mathbb{Z}$ может быть представлена как сумма трёх биекций $f_1, f_2, f_3 : \mathbb{Z} \to \mathbb{Z}$.

Утверждение этой задачи проще объяснить на картинке:

...	a	...	все по разу
...	b	...	все по разу
...	c	...	все по разу
...	d	...	заданные суммы

Есть таблица из четырёх бесконечных вправо и влево строк (разбитых на клетки). Нижняя строка произвольным образом заполнена целыми числами. (Среди них могут быть одинаковые, а некоторые числа могут вовсе не встречаться — никаких ограничений нет.) Нам нужно заполнить три верхние строки целыми числами. При этом требуется, чтобы:

• в каждом столбце нижнее число было равно сумме трёх чисел над ним (на рисунке $a + b + c = d$);

• в каждой из трёх строк встречались все целые числа по одному разу.

(В исходных терминах f соответствует нижней строке, а f_1, f_2, f_3 — трём верхним; первое условие означает, что $f_1 + f_2 + f_3 = f$, а второе — что функции f_1, f_2, f_3 являются биекциями.)

Будем применять диагональный метод так. В каждый момент у нас будет заполнено конечное число столбцов таблицы. При этом, конечно, надо рассматривать только такие заполнения, где суммы заполненных столбцов правильны и где в каждой строке нет одинаковых чисел (потому что такие нарушения уже не исправить). Какие условия надо выполнить? Нам нужно, чтобы:

• каждый столбец был рано или поздно заполнен;

• каждое число рано или поздно появилось во всех строках.

Допустим, мы хотим, чтобы в одной из строк (скажем, в первой) появилось число i, которого там раньше не было. Это значит, что нам надо подобрать такие числа x и y для двух других строк, чтобы $i+x+y$ равнялось заданной сумме d, и чтобы эти самые x и y в соответствующих строках не встречались.

	*	*	*	*		i	
	*	*	*	*		x	
	*	*	*	*		y	
	*	*	*	*		d	заданные суммы

Первое условие означает, что $x + y = d - i$, то есть нам нужно выбрать два целых числа x и y с известной суммой. Это можно сделать бесконечным числом способов (для любого x можно подобрать соответствующее y, и наоборот). Но не все способы годятся: числа, уже встречавшиеся в строке, брать нельзя. Но пока что лишь конечное число столбцов заполнено, так что эти запреты исключают конечное число вариантов, и поэтому всегда можно их избежать. (Скажем, можно взять x очень большим положительным, а y — отрицательным, с нужной суммой.)

Это построение можно выполнить для любой из строк и в любом свободном столбце, так что можно выполнять наши условия по очереди (диагональный метод) и в итоге заполнить все столбцы правильно. На каждое число i уйдёт максимум три шага (по одному для каждой строки, где оно должно появиться). Ещё надо проследить, чтобы не осталось незаполненных столбцов (это легко, так как столбец можно брать любой).

5. Изоморфизм плотных множеств

В этом разделе мы обсудим другой известный результат Кантора: *два счётных плотных упорядоченных множества без первого и последнего элемента изоморфны*. Чтобы не отвлекаться на определения всех использованных слов, рассмотрим конкретный пример (хотя рассуждение будет общим). А именно, рассмотрим два множества:

- $A =$ рациональные числа (простые дроби) в интервале $(0, 1)$;
- $B =$ двоично-рациональные числа (конечные двоичные дроби, числа вида $k/2^n$) в интервале $(0, 1)$.

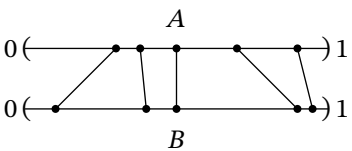
9 Докажите, что между этими двумя множествами можно построить взаимно однозначное соответствие, сохраняющее порядок.

Это означает, что элементы этих множеств можно объединить в пары (рациональное число a, двоично-рациональное число b) таким образом, что каждый из элементов — как множества A, так и множества B — входит ровно в одну пару и что меньший элемент соответствует меньшему (нет пар (a, b) и (a', b') с $a < a'$ и $b > b'$).

Оба множества счётны — это значит, что их элементы можно пронумеровать. Скажем, в A можно сначала перечислить все дроби со знаменателем 2 (такая дробь только одна: 1/2), потом все дроби со знаменателем 3 (их две: 1/3 и 2/3), потом со знаменателем 4 (1/4 и 3/4; мы пропускаем 2/4, так как это число уже было раньше), потом со знаменателем 5, и так далее. Аналогично и для множества B (знаменатели — только степени двойки). Если мы просто запишем эти множества в порядке номеров: $A = \{a_1, a_2, \ldots\}$ и $B = \{b_1, b_2, \ldots\}$ и скажем, что a_i соответствует b_i (объединив a_i и b_i в пару), то получим взаимно однозначное соответствие.

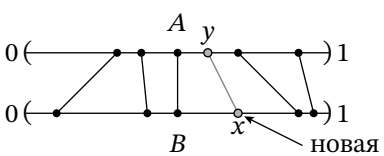
Но вполне может оказаться (а при нашем способе нумерации точно окажется), что это соответствие не сохраняет порядок: может быть $a_i > a_j$ и $b_i < b_j$ одновременно.

Как же построить взаимно однозначное соответствие, сохраняющее порядок? Постепенно, следуя диагональному методу. В каждый момент у нас есть взаимно однозначное соответствие между некоторой конечной частью A и конечной частью B, сохраняющее порядок (меньшие в A соответствуют меньшим в B):



Сохранение порядка означает, что линии, соединяющие пары соответствующих элементов сверху и снизу, не пересекаются друг с другом.

На каждом шаге мы будем добавлять какое-то конечное число пар (=линий, соединяющих точки в A с точками в B), и в итоге получим множество из всех добавленных нами пар. Что нам нужно от этой конструкции? Прежде всего, чтобы не нарушался порядок, и если мы добавляем точку x в интервал между какими-то двумя уже имеющимися, то добавляемая вместе с ней парная точка y должна быть в соответствующем интервале другого множества:



Аналогичным образом, если x оказалась справа от всех имеющихся (или слева), то и y должна быть справа/слева от всех имеющихся точек в её множестве.

Теперь вернёмся к нашей схеме. У нас есть счётное число (последовательность) условий: про каждую верхнюю и про каждую нижнюю точку мы хотим, чтобы она вошла в строимое соответствие. Как мы видели, любое такое условие можно раз и навсегда удовлетворить, добавив одну пару. При этом мы используем то, что между любыми двумя точками — будь то в A или B — всегда можно найти третью (можно взять среднее арифметическое, для рациональных чисел оно рационально, а для двоично-рациональных — двоично-рационально). А также то, что справа от любой точки (и слева тоже) есть ещё другие точки. (Можно взять среднее арифметическое с 1 или 0.)

Эта конструкция позволяет выполнить все условия по очереди и получить взаимно однозначное соответствие, сохраняющее порядок.

Если хочется более явно описать порядок действий, то можно сказать, что мы нумеруем точки в A и B, и действуем сверху вниз (от A к B) на чётных шагах и снизу вверх на нечётных. Действие состоит в том, что мы берём отсутствующую точку (сверху на чётных шагах и снизу на нечётных) с минимальным номером и находим к ней пару, которую можно добавить, не разрушая биекции — любую точку соответствующего интервала. Тогда можно быть уверенным, что после $2k$ шагов (k шагов каждого вида) у нас будут обслужены по крайней мере k первых точек A и k первых точек B. А может, и больше, если какие-то из первых k точек в одном множестве были непредвиденно обслужены во время шагов противоположного направления.

Иногда эту конструкцию называют back-and-forth (туда-сюда), а не диагональным аргументом. Кроме того, в исходной работе Кантора конструкция была несимметричной: он перебирал все точки сверху — и брал в качестве пары точку снизу в нужном интервале, имеющую минимальный номер. (Это гарантирует, что до любой точки снизу дойдёт очередь: если выбирается не она, то число стоящих впереди неё в очереди уменьшается, и это число рано или поздно дойдёт до нуля.) Но не будем спорить о терминологии: важно лишь, что у нас счётное число условий, и что каждое из них можно удовлетворить раз и навсегда конечным расширением монотонной биекции.

6. Теорема Бэра

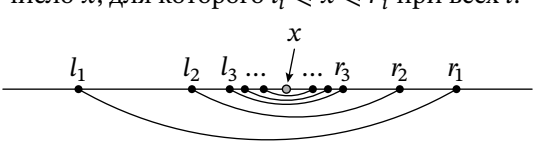
С помощью нашей технологии можно строить не только последовательности и соответствия, но и действительные числа. При этом понадобится *принцип вложенных отрезков*, также традиционно связываемый с именем Кантора (иногда его называют аксиомой Кантора, иногда теоремой — это зависит от выбранного способа изложения). Этот принцип утверждает, что если у нас есть последовательность вложенных отрезков

$$[l_1, r_1] \supset [l_2, r_2] \supset [l_3, r_3] \supset \ldots \supset [l_i, r_i] \supset \ldots,$$

то обязательно есть общая точка (принадлежащая всем отрезкам $[l_i, r_i]$). В терминах неравенств: если числа $l_1, l_2, \ldots$ и $r_1, r_2, \ldots$ таковы, что

$$l_1 \leqslant l_2 \leqslant l_3 \leqslant \ldots, \quad r_1 \geqslant r_2 \geqslant r_3 \geqslant \ldots \quad \text{и} \quad l_i < r_i \text{ при всех } i,$$

то существует число x, для которого $l_i \leqslant x \leqslant r_i$ при всех i.



В диагональных конструкциях этот принцип применяется так: пусть нужно построить число, удовлетворяющее последовательности условий $A_1, A_2, \ldots$; мы строим его постепенно. Сначала находим отрезок $[l_1, r_1]$, все точки которого удовлетворяют условию A_1. Затем внутри этого отрезка находим отрезок $[l_2, r_2]$, все точки которого удовлетворяют условию A_2, и так далее. Если это удастся, то общая точка x всех отрезков по построению будет удовлетворять всем условиям.

Удастся это или нет, конечно, зависит от конкретных условий A_i. Вот пример, когда это удаётся легко.

10 Покажите, что для любой последовательности действительных чисел u_1, u_2, ... существует число, которое не входит в эту последовательность.

В самом деле, у нас есть последовательность условий: A_i требует, чтобы x отличалось от u_i. Надо найти число, удовлетворяющее всем A_i. Первый отрезок возьмём любой, не содержащий числа u_1. (Скажем, можно взять $[u_1 + 1, u_1 + 2]$ — или как-нибудь ещё.) Теперь надо (на i-м шаге) внутри уже имеющегося отрезка $[l_{i-1}, r_{i-1}]$ выбрать отрезок, не содержащий u_i. Это тоже легко сделать разными способами — скажем, можно разделить имеющийся отрезок на три части и заметить, что число u_i может попасть только в одну из них или в крайнем случае в две, если оно попадёт на границу — но не в три.[8]

Если вспомнить, что действительные числа можно отождествить с бесконечными двоичными дробями, то мы получаем новое доказательство несчётности множества всех бесконечных дробей (то, что двум дробям может соответствовать одно число, нам только на руку).

Можно сформулировать и в общем виде, что нам нужно от условий A_i. Будем говорить, что условие A «хорошее», если в любом отрезке[9] можно найти меньший отрезок, все точки которого удовлетворяют условию A. (Топологи сказали бы, что «внутренность A всюду плотна».)

---

[8] Любопытно посмотреть на оригинальное доказательство Кантора 1874 года — русский перевод и ссылку можно найти на с. 18 сборника работ: Георг Кантор, *Труды по теории множеств* (М.: Наука, 1985), который вышел на излёте СССР, несмотря на попытки академика Понтрягина этого не допустить (в автобиографии он писал об этом сборнике: «Теория множеств, очень популярная во времена Лузина, в настоящее время уже утратила актуальность. Моё предложение было принято группой, и книга была отвергнута. Секция с нами согласилась сразу, и это несмотря на то, что перевод сочинений Кантора уже был сделан! Так что [издательству] пришлось его оплатить»). Как часто бывает с первооткрывателями, доказательство излишне запутано — Кантор смотрит, какие из точек u_i попадают в текущий отрезок, без чего мы легко обошлись.

[9] Мы считаем, что все отрезки ненулевой длины: точка — не отрезок.

11 Докажите, что если все условия A_i хорошие, то существует точка, удовлетворяющая одновременно всем условиям A_i.

Собственно, тут нечего и доказывать — мы сформулировали ровно то, что требуется для проведения нашей конструкции.

Это утверждение обычно называют *теоремой Бэра о категории* или просто *теоремой Бэра*[10], потому что слово «категория» здесь имеет совсем не тот смысл, который с тех пор стал общепринятым («теория категорий и функторов»). Оно было опубликовано в самом конце XIX века.

7. Проблема остановки

Диагональные рассуждения часто встречаются в *теории вычислимости*[11] (теории алгоритмов — мы интересуемся тем, что могут и чего не могут сделать алгоритмы). Будем говорить, что последовательность нулей и единиц *вычислима*, если существует алгоритм (программа[12]), который печатает члены этой последовательности по порядку. Мы рассматриваем только программы, которые могут печатать нули и единицы (других операторов вывода у них нет); программа может работать бесконечно. Разумеется, может оказаться, что программа останавливается или с какого-то момента ничего не печатает — тогда получается конечная последовательность. (Их тоже считают вычислимыми, но нас интересуют бесконечные.)

Скажем, можно написать программу, которая вычисляет и печатает один за другим двоичные знаки числа π. Поэтому последовательность цифр числа π вычислима.[13]

Вычислимые последовательности (в отличие от всех) можно записать в таблицу. А именно, мы можем выписывать программы нашего вида в порядке возрастания их длины (для каждой длины есть конечное число программ), и напротив каждой программы писать ту последовательность, которую она вычисляет. (Если последовательность конечна, то можно, скажем, дополнить её нулями, чтобы все последовательности в таблице были бесконечны.)

---

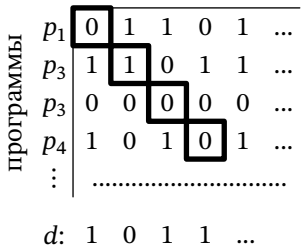
[10] Рене Бэр (René-Louis Baire, 1874–1932), французский математик.

[11] Подробнее про это можно прочесть в учебниках по теории вычислимости (раньше её называли ещё *теорией рекурсии*), скажем, *Вычислимые функции*, М.: МЦНМО, `https://hal.science/lirmm-01486500`.

[12] Мы предполагаем, что читатель знаком с каким-то языком программирования.

[13] Для этого есть эффективные алгоритмы, так что вполне можно успеть вычислить триллионы цифр этого разложения на домашнем компьютере. Но нас интересует только теоретическая вычислимость, независимо от необходимого для этого времени.

Дальше можно взять последовательность, которая стоит на диагонали, и обратить её. Получится последовательность, у которой i-й член противоположен i-му члену i-й последовательности: $d(i) \ne p_i(i)$.



Мы знаем, что этой последовательности не будет в таблице, то есть она не будет вычислима. Но как же так?

12 Что мешает нам вычислить i-й член диагональной последовательности d, найдя i-ю по счёту программу, запустив её, подождав, пока на выходе появятся i битов, и затем взяв противоположный бит?

Мешает «мелочь», о который мы говорили вскользь — что бывают программы, которые так и не доходят до i-го бита. Им соответствуют конечные последовательности. Эти последовательности можно искусственно продолжить, добавив туда нули. Но наше рассуждение показывает, что этого нельзя сделать алгоритмически, иначе получится противоречие. Другими словами, мы сталкиваемся с *алгоритмически неразрешимой проблемой*: сказать по программе, печатающей биты, и по числу i, дойдёт ли когда-нибудь эта программа до i-го бита или нет. Эта проблема (в немного другом варианте[14]) называется в теории алгоритмов *проблемой остановки*, и мы доказали её неразрешимость (предположив, что она разрешима, мы приходим к противоречию).

8. Иммунные множества

Вот ещё один результат из теории вычислимости, где полезна диагональная конструкция. Пусть имеется некоторое множество A натуральных чисел. Будем говорить, что оно *вычислимо бесконечно*, если есть программа, которая печатает бесконечно много его членов. Скажем, множество чётных чисел вычислимо бесконечно, потому что программа может

[14]Проблемой остановки называют обычно такую: дана программа, требуется узнать, остановится ли она или будет работать бесконечно. Если бы эта проблема была алгоритмически разрешима, то была бы разрешима и наша: зная i и p_i, легко переделать программу p_i в другую программу, которая моделирует действия p_i и останавливается, когда p_i печатает i-й бит (а в иных случаях не останавливается).

печатать числа 0, 2, 4, 6, … в порядке возрастания. Мы не запрещаем программе печатать одно и то же число многократно[15] или долго думать над следующим числом, важно только, чтобы она напечатала бесконечно много разных чисел.

Разумеется, всякое вычислимо бесконечное множество будет бесконечным, но верно ли обратное? Может ли быть так, что множество натуральных чисел бесконечно, но нельзя алгоритмически указывать всё новые и новые его элементы? Оказывается, что да (такие множества традиционно называют *иммунными*), и что верно даже более сильное утверждение.

13 Натуральный ряд можно разбить на две части, каждая из которых не будет вычислимо бесконечной.

Разбить на две части — это представить в виде объединения двух непересекающихся множеств A и B. Если описанное в задаче возможно, то обе части будут бесконечны (в самом деле, если A конечно, то B содержит все числа, начиная с некоторого, и их можно перечислять подряд, так что B вычислимо бесконечно).

Как можно это сделать? В каждый момент некоторый начальный отрезок натурального ряда уже распределён между A и B, а про все следующие натуральные числа решение ещё не принято. На следующем шаге ещё сколько-то чисел (хотя бы одно, чтобы мы не застопорились где-то) распределяются между A и B, и так далее.

Нам нужно, чтобы A и B *не были* вычислимо бесконечны. Это можно разбить в последовательность условий: i-е условие говорит, что i-я программа (будем обозначать её p_i) не перечисляет бесконечного подмножества в A, а также не перечисляет бесконечного подмножества в B.

Осталось объяснить, как можно продолжить уже имеющееся распределение начального отрезка и гарантировать выполнение i-го условия. Посмотрим на программу p_i. Может оказаться, что она вообще не перечисляет никакого бесконечного множества (а только конечное). Тогда условие выполнено автоматически. Опаснее, если p_i перечисляет какое-то бесконечное множество X — тогда нам надо гарантировать, что это самое X не является подмножеством ни A, ни B. Раз X бесконечно, а начальный отрезок уже распределённых по A и B чисел конечен, в X есть бесконечно много чисел, не входящих в этот отрезок. Нам достаточно двух из них: договоримся, что одно число помещается в A, а другое помещается в B. (Чтобы соблюсти наше правило о том, что распределён

[15] Но можно было бы и запретить — это не изменит определения. В самом деле, программа может хранить список уже напечатанных чисел и проверять перед каждой печатью, не было ли такого числа раньше.

начальный отрезок, надо ещё что-то решить про все остальные числа, меньшие максимального из этих двух; это можно сделать произвольно.)

Предупреждение: описанный процесс напоминает по форме алгоритм (делаем то-то, если…), но он не является таковым: глядя на программу p_i, мы не можем алгоритмически определить, перечисляет ли она бесконечное множество или нет (на самом деле нам достаточно проверить, перечислит ли она хотя бы два элемента в ещё не фиксированной зоне; это в некотором смысле проще, но всё равно алгоритмически сделать нельзя). Что и не удивительно: если бы процесс был алгоритмическим, то можно было бы перечислять элементы множеств A и B, и они были бы вычислимо бесконечны.

9. Теорема Кантора

Есть другое обобщение рассуждения Кантора, которое уже выходит за рамки построения по шагам, но использует идею диагонали. Мы доказали, что $2^n > n$, показав, что нельзя записать в таблицу из n строк все последовательности из n битов. А также доказали, что нельзя записать в бесконечную таблицу все бесконечные последовательности битов; это иногда записывают как $2^{\aleph_0} > \aleph_0$, где $\aleph_0$ — обозначение для «мощности счётного множества». Можно от конечных множеств перейти к любым, это тоже сделал Кантор (1890, см. упомянутый выше сборник его работ, с. 190). Вот как это делается.

Пусть X — произвольное множество. Будем рассматривать множество всех функций, определённых на (всём) X и принимающих значения 0 и 1. Оно обозначается 2^X, и выбор этого обозначения можно объяснить так. Если X конечно и содержит k элементов, то множество 2^X состоит из 2^k функций: для каждой из k точек в X мы должны решить, равно ли значение функции в ней нулю или единице. Поэтому операцию перехода от X к функциям на X можно по аналогии обозначать как 2^X. Теперь можно сформулировать теорему Кантора.

14 Докажите, что не существует взаимно однозначного соответствия между множеством X и множеством 2^X, каково бы ни было X.

Другими словами, 2^X не *равномощно* X (равномощными считают множества, между которыми есть взаимно однозначное соответствие).

Если X — множество натуральных чисел, то это утверждение как раз говорит, что нельзя пронумеровать все бесконечные последовательности нулей и единиц так, чтобы каждая последовательность имела ровно один номер. И это мы уже видели (антидиагональ будет пропущена). Заметим ещё, что вместо «ровно один» было достаточно предположить, что «хотя

бы один», и уже получалось противоречие. Это же верно и для любого множества X.

Итак, предположим, что для каждого x выбрана некоторая функция $f_x : X \to \{0,1\}$, и покажем, что существует функция $f : X \to \{0,1\}$, которая не встречается среди f_x. А именно, положим $f(x)$ равным $1-f_x(x)$, и тогда f отличается от f_x на аргументе x. Таким образом, f не равна f_x (отличия в одной точке уже достаточно) при любом x, так что f искомая.

Можно вместо функций $f : X \to \{0,1\}$ рассматривать подмножества X, считая, что каждой функции соответствует подмножество тех элементов, где она равна единице. Множество всех подмножеств множества X обозначают $\mathcal{P}(X)$. Теорему Кантора можно переформулировать на этом языке так:

15 Докажите, что $\mathcal{P}(X)$ не равномощно X, каково бы ни было X.

Это прямое следствие предыдущей формулировки (потому что между 2^X и $\mathcal{P}(X)$ существует взаимно однозначное соответствие), но можно перевести и доказательство: если для каждого $x \in X$ выбрано какое-то подмножество $P_x \subset X$, то существует подмножество, которое пропущено, а именно $D = \{x \mid x \notin P_x\}$. В самом деле, D отличается от любого множества P_x в точке x (потому что $x \in D$ в том и только том случае, когда $x \notin P_x$ — таково определение D).

Бертран Рассел[16] заметил, что если в качестве X взять множество всех множеств (представим себе, что такое бывает!), то $\mathcal{P}(X)$ будет подмножеством X, так как тоже состоит из множеств, и теорема Кантора приводит к противоречию: вспомнив её доказательство, мы берём множество D всех множеств x, для которых $x \notin x$, и замечаем, что $D \in D$ тогда и только тогда, когда $D \notin D$ (таково определение D). Это наблюдение — знаменитый *парадокс Рассела*.

10. Трансфинитная индукция

Было бы обидно не рассказать о, пожалуй, самом поразительном открытии Кантора, которое получило название *трансфинитной индукции*. Обычная *математическая индукция* — это когда мы доказываем или строим[17] что-то последовательно для всех натуральных чисел. Мы занимались этим, строя искомый объект и по очереди выполняя счётное число (пронумерованных натуральными числами) условий. Но можно ли

[16] Bertrand Russell (1872–1970), английский логик, философ, лауреат Нобелевской премии по литературе.

[17] Для построений иногда употребляется термин *рекурсия*, чтобы отличать их от доказательств.

как-то обобщить эту технику на несчётные множества, элементы которых нельзя пронумеровать и рассматривать по очереди?

Вот, скажем, мы для любой функции $f : \mathbb{Z} \to \mathbb{Z}$ строили её представление в виде суммы трёх биекций $f_1, f_2, f_3 : \mathbb{Z} \to \mathbb{Z}$. Мы действовали так: если в какой-то точке f_1, f_2, f_3 ещё не определены, можно подобрать им такие значения, чтобы они давали нужную сумму и ещё не встречались — при этом одно из трёх значений можно выбрать произвольно. Так, одно за другим, мы добавляли целые числа в область определения и в область значений функций f_1, f_2, f_3, и в итоге получали искомые функции. А можно ли такое сделать, скажем, для функции $f : \mathbb{R} \to \mathbb{R}$? Всегда ли такую функцию можно представить в виде суммы трёх биекций $f_1, f_2, f_3 : \mathbb{R} \to \mathbb{R}$?

В общем, да[18] — и Кантор придумал, как это сделать. Начать можно так же, как и раньше: будем по очереди рассматривать какие-то действительные числа $x_1, x_2, x_3, x_4, \ldots$ и постепенно определять функции f_1, f_2, f_3 с таким расчётом, чтобы:

- текущая область определения функций f_1, f_2, f_3 была бы одинаковой;
- для всех x из этой области определения сумма $f_1(x) + f_2(x) + f_3(x)$ равнялась бы $f(x)$;
- все значения функции f_1 были бы различны, то же для f_2 и f_3;
- после i-го шага число x_i было бы и в области определения функций f_1, f_2, f_3, и среди значений каждой из этих функций.

Мы уже видели, как это делается. Для каждого i появятся максимум четыре новых столбца. Нам надо, чтобы функции были определены в x_i, на это уйдёт один столбец. Ещё надо, чтобы каждая из них принимала значение x_i, на это могут понадобиться ещё три. (Можно один столбец сэкономить, используя его для двух целей.) Уже имеющиеся значения не меняются, так что в результате получается функция, определённая на всех $x_1, x_2, x_3, x_4, \ldots$

Но беда в том, что числами $x_1, x_2, x_3, x_4, \ldots$ не исчерпываются все действительные числа. (Если бы у нас вместо действительных были рациональные, то проблем бы не было и доказательство было бы завершено.) Кантора это не остановило — почему бы не рассмотреть «трансфинитное число» ω, которое идёт после всех натуральных чисел, выбрать действительное число x_ω, которого нет среди $x_1, x_2, x_3, x_4, \ldots$ и повторить все рассуждения для него? (При выборе пары чисел с нужной суммой у нас уже будет счётное число запрещённых вариантов, но всего чисел несчётно

[18] Такая осторожная формулировка связана с тем, что эти построения используют так называемую «аксиому выбора» и выглядят странно — так что первое время многие математики ворчали, что это не настоящее математическое доказательство, а какая-то фикция и даже шарлатанство.

много, так что по-прежнему какие-то возможные пары останутся.) И на этом действительные числа не кончатся — выберем следующее трансфинитное число (его логично обозначить $\omega + 1$), какое-нибудь ещё не обработанное действительное число $x_{\omega+1}$ и повторим наше построение. То же самое можно сделать для трансфинитных чисел $\omega + 2$, $\omega + 3$, ... — и снова действительные числа не кончатся. Поэтому можно рассмотреть следующее трансфинитное число $\omega + \omega$ (иначе обозначаемое $\omega \cdot 2$), затем $\omega \cdot 2 + 1$, $\omega \cdot 2 + 2$, ..., затем $\omega \cdot 2 + \omega = \omega \cdot 3$, затем (после счётного числа шагов) $\omega \cdot 4$, потом $\omega \cdot 5$, ..., наконец $\omega \cdot \omega$ (иначе обозначаемое ω^2), потом $\omega^2 + 1$ — «и так далее, пока все действительные числа не кончатся и мы не построим искомые биекции».

Слова «и так далее...» выглядят запредельной наглостью. Можно представить себе ещё несколько шагов этого процесса — дойти до $\omega^2 + \omega$, потом $\omega^2 + \omega \cdot 2$, потом $\omega^2 + \omega^2 = \omega^2 \cdot 2$, потом $\omega^2 \cdot 3$, потом $\omega^2 \cdot \omega = \omega^3$, потом ω^4, ω^5, ..., потом число, которое называют ω^ω, «и так далее» — но совершенно не видно, как можно так исчерпать несчётное множество всех действительных чисел (пока что у нас только счётное число номеров).

Тем не менее всему сказанному можно придать точный смысл. Вместо натуральных чисел и индукции по ним Кантор предложил рассматривать *вполне упорядоченные множества*, в которых, как и в натуральных числах, задан порядок (для любой пары элементов известно, какой больше, и из $x < y < z$ следует $x < z$), и этот порядок полный (во всяком непустом подмножестве имеется минимальный элемент). В таких множествах действует принцип наименьшего числа (если какое-то свойство верно не для всех элементов, то существует наименьший, для которого оно неверно) — что позволяет доказывать утверждения по (трансфинитной) индукции. Несколько сложнее можно обосновать и построения с помощью (трансфинитной) рекурсии наподобие нашего. Наконец, существует теорема Цермело — если мы для какого-то множества X имеем «функцию выбора» φ, сопоставляющую с любым его непустым подмножеством Y некоторый элемент $y \in Y$, то описанная нами конструкция (где мы выбираем следующий элемент, применяя φ к множеству оставшихся) может быть доведена до доказательства существования полного порядка в X. (Откуда взять функцию выбора — другой вопрос, но её существование можно объявить «аксиомой выбора».) С помощью теоремы Цермело можно получить достаточно много трансфинитных чисел, чтобы номеров хватило на все действительные числа, и завершить рассуждение.[19]

[19]Сказанное является скорее рекламой великого открытия Кантора, чем его изложением. Подробно можно прочесть, например, в книжке *Начала теории множеств*, М.: МЦНМО, `https://hal-lirmm.ccsd.cnrs.fr/lirmm-03059731`. Конкретно задачи про три биекции там нет, поэтому скажем коротко, как проводится

Ещё одно любопытное утверждение, где полезна трансфинитная индукция: существует множество на плоскости, которое пересекается с любой прямой ровно в двух точках.

11. Теорема Гёделя о полноте

Раз уж мы перешли от математических рассуждений к рекламным разговорам, то скажем ещё об одной конструкции, в которой условия удовлетворяются по очереди — доказательстве *теоремы Гёделя о полноте*[20]. Эта теорема утверждает, что *всякая непротиворечивая теория имеет модель*. Совсем приблизительно, теория — это набор утверждений, а модель — мир, в котором эти утверждения верны. Теория непротиворечива, если, рассуждая логически и приняв утверждения теории, мы не можем прийти к противоречию.

Давайте представлять себе так: теория — это нечто вроде романа, в котором героев как-то зовут, что-то про них утверждается и т. п. Не то чтобы все романы были непротиворечивы[21], но мы будем это предполагать. А модель — это что-то типа виртуальной реальности, в которую мы можем погрузиться и видеть всё, что происходит с героями в каждый момент, где они находятся, с кем встречаются, что делают эти люди — в общем, целый мир, о котором мы знаем всё. И, конечно, требуется, чтобы все утверждения романа в этом мире были верны.

Начав строить модель непротиворечивого романа, мы немедленно сталкиваемся с тем, что он *неполон* в том смысле, что про некоторые утверждения не говорится, верны они или нет. Скажем, из *Евгения Оне-*

соответствующее рассуждение. Возьмём наименьший ординал α мощности континуум, и пронумеруем все действительные числа меньшими ординалами β. Затем для каждого $\beta < \alpha$ мы строим инъекции f_1^β, f_2^β, f_3^β с общей областью определения мощности меньше континуума, которые продолжают все предыдущие f_i^γ (для $\gamma < \beta$), определены на числе номер β и содержат это число среди своих значений, причём новых значений (тех, которых не было в f_i^γ для $\gamma < \beta$) там конечное число (на самом деле — не более четырёх). Для этого мы объединяем все предыдущие биекции и добавляем, если надо, ещё числа (не более четырёх) в область определения, как мы описывали. То, что найдутся ещё не использованные числа, следует из того, что α — наименьший ординал мощности континуум. В самом деле, на шаге β занятых чисел будет не сильно больше, чем ординалов, меньших β (занятых конечное множество для каждого ординала), то есть не больше β, что меньше континуума, так как α было наименьшим ординалом мощности континуум. (Этот существенный шаг — выбор наименьшего ординала α мощности континуум — в нашем неформальном описании совсем отсутствовал.)

[20]Её не надо смешивать с более известной *теоремой Гёделя о неполноте*.

[21]Владимир Андреевич Успенский обратил внимание на то, что в *Докторе Живаго* один из двух сыновей Марфы Гавриловны Тиверзиной, Киприан Савельевич, и женат, и холост.

гина, видимо, нельзя узнать, был ли Зарецкий знаком с Истоминой (хотя они упоминаются в романе — значит, должны присутствовать в виртуальной реальности, где мы можем за ними проследить и узнать, знакомы они или нет).

Поэтому при переходе от теории к модели нужно её пополнить: *полнота* теории означает, что для любого утверждения об объектах теории либо само это утверждение, либо его отрицание логически следует из теории. Это самое пополнение как раз и происходит по нашей схеме — в результате постепенного выполнения последовательности условий. А именно, мы для каждого утверждения S в языке нашей теории формулируем условие A_S: «либо из теории логически выводится S, либо из неё следует, что S неверно». (Одновременно этого быть не может, если теория непротиворечива.) Пополнение (с сохранением непротиворечивости) происходит по шагам, на каждом шаге мы добиваемся выполнения одного из условий A_S. Это делается так: если из теории уже выводится S или уже выводится отрицание S, то ничего делать не надо. Если же ни то и ни другое, то к теории можно добавить на выбор или S, или его отрицание — и то, и другое по отдельности оставит теорию непротиворечивой. (В самом деле, если, скажем, отрицание S приводит к противоречию, то тем самым S доказывается в теории «от противного».) Делая это для всех условий A_S по очереди (если есть две альтернативы, можно выбирать любую), мы получим искомое пополнение.

При построении искомой модели (виртуальной реальности) есть и другая проблема: из сказанного в романе может следовать утверждение о существовании чего-то, но неконкретное — допустим, сказано, что к герою приходил кто-то, но не сказано кто. Ясно, что при построении виртуальной реальности это надо уточнить. Формально говоря, теорию нужно сделать «экзистенциально полной» — если она утверждает, что существует объект с таким-то свойством, то этот объект должен иметь имя.[22] (После этого её снова надо пополнять, и эти операции чередуются.)

Вероятно, эти объяснения уж совсем расплывчаты — так что стоит их воспринимать как рекламу (и теоремы Гёделя о полноте[23], и диагональных конструкций).

Автор благодарен Ивану Яковлеву за возможность рассказать школьникам про диагональный метод, а также Татьяне Коробковой и Виктору Шувалову за советы по улучшению текста.

[22]Если бы *Евгений Онегин* остановился бы на словах «Так ты женат! Не знал я ране!», и Пушкин не ответил бы сам на вопрос «На ком?», то на этот вопрос пришлось бы отвечать нам при экзистенциальном пополнении.

[23]Про неё можно прочесть, например, в книге *Языки и исчисления*, М.: МЦНМО, `https://hal.science/lirmm-01486497v1` .

Содержание

ISBN 978-5-4439-1727-6