

---

# Конструктивная математика: обзор достижений, недостатков и уроков. Часть I

Н. Н. НЕПЕЙВОДА

---

**ABSTRACT.** The whole manifold of constructivist conceptions of last century is analyzed. They are systematized and some aspects and lessons for naive attempts of applying are highlighted.

*Ключевые слова:* конструктивизм, реализуемость, интерпретация Колмогорова, топологические интерпретации, аппроксимации, некорректные задачи, принцип конечной информации

## 1 Введение

XX век в математике характерен появлением неклассических альтернатив традиционной математике как целому. Их мы будем называть общим термином «конструктивизм», поскольку они базировались на желании давать не просто рассуждения, а конструкции, в каком-то смысле реализуемые. Поэтому и различные формы конструктивизма, и интуиционизм на равных фигурируют в данном анализе. Они переживали и расцвет, и надломы, и золотую осень, а ныне находятся, по терминологии Л.Н. Гумилева, на мемориальной стадии. Нынешние работы о конструктивизме либо используют некоторые элементы конструктивной логики (часто примитивно и неточно), либо глубоко и точно описывают, насколько великими были люди и достижения эпохи расцвета. Рассмотрим, какие же уроки можно извлечь из достижений и неудач конструктивизма.

В связи с поставленной задачей в данной работе исторические замечания соединяются с анализом достигнутых результатов и переработкой их в таком стиле, чтобы прояснить их возможные связи с нынешней теорией и практикой. Во многих случаях это требует серьезной переработки материала, который излагался

в соответствии с духом и предрассудками того времени, чтобы приблизить его к духу и предрассудкам нашего. И, конечно же, в отдельных случаях такая переработка перерастала в новое исследование.

Сразу же заметим, что ветераны конструктивизма могут возражать против данного изложения материала, поскольку во многих обоснованиях и модельных конструкциях существенно используются классические методы высокого уровня. На это стоит заранее дать три ответа.

Сам Брауэр не возражал против классической математики, особенно после появления программы Гильберта. Более того, Брауэр эту программу поддержал, но так, что Гильберт возмутился еще больше. Брауэр лишь потребовал, чтобы «практические математики» перестали воображать, что их конструкции чему-то соответствуют в реальности. Эти абстрактные конструкции тем не менее могут признаваться как полезное интеллектуальное орудие. Более того, Брауэр чувствовал, что многие полезные реальные результаты могут быть получены лишь путем прохода через идеальное, поскольку ползучие прямые пути к ним слишком длинны.

Н.А. Шанин строго обосновал, что во многих случаях после построения объекта показать правильность проведенного построения можно и классическими методами без потери конструктивности.

И, наконец, сам автор рассматривает данную работу как выполнение завещанной ему А.А. Марковым задачи: поставить четвертый стул между классикой, конструктивизмом и интуиционизмом и так, чтобы он стоял крепко и всех их соединял. Здесь нужен не эклектизм, а осознанное многоуровневое мышление, когда в каждый момент используются по возможности наиболее подходящие средства, и полученные с их помощью результаты анализируются не с точки зрения корректности обоснования в другом сегменте рассуждений, ставящем перед собой уже совершенно иные цели, а с точки зрения возможности их применения там как (говоря терминами информатики) отдельного модуля с закрытым кодом и открытым интерфейсом. То, что допустимо в одном модуле, может быть абсолютно неприемлемо в другом. Разнородные концепции можно соединять, но лишь на уровне передачи результатов.

## 2 Соглашения об обозначениях

В данной работе используются принятые в современной теоретической информатике  $\lambda$ -обозначения для функций. Это полностью устраняет, в частности, путаницу и нестрогость, возникающие при использовании традиционных обозначений, когда  $x^2$  может означать как конкретное значение, так и функцию преобразования  $x$  в  $x^2$ . Функция записывается как  $\lambda x. x^2$ .

Соответственно применение функции к аргументам записывается как список, в котором первым членом является функция, а последующими — ее аргументы. Во многих тонких случаях эта запись удобнее, чем традиционная математическая запись типа  $\mu(X)$ . Во избежание путаницы с применением функции, обычный список обозначаем угловыми скобками  $\langle a_1 a_2 \dots a_n \rangle$ .

Поскольку используются значения самых разных типов, включая только абстрактные, в формулах мы будем по хорошей традиции информатиков специфицировать все переменные их типами, за исключением того случая, когда *явно* высказано какое-то общее соглашение о типах переменных, действующее в данном фрагменте текста. Например, запись

$$\forall X : \text{Set} \exists n : \text{Nat} A(X, n)$$

означает, что  $X$  — переменная для множеств, а  $n$  — для натуральных чисел<sup>1</sup>. Используются следующие стандартные типы данных и конструкции новых типов.

$\text{Set}$  — тип всех множеств.

$\text{Nat}$  — тип натуральных чисел.

$\text{Int}$  — тип целых чисел.

$\text{Real}$  — тип действительных чисел.

$\text{Set}(\text{Type})$  — тип множеств элементов типа  $\text{Type}$ .

$\langle \text{Type}_1, \dots, \text{Type}_n \rangle$  — тип  $n$ -ок, члены которых перечисленных типов.

$(\text{Type}_1 \rightarrow \text{Type}_2)$  — тип всюду определенных функций из значений первого типа в значения второго типа.

---

<sup>1</sup>В информатике уже твердо осознано, что типы данных — не множества. Здесь мы в соответствующих местах приведем аргументацию, почему их нельзя отождествлять и в конструктивной математике и почему нельзя считать, что каждый тип данных описывается предикатом.

$(\text{Type}_1 \rightarrow \text{Type}_2)$  — тип частичных функций из значений первого типа в значения второго типа.

Поскольку в конструктивизме, как и в информатике, нет возможности обойти использование частичных функций при помощи известного трюка: функция, которая не определена в данной точке, выдает в ней значение  $\text{Error}$ , — постольку явно вводится квазипредикат осмысленности выражения.  $!(f\ t)$  означает, что данное выражение имеет значение. Отметим еще один тонкий момент. Это не всегда означает, что  $!t$ , поскольку  $f$  при вычислениях может просто игнорировать свой аргумент. Еще чаще такое происходит с функциями многих аргументов. Логические предикаты считаются всегда определенными и существенно использующими все свои аргументы. Логические же связки, следуя Клини, определим так, что они могут игнорировать ошибочность некоторых своих аргументов.

$A \vee B$  считается осмысленным и обоснованным, если один из аргументов осмыслен и обоснован. Второй в данном случае игнорируется.

$A \Rightarrow B$  считается осмысленным, если  $B$  осмысленно во всех случаях, когда верно  $A$ . И, наоборот, если  $B$  уже доказано, то  $A$  нас не интересует ни с какой точки зрения.

$A \& B$  считается осмысленным и опровергнутым, если осмыслен и опровергнут один из членов.

$\exists x A(x)$  считается осмысленным и обоснованным, если  $A(t)$  обосновано для некоторого  $t$ , такого, что  $!t$ .

$\forall x A(x)$  считается осмысленным и опровергнутым, если  $A(t)$  опровергнуто для некоторого  $t$ , такого, что  $!t$ .

Мы вообще не задаемся схоластическим вопросом, каково же значение  $P(t)$ , если  $t$  не определено. Формула просто семантически некорректна, если такая ее подформула оказалась в контексте, где она не игнорируется.

И последнее, но одно из самых важных, замечаний.

*Мы не отождествляем алгоритмичность и эффективную вычислимость. Алгоритмические преобразования — один из видов эффективной вычислимости.*

Здесь мы следуем Брауэру, аргументы которого и их математический и методологический анализ приведены ниже.

### 3 Предпосылки конструктивизма

Мы не будем рассматривать реверансы в сторону конструктивности в трудах, скажем, братьев Грассманов и Кронекера. Они не уловили важнейшего аспекта, который делает конструктивизм конструктивизмом и который имел смелость осознать и обнаружить Л. Э. Я. Брауэр [1, 2, 3].

В математике конца XIX века было осознано, что некоторые «теоремы существования» доказываются настолько грязно, что они не дают и, видимо, в принципе не могут дать никакого построения объекта, существование которого утверждается.

**ПРИМЕР 1.** Грязной, в частности, является теорема о существовании неизмеримого множества действительных чисел при любом определении частичной функции *меры*  $\mu$  из множеств действительных чисел в неотрицательные действительные числа, обладающем следующими базовыми свойствами.

1. Мера отрезка  $[0, 1]$  равна 1:  $(\mu [0, 1]) = 1$ .
2. Мера множества из одной точки равна 0:  $(\mu \{a\}) = 0$ .
3. Мера множества  $Y$ , получающегося из множества  $X$  линейным преобразованием  $\lambda x. a \cdot x + b$ , вычисляется как  $(\mu Y) = (\mu X) \cdot a$ .
4. Если  $X_i$  — последовательность непересекающихся множеств, мера каждого из которых определена, то определена и мера их объединения

$$(\mu \bigcup_{i=1}^{\infty} X_i) = \sum_{i=1}^{\infty} (\mu X_i).$$

5. Если определены  $(\mu X)$  и  $(\mu Y)$ , то определены  $(\mu X \cap Y)$  и  $(\mu X \setminus Y)$ .
6. Если  $X \subset Y$  и оба множества измеримы, то  $(\mu X) \leq (\mu Y)$ .

Прежде всего покажем, что измеримое подмножество отрезка  $[0, 1)$  можно циклически сдвинуть внутри этого отрезка и мера его при этом не изменится. Пусть дано измеримое множество  $X$ .

Пусть  $1 > a \geq 0$ . Линейным преобразованием  $\lambda x. x + a$  сдвинем  $X$  на  $a$ . Получившееся множество обозначим  $X_a$ . Согласно свойству 3,  $(\mu X) = (\mu X_a)$ . Теперь поделим множество  $X_a$  на два подмножества:  $X_a^< = X_a \cap [0, 1)$ ;  $X_a^> = X_a \setminus [0, 1)$ . Тогда результат циклического сдвига  $X$  на  $a$  есть  $X_a^< \cup (\lambda x. x + a - 1 X_a^>)$ . Обозначим его  $X \uparrow a$ . Согласно 5,  $(\mu X_a^>) + (\mu X_a^<) = (\mu X)$ . Но  $(\mu (\lambda x. x + a - 1 X_a^>)) = (\mu X_a^>)$ .  $(\mu X \uparrow a) = (\mu X)$ .

Теперь рассмотрим отношение эквивалентности

$$\simeq = \{ \langle x y \rangle \mid (x - y) \in \mathbb{Q} \}.$$

Здесь  $\mathbb{Q}$  — множество рациональных чисел. В каждом классе эквивалентности чисел из  $[0, 1)$  выберем по элементу. Множество всех получившихся элементов обозначим  $P$ . Если  $P$  циклически сдвинуть на рациональное число  $a$ , то  $P \uparrow a \cap P = \emptyset$ . Далее,

$$\bigcup_{a \in \mathbb{Q} \& a \geq 0 \& a < 1} P \uparrow a = [0, 1).$$

Если предположить, что  $P$  имеет меру, то эта мера либо равна 0, либо равна некоторому  $\varepsilon > 0^2$ .

Тогда в первом случае получаем

$$(\mu[0, 1)) = \sum_{a \in \mathbb{Q} \& a \geq 0 \& a < 1} (\mu P \uparrow a) = \sum_{a \in \mathbb{Q} \& a \geq 0 \& a < 1} 0 = 0,$$

а во втором

$$(\mu[0, 1)) = \sum_{a \in \mathbb{Q} \& a \geq 0 \& a < 1} (\mu P \uparrow a) = \sum_{a \in \mathbb{Q} \& a \geq 0 \& a < 1} \varepsilon = +\infty,$$

но и то, и другое абсурдно.

### Конец примера.

Проанализируем данный пример, поскольку он достаточно показателен. Прежде всего, исключительно грязная сущность данной теоремы была понятна почти сразу после ее доказательства. Математики стали искать явные примеры неизмеримых множеств, но найти таковые никак не удавалось. Лишь в 1930 г.

<sup>2</sup>Из последнего свойства меры следует, что  $\varepsilon \leq 1$ , но в дальнейшем рассуждении это нам не понадобится.

Н.Н. Лузин построил такое явно определенное множество, в отношении которого никак не удавалось доказать его измеримость либо неизмеримость, но это уже было гораздо позже.

Далее, в доказательстве теоремы используется *аксиома выбора*, которая, как только она была явно осознана, оказалась замешанной практически во всех подобных «грязных» доказательствах. Сформулируем ее.

Если у нас есть функция  $\Phi$ , перерабатывающая каждое  $\iota \in I$  в непустое множество  $X_\iota$ , то есть и функция  $\varphi$ , которая выбирает по одному элементу из каждого такого множества:  $(\varphi \iota) \in (\Phi \iota)$ .

Если множества  $X_\iota$  состоят из элементов достаточно хорошо определенного типа данных, в котором, например, все элементы можно каким-то стандартным способом упорядочить так, чтобы в каждом непустом подмножестве был наименьший элемент (*вполне упорядочение*), то такую функцию выбора можно просто построить, находя по каждому  $X_\iota$  его минимум.

Неудивительно, что аксиома выбора оказалась эквивалентной «теореме» Кантора о возможности вполне упорядочить любое множество. А Пуанкаре<sup>3</sup>, в частности, увидев роль этой аксиомы во многих грязных теоремах существования, предложил обходиться без нее. До сих пор большинство математиков «традиционной ориентации» уверены, что именно аксиома выбора ответственна за грязные теоремы существования.

Более того, в 1968 г. Соловей строго доказал, что в стандартной теории множеств **ZF** невозможно явно построить ни одного неизмеримого множества. А чуть позже он совместно с другими логиками доказал, что лужинское множество таково, что для него измеримость невозможно ни доказать, ни опровергнуть.

Но почему же без аксиомы выбора не обошлись? А потому, что она же «ответственна» и за многие исключительно красивые и концептуально важные результаты. Например, за теорему полноты классической логики. Без нее не построишь моделей нестандартного анализа, где можно корректно работать с бесконечно малыми и бесконечно большими. И даже столь простое и очевидное на первый взгляд утверждение, что декартово произведение любого семейства непустых множеств непусто, оказывается эквивалентно аксиоме выбора!

<sup>3</sup>Один из тех, кто впервые явно осознал аксиому выбора.

Наряду с прекрасными и концептуально богатыми результатами из аксиомы выбора получается целый ряд парадоксальных и в чем-то почти сумасшедших следствий. Например, шар можно разделить на четыре множества и одну точку таким образом, что конгруэнтными преобразованиями из них можно составить два таких же шара. Очевидно, что эти множества неизмеримы и построить их также нельзя.

Есть и еще один фактор. Просто отказаться от положения, у которого столько интересных и концептуально важных следствий, полностью бесперспективно. Нужно что-то предложить взамен. И были предложены целых два варианта.

Первый вариант основывается на парадоксальном результате (одном из многих, которые получаются при помощи сильной и интересной аксиомы). Рассмотрим следующую бесконечную игру с полной информацией, идея которой принадлежит польским математикам. Пусть дано некоторое подмножество  $X$  отрезка  $[0, 1]$ . Оно заранее известно обоим игрокам. Далее оба игрока по очереди называют двоичные цифры, начиная со старшего разряда дроби. После конца времен подводится итог. Если полученное действительное число попало в  $X$ , то выигрывает первый игрок. Если нет, то второй. Оказывается, при помощи аксиомы выбора можно «построить»  $X$  такое, что ни один из игроков не имеет выигрывающей стратегии. Поскольку наличие «существования» в любой антагонистической игре с полной информацией выигрывающей стратегии для одного из игроков «интуитивно очевидно», появился интересный кандидат на альтернативу аксиоме выбора: аксиома детерминированности, которая именно это и утверждает. В теории множеств с аксиомой детерминированности, в частности, любое множество действительных чисел измеримо.

Но гони черта в дверь, он влезет в окно и приведет с собой семь сотоварищей. В альтернативной теории множеств была найдена игра, в которой *оба* игрока имеют выигрывающую стратегию. А грязные теоремы в ней отнюдь не исчезли.

Вторую альтернативу, базируясь на прекрасной идее Б. Рассела, предложил К. Гёдель. Б. Рассел для избежания парадоксов теории множеств предложил два взаимосвязанных решения. Одно из них стало традиционным в информатике, а именно тре-



бование строго соблюдать *типы* данных. Второе не стало столь популярным. Оно заключается в том, что даже внутри одного типа данных, содержащего абстрактные объекты, скажем, множества, нужно следить за тем, чтобы каждое из появляющихся множеств было бы определимо через предшествующие. К. Гёдель показал, что в бестиповой теории множеств такие множества<sup>4</sup> образуют модель теории множеств. Но в этой модели аксиома выбора как раз и оказалась выполнена, даже если ее не было в исходной модели.

Таким образом, в начале XX века стало осознаваться всеми мыслящими<sup>5</sup> математиками, что положение с математическими построениями весьма плачевно и что с этим нужно что-то делать.

#### 4 Появление конструктивизма

Конструктивизм начался с блестящего интуитивного прозрения голландского аспиранта Л. Э. Я. Брауэра, который осознал, что причина и грязных, и просто неконструктивных «теорем существования» не в какой-то конкретной аксиоме, а в самой логике. Классическая логика изначально создавалась для обоснований и опровержений, а не для построений.

В конструктивной логике основной задачей является нахождение *идеальных умственных построений*. Три слова, которые выделены, являются цитатой из Брауэра и поражают своей точностью. Мы не обязательно гонимся за тем, чтобы конструкция, которую мы создали, была бы полностью применима на практике и, более того, даже не ставим перед собой цели, чтобы наша теория хотя бы в принципе давала лишь практически реализуемые конструкции. В теории нам нужны и *чистые* теоремы существования, когда объект построен, но практического способа реализовать найденное построение нет. Именно ориентация на построения в принципе, а не в реальности, позволила конструктивизму разработать ряд красивых систем, имеющих большее

---

<sup>4</sup>Гёдель назвал их конструктивными и имел некоторое основание для этого. Но мы не используем данный термин, поскольку к конструктивизму они никакого отношения не имеют.

<sup>5</sup>Но не всеми лихо доказывающими теоремы, больше ничем, кроме непосредственного успеха, не интересующимися и поэтому гордо называющими себя «практическими».

отношение к реальности, чем те, которые с самого начала ограничивали себя ориентацией на непосредственную применимость.

Таким образом, в конструктивной логике утверждение об «истинности теоремы» полностью теряет смысл. Теорема должна быть *реализуема* в том смысле, что ее доказательство должно содержать построение всех тех объектов, которые строятся в теореме, и на базе лишь той информации, которая содержится в условиях теоремы. Тем самым математические утверждения, которые выглядят одинаково, могут нести разный смысл в классической и конструктивной математике.

ПРИМЕР 2. Утверждение

$$\forall X : \text{Set} \exists n : \text{Natural} A(X, n)$$

в классической математике означает истинность<sup>6</sup>

$$\exists n : \text{Natural} A(X_0, n)$$

при каждом данном  $X_0$ , рассматриваемом *по отдельности*. В конструктивной же математике та же формула означает существование общего эффективного метода построения числа  $n$  по каждому множеству  $X_0$ .

Брауэр, проанализировав классическую логику, установил, что основное препятствие к реализуемости — сильный закон исключенного третьего  $A \vee \neg A$ . С конструктивной точки зрения он читается «Мы знаем все». Естественно, тогда никакой нужды строить что-то нет. Если мы доказали, что нечто существует, то оно действительно существует в некоем идеальном математическом мире. Такой взгляд на мир, являющийся грубой вульгаризацией взглядов Платона, принято называть *математическим платонизмом*.

Стоит заметить, что сам Аристотель никогда не утверждал закон исключенного третьего в данной форме. Он приводил в качестве контрпримера к сильному закону исключенного третьего высказывание «Завтра будет морское сражение», которое сегодня не истинно и не ложно. Аристотель формулировал этот закон в более мягкой форме: « $A$  и  $\neg A$  не могут быть одновременно ложны». Эту форму можно записать в качестве логической

<sup>6</sup>Вопрос о том, что есть истина, мы разбирать здесь не будем.

формулы  $\neg\neg(A \vee \neg A)$ . Данная логическая формула, как показал сам Брауэр, выполнена в конструктивной интуиционистской логике Брауэра<sup>7</sup>.

Тем самым Брауэр не вводил никаких новых логических значений. Он признавал непознанное и даже непознаваемое, но не считал это равноправной с истинностью или ложностью характеристикой высказывания. В частности, легко видеть, что с течением времени непознанное может стать познанным, а степеней непознаваемости много (начиная со знаменитого высказывания оракула «Познай самого себя», что невозможно для сложной системы). И даже принципиально непознаваемое может быть частично познано.

В своей математической практике Брауэр все время ссылался на нерешенные в тот момент задачи, выбирая те из них, которые казались наименее перспективными и наименее интересными для решения. Например, в качестве такой задачи он часто использовал следующую:

Встречается ли последовательность 0123456789 в десятичном разложении числа  $\pi$ ?

Брауэр не представлял ни мощности современных компьютеров, ни тупости их пользователей, и, наверно, перевернулся бы в гробу, если бы некто, просмотрев несколько миллионов знаков числа  $\pi$ , вычисленных на компьютере, громко объявил бы, что он решил задачу Брауэра. Конечно же, сам Брауэр не желал, чтобы кто-нибудь решал такую задачу, он прекрасно понимал, что, даже если она вдруг будет решена в ту или иную сторону, он моментально подберет другую подобную.

**ПРИМЕР 3.** Рассмотрим, как в стиле Брауэра опровергается теорема о сходимости любой монотонной ограниченной сверху последовательности действительных чисел.

Пусть  $M$  — последовательность десятичных цифр

01234567891011121314151617181920...1000000,

---

<sup>7</sup>Мы специально используем столь длинное наименование, чтобы четко отличать логику, предложенную Брауэром, принятую и развивавшуюся во всех школах конструктивизма вплоть до 80-х годов XX в., от других конструктивных логик, которые начали появляться после 1980 г.

в которой записаны подряд все числа от 0 до 1000000. Рассмотрим последовательность рациональных чисел

$$a_n = \begin{cases} 1 - \frac{1}{2^n}, & \text{если до } n\text{-го знака} \\ & \text{не найдено } M \text{ в разложении } \pi, \\ 2 - \frac{1}{2^m}, & \text{иначе, где } m \text{ — наименьший знак,} \\ & \text{с которого начинается } M. \end{cases}$$

Тогда мы не знаем предела этой последовательности даже с точностью до 0,5.

Конечно же, такой метод рассуждений несколько уязвим, поскольку любая конкретная проблема такого рода может неожиданно быть решена. А самое главное, что он вызывал сильное раздражение у подавляющего большинства не только математиков, но также у прикладников и у физиков, воспитанных в традициях квазирелигии прогресса. Почему это мы ссылаемся на то, чего пока что не узнали? Вот дойдут у кого-нибудь руки, и узнаем!<sup>8</sup> А основывать научные рассуждения на невежестве — противоречит идеалам Великой и Святой Непререкаемой Научной Истины<sup>9</sup>.

Пользуясь теоремой Гёделя о неполноте, еще неизвестной в момент, когда Брауэр интенсивно развивал интуиционизм, но предсказанной им в полемике с Д. Гильбертом, можно модифицировать пример следующим образом.

**ПРИМЕР 4.** Пусть в нашей конкретной теории  $\mathfrak{G}$  — такая формула, для которой неразрешима ни она сама, ни проблема ее неразрешимости. Тогда построим последовательность рациональных чисел следующим образом.

$$a_n = \begin{cases} 1 - \frac{1}{2^n}, & \text{если до } n \text{ нет кодов доказательства } \mathfrak{G} \\ 2 - \frac{1}{2^m}, & \text{иначе, где } m \text{ — наименьший код доказательства } \mathfrak{G}. \end{cases}$$

Тогда мы не знаем предела этой последовательности даже с точностью до 0,5.

Но и данное рассуждение подвержено критике. Следуя в первую очередь идеям критики Брауэра, заметим, что построение

<sup>8</sup>Здесь оппоненты оказались более правы, чем Брауэр.

<sup>9</sup>Сравните с нынешними либералами и «неоконсерваторами», которые точно так же молятся на Великую и Святую Свободу и Демократию.

неразрывно связано с достаточно произвольно выбранной конкретной формализацией математики. А с классической точки зрения, если  $\mathfrak{G}$  неразрешима, то «содержательно истинно», что у нее нет никакого доказательства, и предел данной последовательности 1.

Следующей ахиллесовой пятой в первоначальных рассуждениях Брауэра было то, что он ссылался на «математическую интуицию» как на источник своих концепций. Верно в этой ссылке было только одно: Брауэр отвергал стандартную концепцию научного рационализма, которая и по сей день владеет умами почти всех ученых и особенно околонуучного сообщества. Он не стеснялся основывать свои рассуждения на незнании, почти уравнивая его в правах со знанием. Единственным преимуществом знания в концепции Брауэра было то, что, если незнание иногда устранимо, то раз полученное знание сохраняется навсегда.

Таким образом, Брауэр не был в деструктивной оппозиции к понятию прогресса<sup>10</sup>. Знание Брауэр считал неуничтожимым и прогрессивно развивающимся. Он принципиально отвлекался от того, что, познавая или приобретая новое, мы слишком часто теряем не менее ценное старое. Таким образом, он позиционировал свою концепцию как занимающуюся *чистым знанием*, не игнорируя при этом тот очевидный факт, что, чем больше мы знаем, тем больше мы не знаем, и в пределе приходим к мудрости Сократа: «Я знаю то, что ничего не знаю». Все конструктивистские школы XX века разделяли, наряду с интуиционистской логикой Брауэра, его отношение к математике как к области чистого знания.

В защиту Брауэра стоит привести по крайней мере два важных положения. Во-первых, восточная философия и мистика стали решающими вдохновляющими факторами для Брауэра, и его мировоззрение близко к чистому идеализму<sup>11</sup>. Так что ссылки на интуицию можно считать несколько неудачной попыткой перевода на европейский язык понятия чистого созерцания и дао. Во-вторых, современное науковедение, исследуя научные

<sup>10</sup>К самому этому понятию, а не к его общепринятой вульгаризации!

<sup>11</sup>И во всяком случае намного ближе к исходным платоновским концепциям, чем так называемый «математический платонизм».

революции (одной из которых, безусловно, стало открытие конструктивизма Брауэром), пришло к общему закону: *первооткрыватель чего-то принципиально нового и важного, как правило, неадекватно обосновывает и неадекватно позиционирует свое открытие*. Брауэр здесь не был исключением.

Видимо, для Брауэра не явилось сюрпризом, что его диссертация вызвала раздражение научного сообщества. Может быть, скорее для него было сюрпризом<sup>12</sup>, что его диссертация была тем не менее успешно защищена. И следующий шаг Брауэра еще раз показывает, насколько права восточная мудрость: «Умный человек с честью выходит из такой ситуации, в которую мудрый не попадает». Брауэр не стал разбивать голову о стену, пытаясь пробить броню непонимания и отторжения. Он как будто забыл на несколько лет о своей скандальной диссертации, занялся общей топологией и добился таких результатов, что и поныне считается одним из ее отцов-основателей и классиков. В частности, прекрасен со всех сторон следующий его результат.

#### **Теорема о неподвижной точке Брауэра.**

*Любое непрерывное отображение конечномерного шара в себя имеет неподвижную точку.*

И неожиданно Брауэр, уже получив признание научного сообщества и достигнув такого неформального положения, когда его нельзя было далее игнорировать<sup>13</sup>, заявил, что та область, в которой он добился таких блестящих успехов, никакого смысла не имеет, и что вся математика больна с самого начала. Соответственно математику нужно полностью перестраивать. Брауэр стал (сначала один, а затем при помощи горстки учеников) выполнять эту титаническую задачу возведения здания математики на новом фундаменте и по новым правилам<sup>14</sup>.

Гениальность Брауэра проявилась еще и в том, что он с самого начала не ограничил конструктивизм финитными объектами.

---

<sup>12</sup>И честью для нидерландской науки.

<sup>13</sup>За формальными званиями он никогда не гнался, так и оставшись до самой пенсии старшим преподавателем провинциального нидерландского университета.

<sup>14</sup>Так и вспоминается цитата из Евангелия: «Разрушьте храм сей, и я в три дня воздвигну его» (Иоанн, гл.2, ст. 19).

Он ввел в свою математику, в частности, последовательности, алгоритм для которых неизвестен, и поэтому они не могут считаться полностью описанными финитными объектами. В самом деле, ведь не менее важно конструктивно работать с физическим датчиком, который отнюдь не финитен, чем с конечной последовательностью символов, задающей обычную программу.

Заметим еще раз, что цель данного обзора — не мемориальная, а реальная. Поэтому в дальнейшем изложении мы стремимся трансформировать идеи конструктивизма таким образом, чтобы были видны возможности их приложений к современной информатике и возможности их развития в современной математике (порядок приоритетов здесь очень важен!)

## 5 Базовые принципы и методы конструктивизма

### 5.1 Принцип конечной информации и чум-пространства

Сразу же после начала перестройки математики выяснилось, что конструктивная математика в гораздо большей степени требовательна к выбору исходных понятий и определений, чем классическая. Но, чтобы рассмотреть показательный пример с понятием действительного числа, нам необходимо явно сформулировать тот принцип, который также принадлежит в исходной формулировке Брауэру. Здесь ему придана более общая форма в связи с тем, что он рассматривается для различных разновидностей конструктивизма.

#### Принцип конечной информации

*Пусть  $\Phi$  — эффективное преобразование объекта  $\alpha$  в объект  $\beta$ . Тогда всякая конечная информация, которую можно вычислить по  $\beta$ , может быть вычислена, с использованием  $\Phi$  и конечной информации об  $\alpha$ .*

В частности, в интуиционизме Брауэра конечной информацией о функции финитных объектов считается любое конечное число результатов этой функции на заданных аргументах. Если эти результаты не являются финитными объектами, то в свою очередь берется конечная информация о них.

Рассмотрим с этой точки зрения пример с монотонной ограниченной последовательностью, который рассматривался в предыдущем параграфе.

Пусть  $\alpha$  — монотонная ограниченная сверху последовательность действительных чисел. Пусть некий метод  $\Phi$  претендует на то, что он умеет строить предел такой последовательности. Тогда конечная информация о пределе должна вычисляться по конечной информации о последовательности, то есть, в частности, по конечному числу ее результатов. Пусть ограничение сверху 3. Сконструируем входную последовательность  $\alpha$  действительных чисел следующим образом. Будем подавать на вход  $\Phi$  число 0 столько раз, сколько понадобится, чтобы  $(\Phi \alpha)$  наконец-то выдал ответ с точностью до 1. Если этот ответ будет больше 0.5, то в дальнейшем так и будем подавать на вход 0, если же он будет меньше 2, то станем подавать на вход число 2.5.

Заметим, что здесь мы проявили аккуратность. Поскольку о результате может также иметься лишь конечная информация, границы определены таким образом, чтобы по результату, вычисленному с точностью до 1, можно было с гарантией определить хотя бы одну из этих границ. При этом пришлось отказаться от автоматизма, что рассматриваемые случаи обязательно взаимоисключающие. Если обе границы удовлетворены, то дальше контрпример можно строить любым из двух способов.

Хотелось бы привести конкретный контрпример, а не контрпример относительно некоего гипотетического метода. Но в предыдущем параграфе было показано, что такие «контрпримеры» апеллируют к вполне конкретному незнанию, порожденному либо нынешним состоянием наших исследований, либо принятой в данный момент конкретной аксиоматической системой. А в данном рассуждении мы показали пример весьма общего метода доказательства эффективной неразрешимости задач, совершенно не использующего детали конкретной алгоритмической формализации и базирующегося лишь на принципе конечной информации. Но, чтобы его сформулировать, нам предстоит уточнить неформально вытекающую из принципа конечной информации структуру эффективно заданных (или же эффективно используемых) нефинитных объектов. В нижеследующем построении мы базируемся на идеях Ю.Л. Ершова [6]<sup>15</sup>.

<sup>15</sup>Можно было бы попытаться использовать идеи Д. Скотта, направленные на то же, но у Ю.Л. Ершова намного четче выделено понятие эффективного базиса пространства.



Напомним некоторые понятия из топологии.

Любое частично упорядоченное множество (чум)  $\mathcal{L}$  можно рассматривать как топологическое пространство, если определить базис окрестностей следующим образом. Окрестностью каждой точки  $x$  считается множество всех элементов  $y \in \mathcal{L}$ , таких, что  $y \succ x$ . Эти окрестности  $O_x$  вместе с пустым множеством образуют базис открытых множеств. Соответственно открытыми множествами являются все замкнутые вверх множества, то есть такие  $O \subseteq \mathcal{L}$ , что

$$\forall x (x \in O \Rightarrow \forall y (y \in \mathcal{L} \& x \prec y \Rightarrow y \in O)).$$

Все открытые множества образуются объединением конечных пересечений множеств  $O_x$ .

Такое топологическое пространство несколько необычно для привыкших к пространствам, базирующимся на действительных числах, поскольку если  $y \succ x$ , то  $y$  неотделимо от  $x$  в том смысле, что каждое открытое множество, содержащее  $x$ , содержит и  $y$ . Но для любых двух элементов  $x, y \in \mathcal{L}$  всегда существует такое открытое множество  $O$ , что  $\{x, y\} \not\subseteq O$ . Таким образом, любые два элемента различимы. Топологические пространства, обладающие таким слабым свойством различимости элементов, называются  $T_0$ -пространствами.  $T_0$ -пространства чаще всего строятся, исходя из чумов, и всегда порождают чумы своих элементов. Мы будем называть их здесь *чум-пространствами* или *топологическими чумами*.

Перейдем теперь к определениям, необходимым для строгого представления принципа конечной информации и следствий из него.

**ОПРЕДЕЛЕНИЕ 1.** Подмножество  $X$  чума  $L$  кофинально, если нет элемента, большего всех его членов:

$$\neg \exists x : L \forall y : L (y \in X \Rightarrow y \prec x).$$

Чум-пространство называется *полным*, если выполнены два условия.

1. У каждых двух элементов есть верхняя грань.
2. Любое линейно упорядоченное кофинальное множество элементов имеет верхнюю грань.

Чум-пространство *регулярно*, если оно полно, и у любых двух элементов есть нижняя грань.

*Точкой* чум-пространства называется его максимальный элемент.

*Базис* чум-пространства — всюду плотная совокупность элементов  $X$ .

Чум-пространство называется *полностью эффективным* ( $\mathcal{E}_0$ -пространство), если оно полно и в нем выделен базис элементов, называемых *эффективными приближениями* или просто *приближениями*.

Естественно определяются прямые произведения и прямые суммы в категории чум-пространств. Еще одной важной операцией над чум-пространством  $\mathbb{S}$  является взятие пространства  $\mathbb{S}^*$ , элементы которого называются *совокупностями* элементов  $\mathbb{S}$ . Совокупности — это конечные множества значений из  $\mathbb{S}$ , причем разные значения несравнимы. На совокупностях определяется следующий частичный порядок.

$$S_1 \preceq S_2 \iff \forall a (a \in S_1 \Rightarrow \exists b (b \in S_2 \& a \preceq b)).$$

Таким образом, совокупность считается больше, если она либо уточняет, либо дополняет другую. Заметим, что при этом вполне может быть так, что в большей совокупности меньше элементов, поскольку один элемент может иногда уточнять сразу несколько.

Неформально заметим, что при задании  $\mathcal{E}_0$ -пространства мы будем следить, чтобы его базис действительно был эффективно задан. Но заранее ограничивать себя алгоритмически заданным базисом либо финитными конструктивными объектами мы не будем. Причины этого достаточно принципиальны и показаны ниже.

**ПРИМЕР 5.** Пространство действительных чисел может быть представлено как получаемое следующими двумя шагами. Вначале производим пополнение чума, образованного интервалами с рациональными концами и интервалом  $(-\infty, +\infty)$ , при следующем отношении порядка:

$$[c, d] \preceq [a, b] \iff [a, b] \subseteq [c, d].$$

Неравенство  $\preceq$  можно содержательно понимать следующим образом

- (1)  $i \preceq j \iff j$  уточняет информацию, содержащуюся в  $i$ .

Соответственно самая неточная информация (никакой, кроме типовой принадлежности объекта) в интервале  $(-\infty, +\infty)$ . Это пространство регулярно, поскольку можно построить минимальный интервал, содержащий два данных. Оно не является решеткой, поскольку наибольшего элемента нет. Наибольший элемент можно было бы ввести как пустой интервал, но тогда мы получили бы действительные числа, пополненные значением «невозможная информация о числе».

Далее строим фактор-пространство получившегося  $\mathcal{E}_0$ -пространства по отношению эквивалентности

- (2)  $x \simeq y \iff x$  и  $y$  — точки, и каждые два приближения к ним пересекаются.

Сами действительные числа задаются как точки  $\mathcal{E}_0$ -пространства.

В данной конструкции есть два общих места. Большинство используемых нами пространств строится как пополнение базиса и в дальнейшем факторизация множества точек. Подобный процесс будем описывать следующим образом.

Возьмем в качестве приближений интервалы с рациональными концами и интервал  $(-\infty, +\infty)$  при отношении порядка:

$$[c, d] \preceq [a, b] \iff [a, b] \subseteq [c, d],$$

и факторизуем точки по отношению эквивалентности  $x \simeq y$ .

Получившееся пространство обозначим  $\mathbb{R}$ .

Рассмотренный пример приводит к общему определению  $\mathcal{E}$ -пространства.

**ОПРЕДЕЛЕНИЕ 2.** *Эффективное пространство ( $\mathcal{E}$ -пространство)* — фактор-пространство  $\mathcal{E}_0$ -пространства по отношению эквивалентности, выразимому как ПΣ-формула над приближениями к элементам:

$$\begin{aligned} \forall a (a \preceq x \Rightarrow \exists b (b \preceq y \& R(a, b)) \\ \& \\ \forall a (a \preceq y \Rightarrow \exists b (b \preceq x \& R(a, b))) \end{aligned}$$

либо же П-формула над приближениями к элементам:

$$\forall a, b (a \preceq x \& b \preceq y \Rightarrow R(a, b)).$$

В обоих случаях  $R$  — эффективно проверяемое свойство.

В первом случае такое пространство называется *тестовым пространством*, во втором — *потокком*.

**ПРИМЕР 6.** Рассмотрим теперь несколько других базисов.

**Базис 1.** Двоичные интервалы вида  $[n/2^k, (n+1)/2^k]$ . Отношение вложения как в прошлом примере. Построение пространства действительных чисел как в прошлом примере.

Это пространство — аналог двоичного представления действительных чисел. Его обозначим  $\mathbb{R}_2$ .

**Базис 2.** Можно выбрать троичные интервалы стандартным образом. Но интереснее видоизмененное представление: интервалы вида  $(n/3^k, (n+2)/3^k)$ . Здесь конечные точки интервалов исключены, поскольку интервалы одного уровня пересекаются. Построение пространства такое же, как в предыдущих примерах. Получившееся пространство обозначим  $\mathbb{R}_3$ .

**Базис 3.** Рассмотрим теперь представление на базе непрерывных дробей. Непрерывная дробь представляет собой конечное или бесконечное выражение вида

$$-5 + \frac{1}{1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4 + \dots}}}}$$

Чтобы ясно отделить конечную дробь от бесконечных, будем последний ее этаж записывать в виде  $m + \frac{1}{1}$ . Таким образом, непрерывную дробь можно представить следующей структурой данных: это кортеж либо бесконечная последовательность целых чисел. Все члены ее, начиная со второго, положительные натуральные числа. Последнее число конечной последовательности равно 1. В бесконечной последовательности все числа, начиная со второго, больше 0. Таким образом, в нашей кодировке  $-3$  записывается как  $[-4, 1]$ , а выписанная выше бесконечная непрерывная дробь как последовательность  $[-5, 1, 2, 3, 4, \dots]$ .

Известно, что любое действительное число однозначно разлагается в непрерывную дробь, причем рациональные числа представляются конечными дробями, а иррациональные — бесконечными.

Здесь естественно взять в качестве приближений конечные непрерывные дроби и точки естественно определяются как предел продолжающейся последовательности конечных дробей. Отождествления здесь не нужно, так что в каком-то смысле эта модель действительных чисел простейшая с классической точки зрения. Обозначим ее  $\mathbb{R}_{cf}$ .

**Базис 4.** Аналогичен базису 1, но теперь длина отрезка — любое неотрицательное рациональное число, а не обязательно  $\frac{1}{2^n}$ . Получившееся пространство обозначим  $\mathbb{R}_0$ . С классической точки зрения уж эти-то два базиса кажутся абсолютно изоморфными с точки зрения результата. И там, и здесь в качестве приближений дано значение числа вместе с его точностью.

**Базис 5.** (Сечения Дедекинда) Базисом являются рациональные числа, которые интерпретируются как сегмент  $(-\infty, a]$ . Отношение порядка — обычное отношение на рациональных числах. Точки отождествляются, если в каждой из пары последовательностей  $\alpha, \beta$ , сходящихся соответственно к первой и ко второй точке, для любого члена первой последовательности  $\alpha_n$  и для любого  $\varepsilon > 0$  есть член  $\beta_m$  второй последовательности, такой, что  $\alpha_n - \varepsilon < \beta_m$ , и наоборот. Поскольку предложение длинное, приведем его запись на точном логическом языке:

$$\alpha \sim \beta \iff \forall n : nat \forall \varepsilon : rational (\varepsilon > 0 \Rightarrow \exists m : nat ((\alpha \ n) - \varepsilon < (\beta \ m)) \& \forall n : nat \forall \varepsilon : rational (\varepsilon > 0 \Rightarrow \exists m : nat ((\beta \ n) - \varepsilon < (\alpha \ m))).$$

Наибольшую точку называем  $+\infty$ . Действительные числа  $\mathbb{R}_D$  — точки данного пространства, исключая бесконечность.

**Базис 6.** (Модифицированные сечения Дедекинда) Каждый элемент базиса состоит из двух рациональных чисел  $a < b$ . Второй элемент называется верхней границей. При сравнении двух элементов базиса границы должны совпадать, а первые члены сравниваются, как обычно. Такое определение порядка обеспечивает ограниченность каждой сходящейся последовательности, причем ее верхняя граница задана с самого начала.

При отождествлении двух последовательностей границы вообще не принимаются во внимание.

В этом случае  $+\infty$  не возникает.

С классической точки зрения все эти представления приводят к одному и тому же множеству действительных чисел. С конструктивной точки зрения ситуация намного интереснее, и мы разберем ее ниже, базируясь на анализе В.А. Успенского.

Совершенно другая концепция сходимости представлена в следующем примере. Начнем с важного для последующего изложения определения.

Пусть для некоторого устройства либо для некоторой программной системы имеется машина, обеспечивающая получение результатов каждой программы и каждого входных данных, если эта программа вообще дает результат на этих данных, так называемая *информационная машина*.

**ОПРЕДЕЛЕНИЕ 3.** Пусть дан некоторый дискретный тип программ  $\mathbf{P}$ , чум тестовой информации  $\mathbf{T}$  и чум-пространство  $\mathbb{S} = (\mathbb{A} \times \mathbb{R})^*$ . *Информационная машина* — функционал  $\mathbb{F} : (\mathbf{P}, \mathbf{T} \rightarrow \mathbb{S})$ , неубывающий по второму аргументу:

$$\forall p : \mathbf{P}, t_1, t_2 : \mathbf{T} (t_1 \preceq t_2 \Rightarrow (\mathbb{F} p t_1) \preceq (\mathbb{F} p t_2)).$$

Содержательно информационная машина интерпретируется как способ получения совокупностей пар  $\langle \text{аргумент}, \text{значение} \rangle$  по тесту программы.

**ПРИМЕР 7.** Рассмотрим, как можно задать пространство программно определенных функций.

Пусть базисом являются конечные множества пар натуральных чисел, упорядоченные по включению. Программы для алгоритмов также будем кодировать натуральными числами. Пусть информационной машиной  $K$  является алгоритм, строящийся следующим образом.

Возьмем универсальный всюду определенный пошаговый интерпретатор алгоритмов  $U$ , построенный Постом и обладающий следующими свойствами.

Алгоритм с программой  $p$  определен на  $n$  и дает на нем ответ  $m$  тогда и только тогда, когда для некоторого  $k$   $(U p n k) = m + 1$ . Таким образом, пока интерпретатор не вычислил ответа программы, он дает для данного аргумента  $n$  ответ 0. Поправим

программу интерпретатора таким образом, что, выдав один раз ненулевое значение, он будет выдавать его для всех  $k_1 > k$ .

Каким-то стандартным вычислимым образом взаимно-однозначно закодируем пары натуральных чисел натуральными числами, введя всюду определенную вычислимую функцию  $(\text{pair } m \ n)$  и обратные ей  $(\text{pr}_1 x)$ ,  $(\text{pr}_2 x)$  таким образом, что  $(\text{pair } (\text{pr}_1 x) \ (\text{pr}_2 x)) = x$ . Более того, это кодирование можно сделать таким, что для двух кодов пар при  $x_1 \geq x_2$  члены пар не могли бы оба убывать. Теперь построим всюду определенный алгоритм  $(K \ p \ x)$ , выдающий по программе  $p$  и тесту  $x$ , являющемуся натуральным числом, совокупность всех пар  $\langle n, r \rangle$ , таких, что для некоторого  $y \leq x$   $(U \ p \ n \ (\text{pr}_1 y)) = (\text{pr}_2 y) + 1$ .

Поскольку программы у нас неупорядочены, программно-определимая функция — предел совокупностей, являющихся результатами  $(K \ p \ x)$  при данном  $p$ . Заметим, что в пространстве совокупностей такой предел не обязательно является наибольшим элементом, и тем более не является таким в его замыкании. Ведь вычислимая функция может быть не всюду определена.

**ПРИМЕР 8.** Покажем, как всего одной программой можно задать множество всех функций.

Пусть базисом являются кортежи из 0, 1, упорядоченные по отношению «быть началом». Таким образом, базис у нас — полное двоичное дерево. Тесты совпадают с элементами базиса. Программа всего одна:  $p_0$ .

Информационная функция

$$(I \ p_0 \ x) = x.$$

Содержательно это функция «Чего изволите, босс?» В ответ на тест она выдает именно те результаты, которые заложены в данном тесте. Пределы линейных последовательностей тестов являются точками пространства. В нем определимы все последовательности элементов 0, 1.

Рассматривая последние два примера, мы видим, что линейно упорядоченный чум тестов означает детерминированность наших программ, а наличие несравнимых элементов означает наличие индетерминизма. А неограниченно повторяемый произвол порождает все, что угодно. Кстати, последний пример имеет прямое отношение к теории программирования.

Э. Дейкстра весьма обоснованно заметил в своей книге, что одной из самых вредных подпорок в программировании является необходимость делать альтернативы условного оператора обязательно взаимоисключающими. В самом деле, если мы выбираем наилучшее решение, и обе рассматриваемые альтернативы с точки зрения нашей системы оценок равнозначны, то можно выбрать любую из них. Это выражается следующим условным оператором:

```
if
    x >= y then write(x),
    y >= x then write(y)
fi
```

Этот условный оператор более логичен и гибок, чем традиционные детерминированные конструкции программирования. Но в точно такой же форме выражается и построение из примера 8:

```
if
    x <= x+1 then write(0),
    x <= x+2 then write(1)
fi
```

Заметим, что во всех рассмотренных случаях мы задали абстрактные объекты не просто базисом, а базисом из финитных объектов с разрешимыми отношениями равенства и порядка. Эти финитные объекты на практике могут служить приближениями к абстрактным и рассматриваться как конечная информация о них. Тем самым в каждом конкретном варианте представления абстрактных объектов получает точное представление и принцип конечной информации.

В случае Э-пространств принцип конечной информации может быть конкретизирован следующим образом.

**Топологическая формулировка принципа конечной информации.**

*Пусть есть эффективный функционал  $\Phi$  из одного Э-пространства в другое. Тогда для каждого приближения  $\beta_i$  к  $(\Phi \alpha)$  найдется приближение  $\alpha_j$  к  $\alpha$ , такое, что для каждого  $\alpha_1$ , для которого  $\alpha_j$  является приближением,  $\beta_i$  является приближением к  $(\Phi \alpha_1)$ .*

Например, для действительных чисел, заданных стягивающимися интервалами, для получения интервала приближения к



результату нужен интервал приближения к аргументу, и, таким образом, вычислимые функции оказываются по самому факту их определения непрерывными.

Если два пространства заданы своими базисами, то конструкции прямого произведения и прямой суммы таких пространств очевидны: в первом случае базисом являются пары элементов двух базисов,  $\langle a, b \rangle \preceq \langle c, d \rangle \iff a \preceq c \ \& \ b \preceq d$ . Во втором достаточно взять размеченное объединение двух базисов, добавить к ним глобальный минимальный элемент и считать элементы различных базисов несравнимыми.

**ТЕОРЕМА 1.** *Категории потоков и тестовых пространств декартово замкнуты.*

Ю.Л. Ершов показал, что в таких пространствах естественно определяются и функционалы сколь угодно высоких конечных типов. Мы здесь используем идею его конструкции в пересмотренном виде.

## 5.2 Метод провокации и некорректные задачи

Базируясь на принципе конечной информации, можно дать общий способ установления неразрешимости проблем в конструктивной математике, не зависящий от конкретной формализации алгоритма либо эффективного метода.

### Метод провокации

*Пусть дана проблема построения по каждому  $\alpha$ , имеющему тип T1, для которого выполнено свойство  $A(\alpha)$ , объекта  $\beta$  типа T2, удовлетворяющего свойству  $B(\alpha, \beta)$ . На конструктивном языке она в общем случае записывается формулой<sup>16</sup>*

$$(3) \quad \forall \alpha : T1 (A(\alpha) \Rightarrow \exists \beta : T2 B(\alpha, \beta))$$

*или же, в частном случае, когда построенный объект заведомо конечен,*

$$(4) \quad \forall \alpha : T1 (A(\alpha) \Rightarrow B_1(\alpha) \vee \dots \vee B_n(\alpha)).$$

<sup>16</sup>Именно в общем случае! Если формулы  $A$  и  $B$  не являются чисто дескриптивными, то ситуация может осложниться. Более того, ниже разобраный случай с  $\vee$  относится именно к данной группе исключений.

Пусть функционал  $\Phi$  претендует на то, что он строит по каждому объекту  $\alpha$ , удовлетворяющему свойству  $A(\alpha)$ , объект, удовлетворяющий свойству  $B(\alpha, (\Phi \alpha))$ , либо на то, что  $\Phi$  указывает верный член дизъюнкции. Чтобы установить, что данная проблема эффективно неразрешима, достаточно дать метод построения последовательности приближений  $(\varpi \Phi n)$ , который работает следующим образом. Он вырабатывает вспомогательную последовательность приближений  $(\varpi_0 n)$ , не зависящую от  $\Phi$ , подает ее на вход  $\Phi$  и в момент, когда функционал  $\Phi$  выдаст некоторое приближение к решению, обладающее заранее заданной заведомо достижимой характеристикой (например, некоторой точностью либо ответом на вопрос в конкретной точке), далее выдает в зависимости от этого ответа одну из конечного числа последовательностей приближений  $(\varpi_i n)$ , показывающую ошибочность выданного ответа (например, если выдано число с некоторой точностью, дальнейшие наши приближения находятся вне интервала, в котором может лежать данное число)<sup>17</sup>.

Таким образом, структуру провоцирующего функционала можно описать следующей (квази)программной конструкцией. Поскольку такой язык будет часто использоваться ниже, дадим некоторые пояснения.

Эффективные методы, которые считаются данными извне, задаются директивой `import`. Знания об этих методах описываются как `specification`. Отношение  $x \lesssim \Xi$  означает, что  $x$  является приближением к идеальному объекту  $\Xi$ . Тип приближений для типа `type` описывается как `approx(type1)`. Если дан функционал  $\Phi$ , то  $(\Phi^0 x)$  означает существующий согласно принципу

<sup>17</sup>(Примечание А.П. Бельтюкова) Возражение С.С. Лаврова против диагональной конструкции при доказательстве неразрешимости (в частности, таковы многие варианты «метода провокации»): контрпример для разрешающей процедуры не может использовать саму эту разрешающую процедуру.

Мой комментарий: это надо понимать так, что программа анализа программ является объектом более высокого уровня и внутри анализируемой программы оказаться не может.

В связи с этим, по С.С. Лаврову, все современные конструктивные доказательства алгоритмической неразрешимости и сложностной иерархии, основанные на диагональных методах, некорректны.

**Возражение автора.** Квазипрограммное описание метода провокации, данное ниже, показывает, что программа анализа и выдачи контрпримера ни в коем случае не оказывается *внутри* опровергаемой программы. Она ее вызывает как модуль.

конечной информации функционал, выдающий приближения к результату по приближениям к аргументу.

```

import  $\Phi$ : (type1  $\rightarrow$  type2);
specification
 $\forall \Xi$ : type1 ( $A(\Xi) \Rightarrow B(\Xi, (\Phi \Xi))$ ) &
 $\forall \Xi$ : type1 ( $A(\Xi) \Rightarrow \exists x_2$ : approx(type2) ( $x_2 \lesssim (\Phi \Xi) \& C(x_2)$ ))&
 $\forall x_2, x_3$ : approx(type2) ( $C(x_2) \& x_2 \preceq x_3 \Rightarrow C(x_3)$ )&
 $\forall x_2$ : approx(type2) ( $C(x_2) \vee \neg C(x_2)$ );

comment Описания вспомогательных функций  $\varpi_i$ ,
выдающих разные варианты провоцирующей последовательности
end comment
function  $\varpi$ : (Nat  $\rightarrow$  approx(type1));
parameter i: Nat;
begin
if
    not C( $(\Phi^0 (\varpi_0 i))$ ) then result ( $\varpi_0 i$ ),
    C( $(\Phi^0 (\varpi_0 i))$ ) then
    if
        D1( $(\Phi^0 (\varpi_0 i))$ ) then result ( $\varpi_1 i$ ),
        ...,
        Dn( $(\Phi^0 (\varpi_0 i))$ ) then result ( $\varpi_n i$ )
    fi;
fi
end;
```

ПРИМЕР 9. Покажем неразрешимость задачи  $\forall x : \text{Real} (x \leq 0 \vee x \geq 0)$  при определении действительных чисел через последовательность стягивающихся интервалов длины  $1/2^n$  (названное у нас  $\mathbb{R}$ ).

Пусть дана некоторая функция  $f$ : ( $\text{Real} \rightarrow \{1,2\}$ ), претендующая на решение этой проблемы. Будем подавать на ее вход последовательность  $[-1/2^{n+1}, 1/2^{n+1}]$ , пока  $f$  не выдаст номер варианта. Если она выдаст 1, дальше будем подавать  $[1/2^{n+i+1}, 1/2^{n+i}]$ , если 2, то  $[-1/2^{n+i+1}, -1/2^{n+i}]$ . Тем самым

мы обманули данную функцию, спровоцировав ее дать ответ и в дальнейшем изменив последовательность так, чтобы она под выданный вариант не подходила.

Соответственно неразрешима и задача

$$(5) \quad \forall x : \text{Real} (x < 0 \vee x = 0 \vee x > 0).$$

При развитии интуиционистского анализа методом провокации<sup>18</sup> были впервые открыты (20-е годы) задачи, «решение» которых, предлагаемое в классической математике, некорректно (например, вычисление производной или нахождение нуля непрерывной функции, меняющей знак). Некорректность в форме неразрешимости сохраняется для таких задач во всех формах конструктивизма. Впоследствии эти задачи были переоткрыты классиками и официально названы «некорректными». Но до сих пор, скажем, в знаменитой численной библиотеке CERN FORTRAN имеются алгоритмы, «решающие» некорректные задачи и, конечно же, дающие непредсказуемые ошибки (пример принадлежит акад. С.К. Годунову).

Более того, еще в 20-е годы в интуиционистской математике были открыты и методы превращения некорректных задач в корректные, причем порою более общие, чем впоследствии предложенные классиками этого направления Тихоновым и Самарским, хотя, конечно же, в технической части гораздо менее развитые. А именно интуиционисты обратили внимание на необходимость логической перестановки задачи, а прикладники ограничились переходом от одного метрического пространства к другому, что выразимо в рамках традиционной численной парадигмы вычислительной математики.

Брауэр пытался обратить внимание на некорректные задачи и на практическую полезность полученных результатов, но прикладники его пренебрежительно проигнорировали, что привело в дальнейшем к позиционированию интуиционистского варианта конструктивизма как чистой науки и к избеганию контактов с прикладниками.

Рассмотрим другие варианты применения метода провокации, а также методы превращения некорректных задач в корректные. Это проиллюстрирует, в частности, сильную зависимость

<sup>18</sup>Сам Брауэр так его не называл.

конструктивных результатов от удачного выбора исходных понятий<sup>19</sup>.

ПРИМЕР 10. Покажем, как в конструктивном анализе видоизменить предыдущую задачу, чтобы она стала корректной, а именно дадим метод для решения задачи

$$(6) \quad \forall x : \text{Real}, \varepsilon : \text{Rational} (\varepsilon > 0 \Rightarrow x > -\varepsilon \vee x < \varepsilon).$$

Вычислим  $x$  с точностью до  $\varepsilon/2$ . Пусть  $a$  — рациональное число, представляющее приближенное значение  $x$  (либо середина интервала, либо конкретное представление, если выбрано другое из представлений действительных чисел). Тогда имеет место по крайней мере одно из двух: либо  $a - \varepsilon/2 > -\varepsilon$ , либо  $a + \varepsilon/2 < \varepsilon$ . В первом случае гарантирована выполнимость первого члена дизъюнкции, во втором — второго.

Посмотрев внимательно на этот пример, сразу можно увидеть одну из причин неприятия «практиками» практически ценных советов интуиционистов. Ведь мы не даем однозначного решения, во многих случаях оба решения приемлемы. А, скажем, Тихонов первым пунктом в определении корректной задачи поставил единственность решения, что с логической точки зрения по крайней мере излишне.

Впоследствии практики мучительно долго приходили к необходимости отхода от однозначно определенных методов. В программировании первым обратил внимание на неестественность (и даже концептуальную противоречивость с принципами структурного программирования) требования полной детерминированности условных выражений Э. Дейкстра<sup>20</sup>.

ПРИМЕР 11. Покажем разрешимость задачи

$$\forall x : \text{Real} (x \leq 0 \vee x \geq 0)$$

<sup>19</sup>Не путать с конкретными представлениями понятий! Они становятся важны не в теории, а при приложениях теории к практике, где такие представления нужно выбирать очень хорошо. Поэтому теория, слишком завязанная на конкретное представление данных, моментально теряет большую часть практической ценности.

<sup>20</sup>Впрочем, зачем кидать камень в практиков, когда логики тысячелетиями ставили на первое место в проблеме классификации требование несовместимости различных альтернатив.

при определении действительных чисел через двоичные разложения. Поскольку все двоичные знаки становятся известными, первый же из них определяет знак числа. Разрешима даже более сильная проблема:

$$\forall x : \text{Real} (x < 0 \vee x \geq 0).$$

А вот трихотомия

$$\forall x : \text{Real} (x \leq 0 \vee x = 0 \vee x \geq 0)$$

по-прежнему остается неразрешимой, поскольку, вычислив сколько угодно много двоичных знаков, равных 0, мы не можем гарантировать, что следующий не окажется 1, и на этой основе можно применить метод провокации.

При такой конструкции действительных чисел вычислима и соответственно непрерывна функция

$$(f \ x) = \begin{cases} 1 & \text{при } x \geq 0, \\ 0 & \text{при } x < 0. \end{cases}$$

Заметим, что эти рассуждения применимы не только к 0, но и к любому двоично-рациональному числу, только придется подождать до последнего знака после точки.

Таким образом, в данном представлении действительных чисел числа становятся неравноправными, и оно весьма подозрительно (по меньшей мере) в качестве основы для построения конструктивного понятия действительного числа.

Здесь мы замечаем еще одну особенность конструктивной математики. Она гораздо тоньше анализирует исходные понятия, чем классическая, и сигнализирует не только о грубой ошибке, приводящей к противоречивости теории (как классическая), но и о более тонких концептуальных ошибках, к прямому противоречию не ведущих.

Правда, если рассмотреть последний пример с точки зрения классической топологии, то к нему находится прямая аналогия: множество действительных чисел, базисом топологии на котором взяты полуинтервалы  $[a, b)$ , где  $a$  — двоично-рациональное число. Эту топологию можно было бы назвать топологией двоичной системы счисления.

Рассмотрим еще один любопытный пример.

ПРИМЕР 12. Возьмем определение действительных чисел через непрерывные дроби  $\mathbb{R}_{cf}$ . В данном случае особой точкой является любое рациональное число. Более того, для каждого рационального числа  $a$  решается задача трихотомии

$$\forall x : \text{Real} (x \leq a \vee x = a \vee x \geq a),$$

и еще более того, она решается единым методом для всех рациональных чисел (вычисления непрерывной дроби действительного числа на один этаж глубже, чем дробь данного рационального числа), так что выполнено

$$\forall a : \text{Rational} \forall x : \text{Real} (x \leq a \vee x = a \vee x \geq a).$$

Таким образом, при данном определении действительных чисел, в частности, вычислима функция  $(\text{sgn } x)$ .

Но вот проблема, является ли действительное число рациональным, неразрешима, поскольку как только претендент выдаст ответ, что число рациональное, мы дальше начнем подавать бесконечную непрерывную дробь, а если ответит, что нет, на следующем же шаге закончим дробь.

Однако можно вычислить функцию Дирихле, если модифицировать пространство результатов (причем совершенно безобидно с классической точки зрения).

Возьмем известный пример функции Дирихле

$$(D x) = \begin{cases} 1, & \text{если число рациональное;} \\ 0, & \text{если число иррационально} \end{cases}$$

и модифицированной функции Дирихле, разрывной во всех рациональных точках:

$$(D_1 x) = \begin{cases} \frac{1}{n}, & \text{если рациональное число представляется} \\ & \text{несократимой дробью } \frac{m}{n}; \\ 0, & \text{если число иррационально.} \end{cases}$$

Эти функции вычислимы, если разрешить в начале любой непрерывной дроби иметь цепочку нулей, а бесконечная цепочка нулей — это ноль.

И опять у нас находится классическая аналогия с нестандартной топологией на действительных числах. Возьмем в качестве базиса все замкнутые сегменты с рациональными концами.

Рассмотренные примеры позволяют провести следующее построение.

Определим функционал из  $\mathbb{R}$  в  $\text{Nat}$  следующим образом.

```
function sign : (Real → Nat);
parameter x: Real;

begin
  for i: Nat from 0 by i+1 do
    (x0 i).down > 0 then {result(1);leave},
    (x0 i).up < 0 then {result(-1);leave},
    (x0 i).down ≤ 0 & (x0 i).up ≥ 0 then relax
  od;
end
```

Дадим некоторые пояснения к данному функционалу. Мы используем недетерминированную форму цикла, которая позволяет в случае хорошо структурированной программы обойтись без присваиваний. После **for** описываются параметры цикла (значения, которые изменяются от шага к шагу цикла); после **from** задаются их начальные значения, после **by** задаются изменения параметров на следующем шаге цикла.

Сам цикл подобен условному оператору и состоит из нескольких (не обязательно взаимоисключающих) альтернатив. **leave** — приказ завершить цикл. **relax** — пустой оператор.

Действительное число представляется как последовательность замкнутых интервалов

$$[(x^0 i).\text{down}, (x^0 i).\text{up}].$$

Здесь определена функция, которая почти всегда вычисляет  $(\text{sign } x)$ , но заикливается при  $x = 0$ . Этот пример показывает фактор, упущенный Брауэром в интуиционизме, но учитывавшийся во всех дальнейших вариантах конструктивизма: эффективные операторы могут быть не всюду определены, и самоограничение лишь всюду определенными функциями в конструктивной математике весьма затруднительно корректно провести.



ПРИМЕР 13. Видоизменим пример 9 для случая троичного разложения с интервалами удвоенной длины. Он проходит и для данного случая, что показывает, что корень зла не столько в выборе конкретной системы счисления, сколько в рабском следовании ей. Представление действительных чисел  $\mathbb{R}_3$  оказывается полностью эквивалентно  $\mathbb{R}$ . Это представление изобрел Брауэр.

ПРИМЕР 14. Этот пример<sup>21</sup> показывает, что не всегда разницу в конструктивном статусе представлений можно объяснить в терминах классической топологии. Рассмотрим  $\mathbb{R}$  и  $\mathbb{R}_0$ . Покажем, что нет эффективного функционала, который по каждому элементу  $\mathbb{R}_0$  строит равный ему элемент  $\mathbb{R}$ .

Воспользуемся методом провокации. Будем подавать на вход функционала последовательность сегментов

$$\left[1 - \frac{1}{2^n}, 2\right]$$

до тех пор, пока испытываемый функционал не выдаст значение элемента  $\mathbb{R}$  с точностью до  $1/4$ . После этого обманем его, поскольку у нас ошибка заведомо превосходит 1.

ПРИМЕР 15. Данный пример, принадлежащий Г. Минцу, показывает, что конструктивные структуры не всегда приводят к классическим топологиям.

Рассмотрим объединение двух сегментов  $(-\infty, 0]$  и  $[0, +\infty)$ . Обозначим его  $\mathbb{R}_M$ . Оно представляется через приближения аналогично  $\mathbb{R}$ , но в приближениях не может быть один конец сегмента положительный, а другой отрицательный. 0 здесь не является изолированной точкой, поскольку сегмента  $[0, 0]$  нет и соответственно в данном случае приближения не образуют базиса топологии на  $\mathbb{R}$  в традиционном смысле. Но разрешима проблема

$$\forall x : \text{Real}_M (x \geq 0 \vee x \leq 0).$$

ПРИМЕР 16. И, наконец, рассмотрим пример с программно определенными типами данных. Пусть наши программы имеют следующий вид

<sup>21</sup>Впервые встретившийся в другой конкретной формализации в работах Н.А. Шанина.

```
const a,b: Rational; function fab : (Nat → Rational);
parameter i: Nat;

begin
  if    i=0 then result(a),
      i>0 & (fab i-1)=a then result(a)
      i>0 & (fab i-1)=a then result(b)
      i>0 & (fab i-1)=b then result(b)
  fi;
end
```

Это рекурсивное недетерминированное определение. Содержательно оно означает, что мы до некоторого момента выдаем  $a$ , а затем  $b$ .

Две точки считаются равными, если с некоторого момента значения совпадают. Поскольку эти значения — рациональные числа, вроде бы мы получили те же самые рациональные числа.

Но в этом примере неразрешима задача отличить друг от друга два рациональных числа, поскольку какие бы числа  $a, b$  мы ни взяли, конкретная программа `fab` представляет как раз тот случай, на котором может обмануться любой функционал, поскольку если она слишком долго будет выдавать  $a$ , функционал даст ответ, и его можно будет обмануть.

Здесь мы столкнулись еще с одним вариантом метода провокации для недетерминированных алгоритмов. Показывается, что некоторый вариант исполнения параметра может обмануть функционал, и тем самым мы не можем гарантировать нужный результат. Но вопрос, обманет ли функционал данное конкретное исполнение, остается открытым.

Методом провокации можно анализировать на корректность множество задач. А нахождение конструктивно корректной переформулировки некорректной задачи является одним из способов практического решения некорректных задач и, наверно, потенциально самым мощным из таких способов.

Рассмотрим в связи с вышеизложенным классическую некорректную задачу, которой по традиции мучают первокурсников в стандартных курсах программирования, никогда не упоминая, что она некорректная и их решения смысла не имеют.

ПРИМЕР 17. (Метод деления пополам)

Рассмотрим равномерно непрерывную на отрезке  $[0, 1]$  функцию действительного переменного  $f : (\text{Real} \rightarrow \text{Real})$ . Пусть она удовлетворяет следующим условиям:

1.  $(f\ 0) = -1$ ;  $(f\ 1) = 1$ .
2.  $f$  возрастает на  $[0, 1]$ .

Как видно, у нас условия на функцию даже более жесткие, чем в стандартном методе деления пополам.

Конечным приближением к равномерно непрерывной функции можно считать кусочно-линейную трубку с рациональными точками сочленения кусков<sup>22</sup>, то есть конечную последовательность параллелограммов с вертикальными боковыми гранями и рациональными вершинами, такую, что каждый следующий параллелограмм расположен правее предыдущего по оси  $x$  и пересекается с ним по общей боковой грани. Любая равномерно непрерывная функция представима как предел стягивающейся последовательности таких трубок.

Поставим задачу нахождения корня такой функции (который, как известно, *в принципе* существует). Покажем, что нет эффективного функционала, решающего данную задачу.

Построим последовательность трубок следующего вида

$$(f_n\ x) = \left[ \begin{array}{l} [(0, -1), (0, -1 + \frac{1}{2^n}), (\frac{1}{3}, -\frac{1}{2^n}), (\frac{1}{3}, \frac{1}{2^n})], \\ [(\frac{1}{3}, -\frac{1}{2^n}), (\frac{1}{3}, \frac{1}{2^n}), (\frac{2}{3}, -\frac{1}{2^n}), (\frac{2}{3}, \frac{1}{2^n})], \\ [(\frac{2}{3}, -\frac{1}{2^n}), (\frac{2}{3}, \frac{1}{2^n}), (0, 1 - \frac{1}{2^n}), (0, 1)] \end{array} \right]$$

Пока выдаются такие трубки, корень может быть везде на  $[\frac{1}{3}, \frac{2}{3}]$ . Как только функционал выдаст корень с точностью до  $1/9$ , видоизменим следующую трубку так, чтобы корень попал в незахваченную ответом область данного интервала. Например, так работает следующий условный оператор:

<sup>22</sup>В дальнейшем мы строго обоснуем такой выбор; но сейчас это увело бы нас далеко в теоретические тонкости, не нужные для решения данной конкретной задачи.

```

if    a ≥ 1/2 then
      result [ [ (0, -1), (0, -1 + 1/2^{n+2}), (1/3, -1/2^{n+1}), (1/3, -1/2^{n+2}) ],
                [ (1/3, -1/2^{n+1}), (1/3, -1/2^{n+2}), (1/2, 1/2^{n+1}), (1/2, 1/2^{n+2}) ],
                [ (1/2, 1/2^{n+1}), (1/2, 1/2^{n+2}), (2/3, 3/2^{n+2}), (2/3, 1/2^n) ],
                [ (2/3, 3/2^{n+2}), (2/3, 1/2^n), (0, 1 - 1/2^{n+2}), (0, 1) ] ]
a ≤ 1/2 then
      result [ [ (0, -1), (0, -1 + 1/2^{n+2}), (1/3, -1/2^n), (1/3, -3/2^{n+2}) ],
                [ (1/3, -1/2^n), (1/3, -3/2^{n+2}), (1/2, -1/2^{n+1}), (1/2, -1/2^{n+2}) ],
                [ (1/2, -1/2^{n+1}), (1/2, -1/2^{n+2}), (2/3, 1/2^{n+2}), (2/3, 1/2^{n+1}) ],
                [ (2/3, 1/2^{n+2}), (2/3, 1/2^{n+1}), (0, 1 - 1/2^{n+2}), (0, 1) ] ]
fi;
end

```

Но ведь программу для решения этой неразрешимой задачи пишет любой<sup>23</sup> первокурсник. В чем же дело? В этой программе он всю использует неразрешимую задачу (5), а «мудрые» преподаватели никогда не дают ему тесты, на которых выявляется некорректность данной задачи.

Возникает вопрос, как же решить эту нерешаемую проблему? Нужно просто чуть-чуть видоизменить ее постановку, поняв, что же нам на самом деле может быть нужно на практике. Нам нужно такое значение аргумента, при котором функция была бы практически неотличима от нуля. А найти такое значение — разрешимая задача, и метод деления отрезка пополам великолепно с ней справляется за счет применения недетерминированного условного оператора из (6). Тем самым приходим к корректной постановке конструктивной задачи:

$$\begin{aligned} \forall f : (\text{Real} \rightarrow \text{Real}) ((f\ 0) < 0 \& (f\ 1) > 0 \Rightarrow \\ \forall \varepsilon : \text{Real} (\varepsilon > 0 \Rightarrow \exists x : \text{Real} (x > 0 \& x < 1 \& |(f\ x)| < \varepsilon)) \end{aligned}$$

<sup>23</sup>В нынешнее время приходится добавить: не безнадежно зазомбированный ЕГЭ.

## 6 Интуиционистская логика как конструктивная

Выше мы неоднократно ссылались на то, что для конструктивиста утверждение есть в значительной степени задача на построение. Можно сказать, что он интересуется не истинностью, а реализуемостью. В данном случае реализуемость, конечно же, обычно понимается как реализуемость *в принципе*, и вопрос о том, можно ли уложиться в данные ресурсные ограничения на первый взгляд кажется вторым вопросом, который можно поставить себе после того, как математическая реализуемость установлена. Более того, именно такое отношение к задачам, решаемым в конструктивной математике, было полностью общепринятым вплоть до 80-х годов XX в. Прежде всего рассмотрим, что получалось на данном пути. Ведь пренебрежение чисто практическими аспектами отнюдь не всегда приводит к непрактичным теориям. Гораздо чаще к таковым приводит именно ползучее следование тому, что кажется в данный момент «запросами практики».

А.Н. Колмогоров (1930) (и чуть позже А. Гейтинг, но в менее общем виде) дал общую конструкцию для естественных математических семантик конструктивизма именно как конструктивизма: абстрактное понятие *реализуемости*. Рассмотрим содержательную идею определения реализуемости по Колмогорову.

### ОПРЕДЕЛЕНИЕ 4. Колмогоровская реализуемость.

Сопоставим каждой формуле  $A$  некоторую совокупность объектов  $\mathbb{R} A$ , сообразуясь с содержательным пониманием задачи, решить которую необходимо для обоснования  $A$ . Объекты из данной совокупности будем называть *решениями* задачи  $A$ .  $a \in \mathbb{R} A$  означает, что  $a$  — решение  $A$ , т. е. что  $a \in \mathbb{R} A$ . Пусть  $U$  — наш универсум рассмотрения.

1. Решения элементарных формул задаются по определению.
2. Решение  $A \& B$  есть пара решений  $A$  и  $B$  соответственно, объединенная в единую структуру данных.
3. Решение  $\exists x A(x)$  есть структура данных из конкретного значения  $x_0 \in U$  и реализации  $A(x_0)$ .

4. Решение  $A \vee B$  есть вариантная структура, в которой содержится метка решенной подзадачи (либо  $A$ , либо  $B$ ) и решение указанной задачи.
5. Решение  $A \Rightarrow B$  есть эффективный функционал, определенный на всех решениях  $A$  и преобразующий их в решения  $B$ .
6. Решение  $\forall x A(x)$  есть эффективный функционал, определенный на всех элементах универса  $U$  и преобразующий каждый  $x_0 \in U$  в решение  $A(x_0)$ .
7. Если нет таких  $a$ , что  $a \textcircled{R} A$ , то  $0 \textcircled{R} \neg A$ .

Это определение можно превратить в формальное двумя путями. Более легкий состоит в том, чтобы взять конкретные представления перечисленных структур данных и функционалов (например, кодирование натуральными числами и стандартное понятие алгоритма, как это сделал С.К. Клини, которому и принадлежит исключительно удачный термин «реализуемость»). Второй — попытаться действовать предельно абстрактно с тем, чтобы дать общий метод и общую схему, применимую к целому ряду частных случаев, в том чисел и к таким понятиям решения, которые либо существенно уже общего понятия алгоритма, либо формально шире, либо просто не укладываются в ту же парадигму, что и стандартная алгоритмическая вычислимость. В соответствии с общим духом данной работы здесь избирается второй путь. Как и формализация принципа конечной информации, этот путь требует, чтобы сначала были определены условия на тип объектов, являющихся реализациями, и необходимые операции на нем.

**ОПРЕДЕЛЕНИЕ 5. Пространство реализаций** — совокупность объектов, на которой заданы константы **nihil**, **1**, **2**; всюду определенные *собирающие функции* ( $\text{pair } x y$ ), ( $\text{in}_1 x$ ), ( $\text{in}_2 x$ ); частичные *разбирающие функции* ( $\text{pr}_1 x$ ), ( $\text{pr}_2 x$ ), ( $\text{label } x$ ), ( $\text{content } x$ ), ( $\text{ev } f x$ ), ( $\text{case } a b c$ ), обладающие следующими свойствами.

1. Если  $z = (\text{pair } x y)$ , то  $!(\text{pr}_1 z)$  и  $(\text{pr}_1 z) = x$ ,  $!(\text{pr}_2 z)$  и  $(\text{pr}_2 z) = y$ .

2. Если  $z = (\text{in}_1 x)$ , то  $!(\text{label } z), !(\text{content } z), (\text{label } z) = \mathbf{1}$ ,  
 $(\text{content } z) = x$ .
3. Если  $z = (\text{in}_2 x)$ , то  $!(\text{label } z), !(\text{content } z), (\text{label } z) = \mathbf{2}$ ,  
 $(\text{content } z) = x$ .
4. Если  $a = \mathbf{1}$ , то  $!(\text{case } a\_c)$  и  $(\text{case } a\_c) = b$ .
5. Если  $a = \mathbf{2}$ , то  $!(\text{case } a\_c)$  и  $(\text{case } a\_c) = c$ .

Для представления логики необходимы дополнительные требования на пространство, которые легче всего выражаются в терминах  $\lambda$ -исчисления с типами.

**ОПРЕДЕЛЕНИЕ 6.** Пространство реализаций называется *однократно  $\lambda$ -собирающим*, если для любого выражения  $t[x]$ , построенного при помощи функционалов  $(\text{pair } x y)$ ,  $(\text{in}_1 x)$ ,  $(\text{in}_2 x)$  из элементов данной интерпретации и переменной  $x$ , имеется элемент  $\lambda x. t[x]$ , такой, что

$$(\text{ev } \lambda x. t[x] a) = b \iff b = t[x \mid a].$$

Здесь и далее  $t[x \mid a]$  означает подстановку в выражение элемента вместо свободной переменной.

Пространство реализаций называется  *$\lambda$ -замкнутым*, если оно одновременно является моделью  $\lambda$ -исчисления с типами и типовой неопределенностью, где выполнены следующие приписывания типов:

$$\begin{aligned} \text{Type } [a] &= \tau \otimes \pi \supset \text{Type } [(\text{pr}_1 a)] = \tau; \\ \text{Type } [a] &= \tau \otimes \pi \supset \text{Type } [(\text{pr}_2 a)] = \pi; \\ \text{Type } [a] &= \tau \ \& \ \text{Type } [b] = \pi \supset \text{Type } [(\text{pair } a b)] = \tau \otimes \pi; \\ \text{Type } [a] &= \tau \oplus \pi \supset \text{Type } [(\text{label } a)] = \mathbf{label}; \\ \text{Type } [a] &= \tau \oplus \pi \ \& \ (\text{label } a) = \mathbf{1} \supset \text{Type } [(\text{content } a)] = \tau; \\ \text{Type } [a] &= \tau \oplus \pi \ \& \ (\text{label } a) = \mathbf{2} \supset \text{Type } [(\text{content } a)] = \pi; \\ \text{Type } [a] &= \tau \supset \text{Type } [(\text{in}_1 a)] = \tau \oplus \pi; \\ \text{Type } [b] &= \pi \supset \text{Type } [(\text{in}_2 b)] = \tau \oplus \pi; \\ \text{Type } [a] &= \mathbf{label} \ \& \ \text{Type } [b] = \tau \ \& \ \text{Type } [c] = \tau \supset \\ &\text{Type } [(\text{case } a b c)] = \tau; \\ \text{Type } [\mathbf{nil}] &\text{ может быть любым типом, в том числе и } \mathbf{void}; \\ \text{Type } [\mathbf{1}] &= \mathbf{label}; \\ \text{Type } [\mathbf{2}] &= \mathbf{label}. \end{aligned}$$

**Внимание!** Мы намеренно не приложили никаких усилий, чтобы разбирающие функции были определены только в том случае, когда объект собран при помощи соответствующей собирающей функции. А содержательно, определяя разбирающие и собирающие функции, необходимо следить, чтобы осмысленность разбирающих функций проверялась эффективно, т.е. в случае неприменимости они выдавали точное сообщение об ошибке. Я пока еще не знаю, как формализовать это требование абстрактно и вместе с тем адекватно<sup>24</sup>, но оно гораздо важнее формальной точной обратимости.

С использованием аппарата современной теоретической информатики [7] минимальное требование на замкнутость может быть кратко выражено следующим образом.

#### Условие $\lambda$ -замкнутости

*Любой типизированный  $\lambda$ -терм над функционалами из нашей интерпретации определяет объект соответствующего типа.*

**ОПРЕДЕЛЕНИЕ 7. Колмогоровская реализуемость (формальное определение).** Пусть  $\mathbb{R} A$  — множество решений  $A$ , и  $a \in \mathbb{R} A$  означает, что  $a$  — решение  $A$ , т.е. что  $a \in \mathbb{R} A$ . Пусть  $U$  — наш универс рассматривания. Оператор, преобразующий  $A$  в  $\mathbb{R} A$ , называется *колмогоровской интерпретацией*, если выполнены следующие условия:

1. Для конъюнкции:

$$(*) \forall a, b (a \in \mathbb{R} A \ \& \ b \in \mathbb{R} B \supset (\text{pair } a \ b) \in \mathbb{R} (A \& B)), \\ \forall c (c \in \mathbb{R} A \& B \supset !(\text{pr}_1 \ c) \in \mathbb{R} A \ \& \ !(\text{pr}_2 \ c) \in \mathbb{R} B).$$

2. Для квантора  $\exists$ :

$$(*) \forall a, c (a \in \mathbb{R} A(c) \supset (\text{pair } a \ c) \in \mathbb{R} \exists x A(x), \\ \forall c \left( \begin{array}{l} c \in \mathbb{R} \exists x A(x) \supset \\ !(\text{pr}_1 \ c) \in \mathbb{R} A(\text{pr}_1(c)) \end{array} \right)).$$

<sup>24</sup>Предупреждение для тех, кто попытается, увидев очевидную «дыру», заделать ее простейшим способом. Для всех простейших способов я знаю причины их неадекватности.



3. Для дизъюнкции:

$$\begin{aligned} & (*) \forall a (a \mathbb{R} A \supset (\text{in}_1 a) \mathbb{R} A \vee B), \\ & (*) \forall b (b \mathbb{R} B \supset (\text{in}_2 b) \mathbb{R} A \vee B), \\ \forall c \left( c \mathbb{R} (A \vee B) \supset \left( \begin{array}{l} \text{!(content } z) \ \& \ \text{(label } z) \ \& \\ ((\text{label } z) = \mathbf{1}) \ \& \ (\text{content } z) \ \mathbb{R} A) \ \vee \\ ((\text{label } z) = \mathbf{2}) \ \& \ (\text{content } z) \ \mathbb{R} B) \end{array} \right) \right). \end{aligned}$$

4. Для импликации:

$$\begin{aligned} & (*) \forall a (a \mathbb{R} A \ \& \ f \mathbb{R} (A \Rightarrow B) \supset \text{!(ev } f \ a) \ \& \ (\text{ev } f \ a) \ \mathbb{R} B). \\ & \forall f (\forall a (a \mathbb{R} A \ \& \ \supset \text{!(ev } f \ a) \ \& \ (\text{ev } f \ a) \ \mathbb{R} B) \supset f \mathbb{R} (A \Rightarrow B)). \end{aligned}$$

5. Для квантора  $\forall$ :

$$\begin{aligned} & (*) f \mathbb{R} \forall x A(x) \supset \forall a (a \in U \supset \text{!(ev } f \ a) \ \& \ (\text{ev } f \ a) \ \mathbb{R} A(a)). \\ & \forall f (\forall a (a \in U \supset \text{!(ev } f \ a) \ \& \ (\text{ev } f \ a) \ \mathbb{R} A(a)) \supset f \mathbb{R} (\forall x A(x))). \end{aligned}$$

6. Для отрицания:

$$\begin{aligned} & (*) \text{Если нет таких } a, \text{ что } a \mathbb{R} A, \text{ то } \mathbf{nihil} \mathbb{R} \neg A. \\ & \text{Если } \mathbf{nihil} \mathbb{R} \neg A, \text{ то нет ни одного } a, \text{ такого, что } a \mathbb{R} A. \end{aligned}$$

Если оставить лишь подпункты с (\*), то интерпретация называется *собирающей колмогоровской*, или просто *собирающей*. Если отбросить подпункты с (\*), то интерпретация называется *разбирающей колмогоровской*, или просто *разбирающей*.

Через  $\mathbf{!} \mathbb{R} A$  обозначается тот факт, что  $\mathbb{R} A$  непусто. Если  $\mathbf{!} \mathbb{R} A$ , то  $A$  называется *реализуемой* в данной колмогоровской интерпретации.

Это определение отличается от оригинального в нескольких отношениях. Во-первых, у Колмогорова<sup>25</sup> не было явно указанных отображений и операторов, а прямо говорилось, что множество  $\mathbb{R} A \ \& \ B$  — прямое произведение  $\mathbb{R} A \times \mathbb{R} B$ , множество  $\mathbb{R} A \vee B$  — прямая сумма  $\mathbb{R} A \oplus \mathbb{R} B$ . Пункт, соответствующий  $\mathbb{R} A \Rightarrow B$ , был определен только частично. Было сказано, что

<sup>25</sup>Как и у Гейтинга, как и у многих других вплоть до проникновения в данную область категорного подхода.

элементами  $\mathbb{R} A \Rightarrow B$  являются *эффективные* преобразования  $\mathbb{R} A$  в  $\mathbb{R} B$ , но само понятие *эффективного преобразования* оставалось неопределенным<sup>26</sup>.

Далее, поскольку вплоть до 40-х годов XX в. еще не было ясно, насколько фундаментально в эффективной математике понятие ошибки и что оно неустранимо, у Колмогорова (впрочем, и у Гейтинга тоже) не было никаких оговорок насчет частично определенных функционалов. И, наконец, у Колмогорова везде были эквивалентности, он даже не думал о том, что можно из его реализаций выбирать еще подмножества и получать более слабые колмогоровские интерпретации, соответствующие дополнительным критериям на решения либо даже ограниченным классам построений.

Таким образом, в сущности интерпретация Колмогорова была *мета-интерпретацией*, общим методом построения целого множества интерпретаций. Наше определение еще сильнее подчеркивает этот аспект и ее взаимосвязи с категорными конструкциями. Посмотрите на операторы и отображения, определявшиеся в пунктах 1 и 3. Они прямо соответствуют категорным конструкциям прямого произведения и прямой суммы.

Рассмотрим логические характеристики колмогоровской интерпретации.

**УТВЕРЖДЕНИЕ 1.** *В собирающей колмогоровской интерпретации имеют место следующие правила перехода:*

1. Если  $! \mathbb{R} A$  и  $! \mathbb{R} B$ , то  $! \mathbb{R} A \& B$ .
2. Если  $! \mathbb{R} A$  или  $! \mathbb{R} B$ , то  $! \mathbb{R} A \vee B$ .
3. Если  $! \mathbb{R} A(u)$  для некоторого  $u \in U$ , то  $! \mathbb{R} \exists x A(x)$ .

Таким образом, все прямые собирающие правила естественного вывода обычной классической логики выполнены в любой собирающей колмогоровской интерпретации. Установим дополнительные условия на собирающие интерпретации, при которых и косвенные собирающие правила для  $\Rightarrow$  и  $\forall$  также приемлемы.

<sup>26</sup>Здесь мы на следующем витке спирали практически вернулись к Колмогорову, но длительное время именно за эту недоговоренность его больше всего «пинали» и отказывались признать его приоритет.

Для этого необходимо условие замкнутости, которое содержательно представляется естественным и в некотором смысле минимальным для эффективных функционалов.

### Условие термальной замкнутости

*Любое корректное выражение, составленное из уже имеющихся эффективных функционалов и содержащее свободные переменные, может быть объявлено новым функционалом, аргументами которого являются свободные переменные.*

Точное формальное выражение данного условия в случае собирающих интерпретаций требует некоторого фрагмента модели  $\lambda$ -исчисления с типами, а именно чтобы пространство реализаций было *однократно  $\lambda$ -собирающим*.

Грубо говоря, если мы ухитрились уложиться в ограничения, создав некоторый сложный объект, то его составные части также построены согласно заданным ограничениям. Такое свойство может служить гарантией, что в первом приближении процессы построения описываются адекватно<sup>27</sup>.

<sup>27</sup>Во втором приближении дело может обстоять совсем не так. В частности, известно, что нетривиальное решение можно порою найти, лишь пройдя через ошибку. Уложиться в ограничения для финального результата порою возможно, лишь выйдя за их пределы по ходу построения и затем компенсировав сделанные «ошибки».

Такой прием обыгрывается в рассказе А. Азимова «ВЫХОД ИЗ ПОЛОЖЕНИЯ» из его знаменитой книги «Я, робот». Суперинтеллектуальному роботу дали задачу найти способ летать со сверхсветовой скоростью. Почему-то на этой задаче ломались все его предшественники. Он не сломался, поскольку был интеллектуальнее предыдущих и понял, что, хотя сверхсветовой прыжок приведет к смерти экипажа (а по законам робототехники причинение вреда человеку недопустимо), эта смерть обратима.

Но для компенсации «морального ущерба» робот ухитрился расцветить смерть пародийными видениями типа

«Это была смерть!

Это был мир без движения и без ощущений.

Мир тусклого, бесчувственного сознания — сознания тьмы, и безмолвия, и хаоса. И главное — сознания вечности. От человека остался лишь ничтожный белый клочок — его «я», заочневшее и перепуганное. . .

Потом проникновенно зазвучали слова, раскатившиеся над ним морем громового гула:

— На вас плохо сидит ваш гроб? Почему бы не попробовать эластичные

ПРИМЕР 18. Рассмотрим формулу  $\forall x (A(x) \vee \neg A(x))$ . В колмогоровской интерпретации ее реализуемость означает, что имеется такое  $f$ , что для всех  $x_0 \in U$

$$\begin{aligned} &!(\text{label } (ev f x_0)), \\ &!(\text{content } (ev f x_0)), \\ &(\text{label } (ev f x_0)) = \mathbf{1} \supset (\text{content } (ev f x_0)) \textcircled{\mathbb{R}} A, \\ &(\text{label } (ev f x_0)) = \mathbf{2} \supset (\text{content } (ev f x_0)) \textcircled{\mathbb{R}} \neg A. \end{aligned}$$

Из перечисленных свойств реализации следует, что функция  $\lambda x. (\text{label } (ev f x))$  эффективно решает проблему распознавания, выполнена или нет  $A$ . Заметим, что функция  $\lambda x. (\text{content } (ev f x))$  справляется с такой работой намного хуже.

Возможность краткого и красивого выражения разрешимости является одним из преимуществ конструктивизма в области выразительности.

Из условия  $\lambda$ -замкнутости следует тождественная реализуемость всех формул, доказуемых в интуиционистской логике. Но это условие тем не менее не гарантирует полной формальной адекватности реализуемости для интуиционистской логики. Например, для любой замкнутой формулы  $A$  реализуемо  $A \vee \neg A$ . В самом деле, если у нас есть реализация  $A$ , то реализуема первая часть дизъюнкции, если ее нет — вторая. Таким образом, интерпретация реализуемости интуиционистски замкнута лишь для равномерной реализуемости, когда логическая формула считается реализуемой, если есть типизируемый замкнутый  $\lambda$ -терм, который после постановки типов входящих в формулу пропозициональных букв и предикатов как типов переменных становится реализацией логической формулы. Итак, говорить о реализуемости логической формулы можно лишь в контексте. Рассмотрение формулы как формулы с предикатными и пропозициональными переменными — один из этих контекстов. Другой — наличие единой реализации для всех интерпретаций. Ни тому, ни другому требованию равномерности  $A \vee \neg A$  не удовлетворяет.

---

гробы фирмы Трупа С. Кадавра? Их научно разработанные формы соответствуют естественным изгибам тела и обогащены витаминами. Пользуйтесь гробами Кадавра — они удобны. Помните... вы... будете... мертвы... долго... долго!..”

Сложности, связанные с чисто формальной интерпретацией конструктивных формул через реализуемость, привели к тому, что в логических работах основными интерпретациями стали другие. Но для нас основной интерпретацией остается реализуемость, поскольку именно она выражает основное назначение конструктивной логики. Более того, для использования теории в качестве основы для построений, необходимо иметь не просто аксиомы, а аксиомы, помеченные объектами, их реализующими. Другими словами, *каждая конструктивная аксиома описывает некоторый программный модуль.*

**ОПРЕДЕЛЕНИЕ 8.** Конструктивная теория — теория, в которой каждая аксиома снабжена именем реализующего ее объекта:  $a \textcircled{R} \text{Axiom}$ . Модель конструктивной теории  $Th$  — такая колмогоровская интерпретация, в которой объект, специфицирующий аксиому, реализует данную аксиому.

Итак, для конструктивных теорий недостаточно абстрактной «истинности» аксиомы в любой модели. Нужно, чтобы ее реализовал именно тот объект, для которого она объявлена спецификацией в описании теории (или же, эквивалентно, объект, который объявлен реализацией аксиомы). Установим теорему корректности интуиционистской логики относительно колмогоровских интерпретаций с условием  $\lambda$ -замкнутости для произвольных теорий и выясним принципиальные препятствия, вырастающие перед абстрактной теоремой полноты.

**ТЕОРЕМА 2.** (Теорема об извлечении программы) По интуиционистскому доказательству формулы  $A$  в конструктивной теории  $Th$  можно построить  $\lambda$ -терм, реализующий  $A$  в любой конструктивной модели  $Th$ .

**Доказательство.** Воспользуемся формализацией выводов в виде графов, эквивалентно переформулировав ее таким образом, что правило размножения применяется лишь к допущениям вспомогательных выводов. Строим реализацию каждой формулы, встречающейся в выводе, как типизированный  $\lambda$ -терм, зависящий от свободных переменных, соответствующих действующим допущениям и произвольным переменным. Поэтому в описании правил вывода будем ссылаться на посылки правила как на специфицированные формулы вида  $a \textcircled{R} A$ .

1. Аксиома  $a \textcircled{R} A$  остается без изменений.
2. Допущению сопоставляется новое, не использованное ранее, имя свободной переменной соответствующего типа  $x$  :  
Туре  $[A] \textcircled{R} A$ .
3. Правило извлечения следствий порождает применение функции:  $a \textcircled{R} A$  и  $f \textcircled{R} A \Rightarrow B$  порождают  $(f a) \textcircled{R} B$ .
4. Правило дедукции порождает  $\lambda$ -абстракцию. Если  $x \textcircled{R} A$  — допущение вспомогательного вывода-посылки,  $t \textcircled{R} B$  — его результат, то  $\lambda x. t \textcircled{R} (A \Rightarrow B)$ .
5. Правило разъединения порождает соответствующую проекцию пары: если  $c \textcircled{R} A \& B$ , то  $(\text{pr}_1 c) \textcircled{R} A$ .
6. Правило соединения порождает пару: если  $a \textcircled{R} A$ ,  $b \textcircled{R} B$ , то  $(\text{pair } a b) \textcircled{R} A \& B$ .
7. Правило порождения  $\vee$  порождает вложение: если  $a \textcircled{R} A$ , то  $(\text{in}_1 a) \textcircled{R} A \vee B$ .
8. Правило разбора случаев порождает условное предложение: если  $a \textcircled{R} A \vee B$ ;  $t[x] \textcircled{R} C$ , где  $C$  — результат первого вспомогательного вывода, а  $x \textcircled{R} A$  — его допущение;  $r[y] \textcircled{R} C$ , где  $C$  — результат второго вспомогательного вывода, а  $y \textcircled{R} B$  — его допущение, то

$$(\text{case } (\text{label } a) t[x \mid (\text{content } A)] r[y \mid (\text{content } A)]) \textcircled{R} C$$

как результат правила.

9. Правило вспомогательной константы придает значение этой константе, которое в дальнейшем подставляется вместо нее. Если  $a \textcircled{R} \exists x A(x)$ , то значением вводимой вспомогательной константы  $c_i$  становится  $(\text{pr}_1 a)$ , и  $(\text{pr}_2 a) \textcircled{R} A[x \mid c_i]$ . В дальнейшем во всех упоминаемых термах вспомогательные константы заменяются на их значения.
10. Правило доказательства на примере порождает пару. Если  $a \textcircled{R} A[x \mid t]$ , то  $(\text{pair } at) \textcircled{R} \exists x A(x)$ .

11. Правило обобщения порождает  $\lambda$ -выражение. Если  $z$  — произвольная переменная вывода,  $t \text{ @ } A(z)$ , то  $\lambda z. t \text{ @ } \forall x A(x)$ .
12. Правило перехода от общего к частному порождает вызов функции. Если  $f \text{ @ } \forall x A(x)$ , то  $(f t) \text{ @ } A[x | t]$ .
13. Правило приведения к абсурду. **nihil** @  $\neg A$ .
14. Из лжи следует все, что угодно. **nihil** @  $B$ .

Тривиальной индукцией по построению вывода доказывается корректность указанного алгоритма извлечения программы и соответственно равномерная реализуемость каждой доказуемой формулы.

Q.E.D.

Заметим, что данная теорема в нашей форме явно иллюстрирует несколько тонкостей. Во-первых, реализации  $\neg A$  либо формулы, полученной по *ex falso quodlibet*, никак не используют реализации предыдущих формул. В доказательстве корректности используется лишь тот факт, что одновременно  $A$  и  $\neg A$  не могут реализоваться ни в какой интерпретации. Во-вторых, даже в отсутствие отрицания целые куски доказательства могут не использоваться в реализации теоремы. В частности, к такому эффекту приводят формулы вида

$$\forall x, y (A(x, y) \Rightarrow B(x)).$$

$y$  здесь оказывается промежуточным значением, которое вычисляется, а потом забывается.

Теперь устраним сделанное предположение о том, что правило размножения применяется лишь к допущениям. Тогда необходимо видоизменить язык создаваемых программ, введя в него описания констант:

**const c=t;**

В дальнейшей части того блока, где введено описание константы,  $c$  имеет то же значение, что и  $t$ . Отметим известные факты, касающиеся функциональных языков. Если все вычисления детерминированы, то описания констант формально не изменяют

выразительной силы языка, но сокращение длины программы при этом может быть более чем в экспоненту раз. Если же вычисления недетерминированы, то описания констант дают принципиально новый эффект.  $s$  получает результат одного из вычислений  $t$  и передает его в неизменном виде, а если записать  $t$  многократно, каждый его экземпляр может дать свое значение.

Заметим, что предыдущее рассмотрение заодно показывает, что построенный нами алгоритм легко модифицируется при изменениях логического исчисления. Но эти модификации могут оказаться весьма существенными с практической точки зрения. Далее, только что приведенные соображения показывают, что язык логики в принципе нейтрален по отношению к детерминированности либо недетерминированности вычислений, но малейшие изменения в конкретной формализации могут склонить чашу весов либо в одну, либо в другую сторону.

Обобщая все сказанное, можно заметить, что интуиционистская логика является концептуально последовательным и красивым выражением понятия реализуемости в  $\lambda$ -исчислении с типами, но не совпадает с этим исчислением. В доказательстве не все части переходят в программу. Этот факт на самом деле имеет фундаментальное значение для приложений логики к информатике, а его игнорирование — одна из бед нынешнего примитивного и основанного на грубой силе подхода к приложениям логики. Итак, *в достаточно сложном конструктивном доказательстве с необходимостью возникают неконструктивные части*. Далее, эксперименты с реализуемостью, сделанные в данном параграфе, показывают, что есть широкий спектр возможностей варьировать понятие реализуемости. И, наконец, изменения в выборе формы представления доказательств, вроде бы неважные с внутренней логической точки зрения, могут приводить к принципиальным сдвигам в соотношениях между доказательствами и конструируемыми объектами.

## Литература

- [1] *Brouwer L.E.J.* Over de grondslagen der wiskunde. Amsterdam: Thesis, 1907.
- [2] *Brouwer L.E.J.* De onbetrouwbaarheid rer logische principes // Tijdschrift voor wijsbegeerte, **2**, 1908.
- [3] *Brouwer L.E.J.* Intuitionisme en formalisme. Groningen, 1912.



- [4] *Brouwer L.E.J.* Besitzt jede reele Zahl eine dezimalbruchentwicklung?// Proc. Acad. Amsterdam, **23**, p 949–954.
- [5] *Гейтинг А.* Интуиционизм. Введение. М., 2010.
- [6] *Ершов Ю.Л.* Теория нумераций. М.: Наука, 1977.
- [7] *Митчелл Дж.* Основания языков программирования. М.; Ижевск: РХД, 2010.