

Числа арифметики Пеано не обладают необходимыми для вычислимости свойствами и никакое логическое расширение арифметики Пеано не создаёт им эти свойства

Д. Л. Гуринович

30 января 2023 года

УДК 510.51+511.11

Содержание

	Аннотация	1
1	Арифметика Пеано не обладает свойствами, необходимыми для вычислимости	2
2	Формальная арифметика и аксиомы записи чисел-строк	12
	I. Модифицированные аксиомы Пеано	12
	II. Связь чисел и строк	16
3	Статья «String Theory»	24
4	Остальные аксиомы для арифметики строк	29
5	Заключительные замечания и благодарности	36
	Список литературы	46

Аннотация

Числа арифметики Пеано не обладают необходимыми для вычислимости свойствами и никакое логическое расширение арифметики Пеано не создаёт им эти свойства. То есть, с точки зрения возможности применять числа арифметики Пеано в качестве вычислимых объектов внутри какой-либо теории они не были аксиоматизированы ни арифметикой Пеано, ни каким-либо иным образом..

В настоящей статье обосновано наличие кризиса в нынешнем понимании вычислимости в математике и начато построение соответствующей теории, которая включает в себя замену чисел арифметики Пеано на совместимые с практической вычислимостью числа-строки при сохранении у них необходимых и корректных свойств чисел арифметики Пеано.

1 Арифметика Пеано не обладает свойствами, необходимыми для вычислимости

В математике (в теории вычислимости) сейчас считается, что любое алгоритмическое вычисление может быть исчерпывающим образом записано и исследовано на языке арифметики и в рамках её логики, используя, в частности, представимые (а при необходимых арифметических расширениях теории Пеано и определимые) в арифметике частичные рекурсивные функции чисел арифметики Пеано. Где под числами арифметики Пеано здесь подразумеваются аксиоматизированные в арифметике Пеано числа.

Сказанное в предыдущем абзаце – даже если бы оно было истинным – не может быть доказано, так как охватывает неопределённо широкий круг «интуитивных алгоритмических вычислений». Поэтому приведённое утверждение считается сейчас (в начале 2023 года) истинным тезисом, принимаемым без доказательства. Тезис был сформулирован Чёрчем и называется поэтому «тезис Чёрча».

Но есть некоторые способы построения и исполнения алгоритмов – примером чему служит «машина Тьюринга» – для которых даются доказательства соответствия результатов работы этих алгоритмов и соответствующих им частичных рекурсивных функций.

Кстати, Машина Тьюринга (МТ) тоже считается «универсальным вычислителем», как и частичные рекурсивные функции арифметики считаются таковыми на начало 2023 года. Только МТ считается «практическим» универсальным вычислителем – это мнение называется «тезис Тьюринга». Вместе оба тезиса образуют тезис Тьюринга-Чёрча.

Имеющиеся сейчас доказательства о сводимости вычислений на машине Тьюринга к частичным рекурсивным функциям арифметики бывают двух типов:

1. Входные данные МТ (машины Тьюринга) рассматриваются как числа, записанные единственным выбранным способом;
2. Входные данные МТ рассматриваются как строки (математики используют термин «слова», но будем использовать термин из программирования – где работа со строками достигла несравнимо более высокого уровня, чем работа со «словами» в современной математике). Но и строки рассматриваются в таких доказательствах как составленные из символов некоторого единственного выбранного алфавита.

Заметим, что в имеющихся доказательствах для п. 1 рассматриваются только арифметические свойства «практических чисел» (назовём их так) – которые общие для любого способа записи чисел. И на этом основании утверждается, что нет разницы между способами записи чисел. Однако если рассматривать длину записи числа, то нет ничего общего между длиной числа сто в десятичной записи:

100 – длина этого «практического числа» в десятичной системе счисления равна трём;

И длиной этого же числа, записанного в двоичной системе счисления:

1100100 – длина этого «практического числа» в двоичной системе счисления равна семи;

Но в рамках арифметики оба этих варианта «практического числа» – одно и то же число ариф-

метики и никакая рекурсивная функция с этим аргументом не сможет выдать 2 разных результата при одном и том же входном аргументе.

Возможно возражение, что длина – это функция из п.2 и надо рассматривать этот случай отдельно. Но ведь с таким же успехом можно вычислять длину практического числа, увеличенного на 1. И тогда обойтись без числовых свойств окажется невозможным. Поэтому вопрос о длине практического числа – совершенно корректный и он не сводится, вообще говоря, к п.2 выше.

И вычисление длины числа-строки – это, конечно же, вычисление. При этом у собственно «числа арифметики Пеано» нет длины, так как одному и тому же «числу арифметики Пеано» соответствует бесконечное количество типов чисел-строк с одинаковым значением как у числа арифметики, но с разными длинами. Таким образом, для чисел-строк есть такие вычисления, для которых нет аналогов у чисел арифметики Пеано.

Сопоставление чисел арифметики с практическими числами (п.1 выше) – наиболее прямолинейный подход для обнаружения проблем из-за схожести сопоставляемых объектов. Проблема с вычислением длины практического числа – это вопрос о том, есть ли разница между разными представлениями одного и того же числа на практике для арифметики? Можно ли средствами арифметики добиться какого-то соответствия данной разницы с результатом некоторой или некоторых функций арифметики?

Но у нас находится довольно простое рассуждение («простое» – после того, как оно найдено), выявляющее недостаточную выразительность арифметики Пеано в отношении вычислимости чисел-строк:

Соответствует ли арифметике переход от числа 10 в десятичной системе счисления к 11 в результате инкремента (функции следования, функции увеличений на 1)? Да. Соответствует ли арифметике переход от 1010 (то же десятичное число 10, но записанное в двоичной системе счисления) к 1011 в результате инкремента? Да, тоже соответствует. Но для выполнения вычисления число должно быть записано либо в десятичной системе счисления – как 10, либо (исключающее «или»!) в иной (например – в двоичной, как 1010) системе счисления или в иной форме вычислимого представления. Только тогда можно вычислить, что следующим числом будет 11, записанное в десятичной системе счисления, либо (исключающее «или»!) в иной (например – в двоичной, как 1011) системе счисления или в иной форме вычислимого представления.

А то, что для арифметики возможны оба взаимоисключающих с точки зрения законов вычислимости способа записи, ставит арифметику Пеано и её объекты (числа арифметики Пеано) вне этих законов, потому что арифметика вступает в противоречие (не логическое, а опосредованное фактическое) с законами вычислимости в своих интерпретациях, которые полностью соответствуют теории (арифметике Пеано) и неотличимы друг от друга в рамках её логики.

В следующем разделе будут приведены аксиомы арифметики Пеано, а позже приведём даже аксиоматику логически полной арифметики 2-го порядка для желающих формального сопоставления их с логикой приведенного рассуждения о невыразимости вычислений функций, оперирующих числами-строками, средствами арифметики Пеано и средствами любых её арифметических расширений.

Говоря «числа арифметики Пеано», я подразумеваю не те практические числа, которые представляют собой вычислимы интерпретации арифметики, а такие абстрактные «объекты», которые соответствуют сразу всем возможным интерпретациям арифметики Пеано – в том числе интерпретациям, не являющимся вычислимыми – а такие мы тоже рассмотрим ниже.

И никакие «хитрости» вроде пар (номер системы счисления, номер числа в этой системе счисления) и их канторовской нумерации – не помогают в построении логики вычислимости. Потому что вопрос идёт о реальной вычислимости, а никакое число арифметики – канторовская это нумерация, или что бы то ни было другое – не являются вычислимыми.

Тут как в случае инвентарных номеров для объектов учёта. Номера не заменяют собой свойства объектов учёта. И нам нужна теория для исследования свойств вычислимости, а не для построения соответствия между вычислимыми объектами и объектами той теории, в которой нет необходимых для вычислимости свойств.

И никакие функции на основе объектов (в областях определения и значений этих функций), которые не обладают свойствами вычислимости – не позволят нам создать эти свойства. «Из ничего не выйдет ничего» (Эпикур, Лукреций, Аврелий и т.д.).

В приведенном рассуждении об инкременте для 10-ной и 2-ной систем счисления речь идёт не о логическом противоречии в арифметике, потому что интерпретации существуют независимо друг от друга и их «логики» не конфликтуют между собой. Но если бы их логики вычислимости обе соответствовали одной и той же теории (а теория обладала бы логикой вычислимости), и эти интерпретации при этом были неотличимы друг от друга в рамках этой теории, то тогда их логики вычислимости вступили бы между собой в конфликт и «полем боя» оказалась общая для них теория, включающая в себя взаимоисключающие варианты логики вычислимости этих интерпретаций. Конфликт не возникает потому, что логикой вычислимости арифметика Пеано не обладает.

Итак, в рамках интерпретаций арифметики, построенных на базе чисел-строк, мы не можем отличить одну вычислимую интерпретацию арифметики от другой аналогичной интерпретации. Но, возможно, строки слишком «конкретные»? Возможно, есть менее «жёсткие» интерпретации, которые не дают конкретной длины – что мы имеем для чисел-строк – но гарантируют, что размеры конечны и вычисления занимают конечное время – путь и неизвестно какой?

Сразу оговоримся, что даже если бы и нашлась форма записи чисел, обладающая необходимыми для вычислимости свойствами, логику которых содержала бы в себе арифметика – даже в этом случае нам пришлось бы придумывать теорию для чисел-строк. Потому что «практические числа» – это числа-строки, как правило. В том числе и в сфере ИТ числа хранятся в памяти компьютера в соответствии с логикой строк. А вот необходимая логика вычислимости для этой формы записи чисел отсутствует в арифметике, как мы только что разобрали.

Так обеспечивает ли арифметика необходимые для вычислимости свойства – пусть для некой «менее конкретной» своей интерпретации, чем на базе чисел-строк?

Увы – есть невычислимые интерпретации арифметики Пеано среди именно чисел-строк. И это означает, что иные интерпретации (не строковые) тоже неотличимы в рамках арифметики от таких интерпретаций чисел-строк, логика которых заведомо несовместима с вычислимыми интерпретаци-

ями. Что доказывает отсутствие у арифметики необходимых для вычислимости свойств – даже для каких-то интерпретаций, отличных от интерпретаций, построенных на базе чисел-строк. А именно:

Пусть, ноль – это $1/1$ часть цифр ноль в бесконечной строке с повторяющейся группой цифр: 0000000000...

Один – $1/2$ часть цифр ноль в бесконечном периоде: 1010101010...

N – это $1/2^N$ часть цифр 0 в бесконечном периоде. То есть 1111..101111..101111..10

Где 1111..1 обозначает $2^N - 1$ цифр 1.

Можно предложить и мене сложную интерпретацию:

0 – это 000000...

1 – это 101010...

2 – это 110110110110...

3 – это 1110111011101110... и т.д.

Когда мы говорим про конечное время вычислений, то необходимым условием для этого является конечный размер используемых в вычислении входных данных и конечный размер результата вычислений. Притом сами входные данные могут быть и бесконечными, но использовать в этом случае надо лишь их конечную часть. Приведенное требование не слишком сложное в плане понимания свойств «интуитивной алгоритмической вычислимости». Но этому не слишком сложному требованию арифметика не соответствует, как видно по приведённой сейчас интерпретации.

Для того, чтобы сделать вычисления на практике – надо иметь конкретную запись числа и правила перехода от записи данного числа к следующему (увеличенному на 1) за ним. А арифметика Пеано и любые её арифметические расширения, вплоть до полной теории второго порядка даёт только ту логику, которая является общей для любого способа записи чисел, то есть – ни для какого конкретного. И практического вычисления с этим «неизвестно чем» выполнить невозможно.

Мы рассуждаем об арифметике и отсутствии в ней необходимых для вычислимости свойств с точки зрения теории чисел-строк. Пусть пока у нас нет формальной теории чисел-строк, но мы понимаем, о чём речь (мы не можем строить формальную теорию объектов, если до этого у нас не сложилось их неформальное понимание). И мы не можем разбирать свойства теории её собственными средствами – как известно из 2-й т. Гёделя о неполноте. Напомню, что 2-я т. Гёделя о неполноте – это о невозможности в рамках непротиворечивой теории доказать её непротиворечивость. То есть, непротиворечивая теория не может исключить то, что она противоречива и в ней доказано всё.

Таким образом, мы опираемся на логику чисел-строк при обосновании отсутствия у арифметики необходимых для вычислимости свойств. Это означает, что мы считаем логику чисел-строк достаточно выразительной, чтобы выражать заведомо невычислимые объекты. Но тогда и здесь возникает вопрос – что в логике чисел-строк позволяет нам считать некоторые варианты этих объектов невычислимыми?

Ответ очевиден: их бесконечность. Мы по опыту знаем, что невозможно охватить («вычис-

лить») то, что бесконечно. И вот здесь мы уже опираемся на реальность, которая и служит «точкой остановки» в нашем погружении ко всё более «исходным» посылкам. И опора на реальность – это не что-то абсолютное, а всего лишь наш опыт. И в другой реальности (в каком-то «параллельном мире») этот опыт мог бы оказаться иным.

Мы считаем, что если вдруг число-строка из примера про невычислимые (бесконечные) числа-строки действительно удовлетворяет требованию к своей структуре (действительно является числом-строкой), то это не проявляется каким-либо вычислимым образом. Например, я в таком случае ещё ни разу не слышал глас Божий: «Это бесконечная строка, Дмитрий, и она вся состоит из повторения начального блока единиц и нуля». Тем более, я не слышу глас Божий каждый раз, когда мне потребовалось бы «божественное вычисление» в отношении верификации неограниченного количества чисел-строк.

Разумеется, я не могу доказать, что никто другой не слышит «проверяющий» бесконечные числа-строки глас Божий или я не обрету способность слышать подобное в будущем. Но я исхожу из того, что такое не происходит и не будет происходить в нашей вселенной. В этом и состоит та «точка опоры» в реальности, на которую опирается наше доказательство об отсутствии необходимых для вычислимости свойств у арифметики.

В разбираемом вопросе мы должны опираться на наш опыт и реальность – какой мы её знаем. И такая необходимость при построении математических теорий возникает не каждую тысячу лет (геометрию аксиоматизировали более, чем 2,2 тысячи лет назад, а числа математики взяли из практики ещё раньше) – поэтому не удивительно, что математики не заметили полтора столетия тому назад (когда в 1860-е годы был поставлен вопрос о построении формальной теории для натуральных чисел), что они не учитывают те свойства в этой теории, которые необходимы для вычислимости.

А почему сорвалось дальнейшее построение теории для вычислимых чисел и восполнения допущенных упущений? В том числе и потому, что накануне и в ходе второй мировой войны было уничтожено немецкое математическое сообщество, а позже был убит или доведён до самоубийства Алан Тьюринг (фактически, он был искалечен по решению суда) в Англии – впрочем, едва ли один выдающийся математик мог доделать то, что создавалось с решающим вкладом уничтоженного к моменту гибели Тьюринга немецкого математического сообщества. И Гёдель вне этого сообщества уже не отвечал на самые принципиальные вопросы, потому что их не ставил умерший к тому моменту Гильберт, и равных ему последователей уже не появлялось.

А немецкое математическое сообщество стало таким выдающимся во второй половине 19 века и до своей гибели потому, что именно в нём был начат период логических открытий – начиная с публикации о неевклидовой геометрии Лобачевского – при содействии Гаусса – и далее трудами Георга Кантора при построении теории множеств, который унаследовал капиталы своего отца, заработанные в России, и поэтому имел необходимые условия для напряжённой умственной деятельности – даже не взирая на отсутствие необходимых для этого доходов от работы математика в силу конфликта с Леопольдом Кронекером (с грубыми этическими нарушениями со стороны последнего), который не позволял Георгу Кантору получить достойно оплачиваемую – с учётом его

математической квалификации и решаемых им задач – работу в Берлине и немало способствовал разрушению его душевного равновесия и его математической гениальности.

Я очень упрощаю, конечно. Было много чего невероятно интересного и передового в немецкой математике сверх сказанного выше со второй половины 19 века – достаточно почитать биографию Давида Гильберта (одной из ключевых фигур в периоде логических открытий), например, который активно общался и сотрудничал с замечательными математиками своего времени. Но нас тут интересуют первопричины интереса и исследований в области формализации логики, натуральных чисел и вычислимости. Принципиальным вопросом тогда стало преодоление парадоксов – возникших особенно часто в создаваемой теории множеств. И именно Гильберт предложил программу и методы формализации и исследования математических теорий средствами самих теорий, на базе чего Гёдель и многие другие математики позже получили свои глубокие и опустошительные результаты в вопросах неполноты, неразрешимости и т.п.

Необходимо учитывать и вклад германских чиновников того времени – таких, как Фридрих Альтхофф – глава университетского отдела в Министерстве образования Пруссии, чья система стала позже образцом для исследовательских университетов в Соединенных Штатах и большей части Европы (информация из Википедии). Нельзя не упомянуть и о Феликсе Клейне, который помимо своих математических достижений сумел преодолеть тяжелейшую депрессию (после крайнего переутомления от успешной «гонки» с Пуанкаре) и показал себя замечательным лектором, педагогом и организатором науки. В огромной мере благодаря его усилиям Гёттингенский университет стал на многие десятилетия лучшим для математических исследований и образования в Германии, если не в мире.

Такой комплекс благоприятных случайностей не повторился после 2-й мировой войны. Да и после 1-й мировой войны немецкое математическое сообщество «по инерции» ещё сохраняло свои возможности, но не развивалось, как прежде – даже Эмми Нётер не получала приличной заработной платы в Гёттингене, который всё ещё оставался столпом немецкого математического сообщества. А после 2-й мировой войны период логических открытий прервался. Это в России после катастрофы 1917 г. было утрачено понимание необходимости качественных условий для напряжённой умственной работы (утрачено не в научной среде, конечно, но она в значительной мере была уничтожена Революцией), но факты именно таковы, что в экстремальных условиях выживают только бактерии-экстремофилы и всякие тихоходки. А внедренный на революционном раже девиз «Талант всегда сам пройдёт себе дорогу» – это прекрасный путь «эволюции» от человека к тараканам, которые отлично выживают, и это их талант, который оказывается единственно ценным в свете приведённого лозунга.

Кое-что из обломков немецкого научного сообщества удалось собрать – это были школы ядерных технологий и ракетостроения – в США и СССР. Но исследования в основаниях математики и подобное наследие немецкого математического сообщества не интересовало никого, на фоне решения задач выживания в технологической гонке политических полюсов мира того времени.

Кстати, для демонстрации отсутствия свойств вычислимости в арифметике можно обойтись без обращения к числам-строкам. Допустим, у нас есть последовательность объектов, удовлетворяющих

свойствам арифметики. То есть – некоторая интерпретация арифметики. Тогда:

Мы можем переставить 2 любых «соседних» числа и при этом построить новые арифметические операции, удовлетворяющие аксиомам арифметики (значит, можно задавать любой порядок минимум на конечных отрезках – путем простых перестановок). Поэтому арифметика не позволяет отличить, в каком порядке расположены числа-строки друг за другом.

Насчет простой перестановки конкретнее: Допустим – переставили 1 и 2 местами. Чтобы посчитать любую «новую» арифметическую операцию при таком «нарушенном порядке» сначала переименовываем операнды обратно – меняя обозначенное 1 на 2 и наоборот – если 1 и/или 2 есть среди операндов. Вычисляем результат нужной операции «старым» способом и совершаем с результатом операцию ту же замену (если есть что менять), что и с операндами. Подобную «модернизацию» легко проделать с любыми функциями и предикатами, разумеется.

То есть, с практической точки зрения объекты арифметики – даже не вычислимы числа. В арифметике есть только некоторые правила (аксиомы), которые при операциях с этими объектами должны выполняться. А практические числа, которые соответствуют этим правилам – не могут быть вычислены по этим правилам.

В чём же принципиальное упущение арифметики Пеано (и любых её арифметических расширений) в вопросе вычислимости по сравнению с логикой чисел-строк, например? В том, что число-строка не равна другому числу-строке не только в случае, если они соответствуют разному количеству операций увеличения на 1 от нуля, но и в том случае, если форма записи у них – разная. А именно:

Число 121 (десятичная запись), это:

$$1 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$

А в двоичном виде это:

$$1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

И такое разложение можно доказать в арифметике и вывести правила сложения и умножения «в столбик». Только в этих методах «вычисления в столбик» не будет практического смысла – потому что сначала надо разложить на разряды, чтобы применять эти методы вычисления. А как разложить на разряды, не имея конкретной записи числа? Не имея какого-то исходного разложения на разряды для того, чтобы перейти к нужному?

И дело тут в том, что в рамках арифметики разные записи одного и того же числа равны между собой. Если основание системы счисления указывать нижним индексом, то для арифметики верно:

$$121_{10} = 1111001_2$$

Но эти же числа не могут быть равными, если рассматривать их как числа-строки:

$$121_{10} \neq 1111001_2$$

Более того, если у нас один способ записи (пусть способ А) отличается от другого способа записи чисел (пусть способ В) только одной-единственной перестановкой чисел с номерами 100 и 101, например, то и в этом случае верно:

$$i_A \neq i_B$$

То есть – никакие числа-строки разного типа не равны между собой. Просто потому, что есть

математические операции, которые можно одинаково применять к «равным» числам разного типа (доказываем от противного) и прийти к разным результатам на 100 и 101 местах. А аксиомы равенства требуют сохранения равенства при любых одинаковых операциях – сколько бы раз их не повторять. В качестве таких повторяющихся операций можно взять операцию увеличения на 1 (для любых чисел) и операцию уменьшения на 1 (для любых чисел, кроме 0).

А из сказанного следует, что объектов арифметики Пеано меньше, чем способов записи чисел-строк. Элементарно доказывается диагональным методом Кантора. Только при построении диагональной последовательности, отличной от всех исходных, надо использовать те числа-строки, которые ещё не использовались в этой последовательности на предыдущих шагах её построения. Притом доказывать можно для одного и того же набора цифр (размер алфавита не меняется) для всех рассматриваемых способов записи чисел.

Кстати, Кантор доказал несчётность действительных чисел на интервале от 0 до 1, опираясь на числа-строки, а не на числа арифметики. Но этого, похоже, никто не заметил.

В конце данного раздела сформулируем те аксиомы «нарушения равенства» из-за строковых свойств, которые принципиально отличают числа-строки от чисел арифметики Пеано.

Во-первых,

Строки (включая числа-строки) разного типа не могут быть равны. А говоря иными словами – если хотя бы один символ строки одного типа равен некоторому символу другого типа, то эти типы – одинаковые. Отсюда – первая аксиома нарушения равенства из-за строковых свойств:

$(S_1) e_U(i) = e_K(j) \Rightarrow (\forall n e_U(n) = e_K(n)) \wedge (0_U = 0_K \vee \forall a (a \neq 0_U \wedge a \neq 0_K))$ – если в первом алфавите есть символ, равный символу из второго алфавита, то эти алфавиты полностью одинаковы как по символам, так и по соответствующим этим алфавитам числам-строкам. Для равенства всех соответствующих чисел-строк двух типов достаточно равенства нулей из этих двух типов. Остальные «соответствующие числа» и равенства между ними следуют из свойств равенства и операции инкремента. То есть, строки и числа-строки данного типа задаются не только алфавитом, но и способом построения чисел-строк на его основе – если для данного типа предусмотрен такой способ.

Оговорка про «если для данного типа предусмотрен такой способ» представлена вот этой частью аксиомы $(S_1): \forall a (a \neq 0_U \wedge a \neq 0_K)$. Которая соответствует тому случаю, когда нет чисел-строк ни типа U , ни типа K .

Обозначения тут такие:

n_U – число n , записанное соответствующим числом-строкой типа U .

$e_U(i)$ – элемент алфавита (символ или пустая строка) типа U для элемента номер i в этом алфавите. Номер – без указания типа по той причине, что данная функция даёт один и тот же результат независимо от типа числа-строки. То есть, $e_U(i_U) = e_U(i_K)$ для любых типов U и K .

Пустые строки разного типа не равны друг другу. Так принято в ИТ-сфере и из практических соображений мы тут будем считать так же. Это не имеет принципиального значения для вопросов вычислимости, но проще опираться на проработанные (в ИТ-сфере) подходы, чем «изобретать велосипед». К тому же в некоторых случаях неравенство пустых строк является обязательным – например, это мы увидим в разделе о статье String Theory, где нулем выступает пустая строка.

Обозначение для пустой строки типа U такое:

$$\Theta_U$$

В силу сказанного будем использовать обозначение $e(\Theta_U, i)$ наряду с обозначением $e_U(i)$. То есть – пустую строку данного типа мы можем использовать вместо обозначения типа в таких конструкциях.

Тип всех символов в данной строке (включая число-строку) одинаковый. И переход к новому числу при увеличении исходного числа на 1 (функция инкремента) тоже не меняет тип числа-строки. Это связано с тем, что необходима гарантия конечности набора «строительных элементов» чтобы обеспечить необходимые для вычислимости свойства. Подробнее про смысл ограниченности набора «строительных элементов» будет сказано в следующем разделе, связь чисел и строк, пункт «в-третьих».

Можно, конечно, представить себе такую логику, когда при увеличении на 1 происходит переход от числа-строки одного типа к числу-строке другого типа. Но если верен принцип ограниченности набора символов (цифр), то логично рассмотреть объединение тех алфавитов цифр, которые участвуют в построении чисел-строк данного способа записи. И этот алфавит будет отличаться от любого исходного. Тут аналогия с объектно-ориентированным программированием (ООП) в котором изменение одного свойства в сложном типе (классе) означает создание нового типа (класса). Так и «алфавит» является инкапсулированным (встроенным) свойством (если говорить в терминах ООП) для данного типа строк и чисел-строк. И такими же «инкапсулированными» свойствами являются ноль и операция инкремента для данного типа чисел-строк. И не только эти свойства «инкапсулированы» в соответствующие типы строк (включая числа-строки).

Отметим, что равенство всех «соответствующих» чисел-строк двух типов означает, что у любой такой пары «соответствующих чисел-строк» равны символы с одних и тех же мест этих двух строк. Это правило сравнения строк на предмет их (не)равенства. Правило общеизвестное, мы его не забудем и запишем в качестве ещё одной аксиомы, но позже.

Но здесь сразу напишем обозначения для операции взятия символа с нужного места строки:

$s(a, n)$ – символ с места n числа-строки a . А если длина строки меньше n , то функция возвращает пустую строку Θ_U – того же типа U , что и строка a . Здесь второй числовой аргумент (n) не имеет индекса типа по той же причине, что и в $e_U(i)$. То есть, потому, что $s(a, n_U) = s(a, n_K)$.

Во-вторых,

Строки с алфавитами разного размера не могут иметь один тип. Иными словами, если символы с номером 0 из 2-х алфавитов равны, то алфавиты имеют одинаковый размер. Это очевидная теорема, вытекающая из аксиомы (S_1), так как при равенстве любых символов из 2-х алфавитов – эти алфавиты полностью совпадают. И любые функции от пустых строк из этих алфавитов дают один и тот же результат. Среди этих функций есть и функция, возвращающая размер алфавита. Это наша первая теорема, полученная из аксиом нарушения равенства из-за строковых свойств:

$$(\tau.S_1) e_U(0) = e_K(0) \Rightarrow E_9(\Theta_U) = E_9(\Theta_K)$$

Обозначения тут такие:

$E_9(\Theta_U)$ – это «размер» алфавита типа U , записанный в десятичной системе счисления. «Раз-

мер» здесь – это количество не пустых символов в нём минус один. То есть, при «размере» алфавита, равного нулю, есть хотя бы один не пустой элемент алфавита (символ). Здесь нам надо выбрать систему счисления, в которой функция возвращает «размер» алфавита, потому что сравнивать числа (равны они или нет как числа) мы можем только в рамках одного и того же их типа. Для указания системы счисления тут используется число, на один меньше номера системы счисления. Это сделано не только для краткости (писать одну цифру вместо 2 для самого распространенного варианта), о чём скажу чуть ниже. Разумеется, выбор десятичной системы тут сделан «без ограничения общности» - вместо неё могла быть и любая другая вычислимая система счисления.

Поясню, почему используются «размеры» алфавита и нумерации систем счисления на 1 меньшие «обычных» значений. Потому, что ноль – особое число. И ему тогда соответствует особый случай алфавита с пустым элементом алфавита (пустой строкой) и единственным (не пустым) символом, а запись чисел в таком алфавите соответствует система счисления «счётных палочек». Из одного и того же символа данного алфавита невозможно строить разные числа иначе, кроме как повторяя этот символ в числе-строке некоторое уникальное (для данного числа) количество раз. В типичном случае это приведет к размеру, который экспоненциально (как минимум) больше размера того же числа, но записанного в позиционной системе счисления. То есть – индекс типа 0 (ноль) означает тут такой алфавит и такой способ записи числа, который заведомо отличается от достаточно эффективной (по своим размерам) позиционной записи для чисел-строк.

Мы ещё вернёмся в следующем разделе к тому, как аксиоматизировать систему счисления для данного типа, но заранее понятно, что это не сложно.

В-третьих,

Условимся, что для преобразования числа-строки i_K типа K к такому же числу i_U , записанному как число-строка типа U , мы будем использовать следующую операцию преобразования типа чисел-строк:

$$(e_U(0)) i_K$$

Операция преобразования типов имеет вид, аналогичный явному преобразованию типов в языке программирования Си. Только вместо имени типа, который пишется в языке Си в качестве левого операнда (который в круглых скобках), мы пишем первый (точнее, с номером ноль) элемент в алфавите нужного типа. А в алфавите у нас всегда есть минимум два элемента – хотя бы один не пустой символ и пустой элемент алфавита.

Свойства этой операции преобразования чисел-строк «с сохранением числа» такие:

$$(S_2) 0_U = (e_U(0)) 0_K$$

$$(S_3) i_U = (e_U(0)) j_K \Rightarrow i'_U = (e_U(0)) (j'_K)$$

Заметим, что число-строка ноль данного типа и порядок увеличения числа-строки данного типа на единицу задаются соответствующими аксиомами типа. Которые мы рассмотрим в следующем разделе, как уже упоминалось.

2 Формальная арифметика и аксиомы записи чисел-строк

I. Модифицированные аксиомы Пеано

Что касается чисел-строк конкретного типа, то они подчиняются, конечно, обычным свойствам арифметики, поэтому для них должны быть верными аксиомы теории Пеано и любых её арифметических расширений в случае фиксированного типа чисел-строк. Запишем явным образом аксиомы теории Q , достаточной для представления рекурсивных функций, а затем и её расширение до теории Пеано.

Дж. Булос, Р. Джеффри «Вычислимость и логика» Глава 14. Представимость в Q . Аксиомы теории Q .

$$(q_1) \forall i \forall j (i' = j' \Rightarrow i = j)$$

$$(q_2) \forall i 0 \neq i'$$

$$(q_3) \forall i (i \neq 0 \Rightarrow \exists j i = j')$$

$$(q_4) \forall i (i + 0 = i)$$

$$(q_5) \forall i \forall j (i + j' = (i + j)')$$

$$(q_6) \forall i (i \times 0 = 0)$$

$$(q_7) \forall i \forall j (i \times (j') = (i \times j) + i)$$

Отметим, что для знака умножения использован символ « $\times$ », потому что символ « $\cdot$ » мы будем использовать для обозначения конкатенации.

Теперь расставим индексы типа (индекс U) в добавление к обычной записи. А позже разберём, почему это необходимо.

$$(Q_1) \forall_U i_U \forall_U j_U (i'_U = j'_U \Rightarrow i_U = j_U)$$

$$(Q_2) \forall_U i_U 0_U \neq i'_U$$

$$(Q_3) \forall_U i_U (i_U \neq 0_U \Rightarrow \exists_U j_U i_U = j'_U)$$

$$(Q_4) \forall_U i_U (i_U + 0_U = i_U)$$

$$(Q_5) \forall_U i_U \forall_U j_U (i_U + j'_U = (i_U + j_U)')$$

$$(Q_6) \forall_U i_U (i_U \times 0_U = 0_U)$$

$$(Q_7) \forall_U i_U \forall_U j_U (i_U \times (j'_U) = (i_U \times j_U) + i_U)$$

Обозначение $\forall_U i_U$ означает, что в качестве возможных значений связанной переменной i_U выступают только все значения чисел-строк заданного типа. Логика такого «типированного» квантора общности (пояснение обозначения):

$$\forall_U i_U A(i_U)$$

Эквивалентна следующей формуле:

$$\forall i (\Theta_U = s(i, 0) \Rightarrow A(i))$$

Дальше мы разберем и функцию взятие символа $s(a, j)$ с произвольного места j строки a , и пустую строку заданного типа Θ_U .

Аналогично, формула $\exists_U i_U A(i_U)$ эквивалентна формуле $\exists i (\Theta_U = s(i, 0) \wedge A(i))$, что соответствует записи квантора существования $\exists i \dots$ в виде $\neg \forall i \neg \dots$. Действительно:

$$\exists i (\Theta_U = s(i, 0) \wedge A(i)) - \text{эквивалентно:}$$

$\neg \forall i \neg (\Theta_U = s(i, 0) \wedge A(i))$ – эквивалентно:

$\neg \forall i (\Theta_U = s(i, 0) \Rightarrow \neg A(i))$ – эквивалентно:

$\neg \forall_U i_U \neg A(i_U)$ – эквивалентно:

$\exists_U i_U A(i_U)$.

В той же книге: Глава 15 Неразрешимость, неопределимость и полнота. Добавление аксиом индукции к теории Q и переход к теории Пеано – «теории Z »:

$(q_i) A(0) \wedge \forall i (A(i) \Rightarrow A(i')) \Rightarrow \forall i A(i)$

То же с добавлением индекса типа:

$(Q_i) A(0_U) \wedge \forall_U i_U (A(i_U) \Rightarrow A(i'_U)) \Rightarrow \forall_U i_U (A(i_U))$

Разумеется, для теории чисел-строк и строк формула $A(i_U)$ понимается как формула, которая может зависеть ещё и от разных строковых аргументов, даже если они не являются числами-строками.

При этом аксиому Q_3 можно исключить, так как она выводится из аксиомы индукции и аксиомы Q_2 .

И ещё нужно добавить к данным аксиомам аксиомы логики, чтобы у нас получилась теория первого порядка с равенством. Но не будем здесь этого делать, так как у нас задача разобраться в природе вычислимости, которая создаёт основу для формализации, но формализацией в полной мере ещё не является. Однако про свойства равенства надо сказать отдельно.

В логике сформулированы аксиомы равенства для произвольных предикатов, обозначенных в формуле (J_2) как формульная переменная $A(\dots)$:

$(J_1) a = a$

$(J_2) a = b \Rightarrow (A(a) \Rightarrow A(b))$

Данные аксиомы без индекса типа, и в силу своей общелогической природы не нуждаются в том, чтобы этот индекс добавлялся.

Можно заменить эти общелогические аксиомы на аксиомы для конкретной теории первого порядка с равенством, избавившись от формульной переменной. Смотри, например, Э. Мендельсон, «Введение в математическую логику», Глава 2. Теории первого порядка. Раздел 8. Теория первого порядка с равенством. Предложение 2.26. <критерий того, что в теории первого порядка истинны аксиомы равенства>

Мы не будем этого делать сейчас, потому что подобные вопросы относятся уже к оптимизации набора исходных аксиом, а это сейчас не приоритетный вопрос. Поэтому будем пока считать, что среди общелогических аксиом есть и аксиомы для равенства. Но поскольку аксиомы Пеано взяты из конкретного учебника, в котором такая оптимизация должна была быть выполнена, то надо отметить, что обычно необходимо добавлять к теории первого порядка следующие «адаптированные» аксиомы о равенстве (при замене общелогических аксиом равенства на аксиомы равенства для данной теории без использования формульных переменных):

$x = x$

$x = y \Rightarrow (z = x \Rightarrow z = y)$

Первая из этих аксиом (рефлексивности) в теории Пеано не аксиоматизируется, так как она

следует для натуральных чисел из аксиомы $x + 0 = x$. Но в учебнике А. С. Герасимова «Курс математической логики и теории вычислимости» рефлексивность равенства не доказывается, а аксиоматизируются для теории Пеано. Что никаких противоречий не создаёт и тоже допустимо, хоть небольшая оптимизация (оптимизация не меняет логику теории) списка аксиом не исполнена.

А вот вторая из приведенных «адаптированных» аксиом для равенства имеется даже в теории Пеано. Занятно, что в отличном учебнике Дж. Булос, Р Джеффри – «Вычислимость и логика» вопрос о равенстве для теории Пеано оказывается упущенным из вида, и среди аксиом теории Пеано нет необходимой аксиомы о транзитивности равенства.

Остальные аксиомы Пеано тут – это аксиомы, которые выполнены для не просто чисел, но для чисел-строк определённого типа, на который указывает индекс U .

Для обозначения предметных переменных (и строк, и чисел-строк) будем использовать все переменные из прописных букв, кроме i, j, k, l, m, n – которые оставим для чисел-строк. И кроме переменной q , которую оставим для значений чисел-строк и пустой строки Θ_U – точнее, пустых строк, потому что тут возможны разные типы. Обычно для индекса типа в аксиомах будем использовать индексы U, K (прописные). Конечно, эти индексы типа могут приписываться к переменной – когда тип строки для этой переменной известен.

Индекс типа числа-строки нам нужен для того, чтобы связать то, как числа данного типа (их логика задана модифицированными аксиомами Пеано с соответствующим индексом типа) записаны при помощи символов алфавита данного типа и операций конкатенации. Понятно, что десятичная запись, например, будет отличаться от записи числа в стандартной интерпретации (далее будет понятно, о чём речь).

Аксиоматика арифметики даёт разные арифметические операции в сочетании с разными аксиомами построения нуля и инкремента. А это значит, что во избежание противоречий, необходимо разделять одинаковые арифметические аксиоматики на отдельные экземпляры этих аксиоматик, каждый из которых (экземпляр аксиоматик) объединяется с аксиоматикой только своего типа строк и чисел-строк.

И это же означает, что аксиоматика арифметики Пеано не является полноценной аксиоматикой для вычислимых натуральных чисел. Это всего лишь общая логика для разных способов построения вычислимых натуральных чисел. И это настолько глубокая абстракция, что для построения логики каких-нибудь вычислимых натуральных чисел, эту аксиоматику Пеано надо «конкретизировать» (создать отдельный экземпляр этой аксиоматики) именно для выбранного способа построения вычислимых натуральных вычислимых чисел, и не использовать его больше ни для какого другого способа подобного построения – чтобы избежать противоречий из-за аксиом нарушения равенства по причине строковых свойств, например.

Например, сумма в арифметике Пеано – это возможность посчитать глубину вложенности композиции инкремента. А умножение – это возможность повторять эту композицию инкремента заданной глубины заданное количество раз – то есть, сделать её аналогом инкремента, только масштабированным аналогом.

Сказанное о сложении и умножении кажется очевидным для любых случаев счётных объектов.

Но именно в том и счётность – на той же бумаге это работает. А если живые рыбки в аквариуме с не прозрачными лабиринтами – то попробуйте узнать, сколько их запущено в аквариум. Хотя их можно было пускать по одной и вести счёт на бумажке. Но если эти записи не хранить, то как можно использовать такие сложные аквариумы как готовые суммы, например?

А без аксиом нарушения равенства (из-за строковых свойств, например) в арифметике нет логики разделения между вычислимыми и не вычислимыми способами записи чисел-строк. Не говоря уж о способе отделить типы вычисляемых чисел-строк друг от друга.

Кроме того, число-строка состоит только из символов единственного типа и в однозначном порядке. Но этого нет для арифметики. Сочетание аксиом нарушения равенства с алфавитом делает запись числа-строки однозначной.

В то же время для цифры (стоящей на заданном месте в заданном числе-строке заданного типа) чтобы получить её номер в алфавите заданного типа – для этого в арифметике Пеано вполне могут быть нужные функции (представления для нужных функций). Разумеется, результатом будет число арифметики, и оно не будет обладать свойствами вычислимости. И для одного и того же числа арифметики Пеано и одного и того же места «цифры» мы будем получать разные результаты, в зависимости от интересующего нас типа числа-строки. При этом потребуется сопоставить интересующие нас типы (не все, а какую-то их счётную выборку) числам арифметики, чтобы оперировать ими внутри арифметики Пеано.

В то же время строка (или число-строка) содержит однозначную информацию о своём типе и последовательности символов для себя, как строки. И среди чисел-строк имеются заведомо вычисляемые их типы.

II. Связь чисел и строк

Теперь, сохраняя те же условности обозначения для строк, имеющих свойства чисел (для переменных со значением число-строка мы используем оговоренные ранее буквы) зададим связь между алфавитом для строк и числами.

Чтобы придать числам свойства строк, достаточно задать строку, которая будет 0 (нулём) данного типа и задать операцию добавления единицы (функцию следования или инкремент) в терминах строк. После этого остальные арифметические операции в терминах строк можно легко вывести на основании свойств строк и типированных аксиом арифметики Пеано.

Вот нужная для «превращения» чисел ещё и в строки аксиома (один из бесконечного количества возможных вариантов подобной аксиомы), которую можно при желании разбить на отдельные аксиомы – отдельная аксиома из каждого члена следующей конъюнкции.

Условимся только, что знаки неравенства (кроме знака $\neq$) – это сравнение числовых величин, поэтому от типа чисел-строк они не зависят. Поэтому вместо i_U можно писать i , если вопрос касается только неравенств (кроме знака $\neq$) и функций, зависящий от числовой величины своего аргумента, но не от его типа.

(S_U) Аксиома связи чисел и строк:

$$(i \leq U \Rightarrow i_U = e_U(i))$$

$$\wedge (i < U \Rightarrow e_U(i)' = e_U(i'))$$

$$\wedge (0_U = U_U \Rightarrow i'_U = i_U \cdot e_U(0))$$

$$\wedge (0 < U \Rightarrow e_U(U)' = (0') \cdot e_U(0))$$

$$\wedge ((k_U \neq 0_U \wedge i < U) \Rightarrow (k_U \cdot e_U(i))' = k_U \cdot e_U(i'))$$

$$\wedge (k_U \neq 0_U \Rightarrow (k_U \cdot e_U(U))' = (k'_U) \cdot e_U(0))$$

$$\wedge (E_9(\ominus_U) = U_9)$$

Пояснения:

1. Будем считать алфавит $e_U(i)$ чисел-строк типа U состоящим из одних только цифр (информация о размере алфавита типа U представлена в последней конъюнкции данной аксиомы), притом ноль – первый элемент $e_U(0)$, а самая большая цифра – последний не пустой элемент $e_U(U)$.

2. Тип в рассматриваемой сейчас аксиоматизации соответствует $(U + 1)$ -ичной системе счисления для $U > 0$, и записи числа в стандартной интерпретации для $U = 0$. Например, для $U = 9$ получается аксиома для увеличения на 1 при десятичной записи числа.

3. Первый член конъюнкции аксиомы $(i \leq U \Rightarrow i_U = e_U(i))$ как раз говорит о том, что числа-строки типа U от нуля до числа, равного U , равны соответствующим цифрам (символам).

4. Второй член конъюнкции $(i < U \Rightarrow e_U(i)' = e_U(i'))$ аксиомы говорит о том, что увеличение на 1 числа, меньшего U (а такое число записано одной цифрой) даёт число, которое равно следующей цифре.

5. Третий член конъюнкции $(0_U = U_U \Rightarrow i'_U = i_U \cdot e_U(0))$ аксиомы говорит о том, что увеличение на 1 числа, записанного в стандартной интерпретации ($0_U = U_U$) означает всего лишь приписывание ещё одной цифры. Тут мы видим, что такое «стандартная интерпретация» для числа i . Это строка $000 \dots 00$, в которой единственный символ (цифра) алфавита «0» повторяется $i + 1$ раз.

6. Четвёртый член конъюнкции $(0 < U \Rightarrow e_U(U)' = (0'_U) \cdot e_U(0))$ аксиомы говорит о том, что увеличение на 1 числа, равного максимальной цифре (если речь не о стандартной интерпретации – поэтому условие $0 < U$), даёт число-строку «10», условно говоря.

7. Пятый член конъюнкции $((k_U \neq 0_U \wedge i < U) \Rightarrow (k_U \cdot e_U(i))' = k_U \cdot e_U(i'))$ аксиомы говорит о том, что увеличение на 1 числа, которое состоит из нескольких цифр и не начинается с нуля, но начинается всё же на число $(k_U$ – это именно число по принятому нами соглашению о буквах, а не строка вида «01», например, которая числом-строкой не является), и чья последняя цифра с номером i меньше U , даст почти то же число-строку, но последняя цифра будет с номером $i + 1$ вместо i .

8. Шестой член конъюнкции $(k_U \neq 0_U \Rightarrow (k_U \cdot e_U(U))' = (k'_U) \cdot e_U(0))$ аксиомы говорит о том, что увеличение на 1 числа, которое состоит из нескольких цифр и не начинается на ноль, и чья последняя цифра с номером U – означает такое число-строку, где в результате инкремента вся часть до последней цифры (не включая её) заменяется на эту же часть, увеличенную на 1, а последняя цифра будет с номером 0 вместо U . Этот пункт годится и для стандартной интерпретации, а третий член конъюнкции можно, поэтому, заменить менее сложным $(0_U = U_U \Rightarrow 0'_U = 0_U \cdot e_U(0))$. Но пока вопрос оптимизации аксиом не приоритетен, зато с точки зрения понимания удобно, когда третий член конъюнкции содержит всю информацию об инкременте стандартной интерпретации.

9. Седьмой и последний член конъюнкции $(E_9(\Theta_U) = U_9)$ аксиомы говорит о том, что алфавит в данном случае состоит только из цифр – о чём было сказано в п.1 данного списка.

Надо ещё отметить, что типированные переменные можно рассматривать подобно типированным кванторам – вместо формулы вида:

$$A(a_U)$$

Мы рассматриваем формулы вида:

$$\Theta_U = s(a, 0) \Rightarrow A(a)$$

В дальнейшем мы рассмотрим подробнее функцию взятия символа $s(a, i)$ с места i строки a . Удобно считать, что значимые символы (то есть, когда это не пустая строка) начинаются с 1-й позиции с точки зрения функции $s(a, i)$, а на нулевом месте строки – всегда пустая строка того же типа, что сама строка. Такое соглашение не вносит логических искажений в понимание строк, но удобно для работы. Если бы у нас отсчёт значимых символов в строке начинался с места 0 для некоторого аналога функции $s(a, i)$, то и в этом случае мы могли бы определить описанную сейчас функцию $s(a, i)$. Напоминаю, что у второго аргумента функции $s(a, i)$ не указан тип (это число-строка некоторого типа) потому что смена типа числа-строки (при сохранении неизменной числовой величины данного числа-строки) не изменяет результат данной функции.

Аналогична ситуация со способом отличить между собой строки и числа-строк. Используемые нами числа-строки мы можем отличить от строк, которые не являются числами. В разбираемом нами примере записи чисел-строк они либо состоят из одной и той же цифры «0» – для стандартной интерпретации (тип 0). Либо на цифру ноль начинается только ноль (если тип числа-строки не 0), а в остальном строка должна быть конечной и состоять из одних только цифр и тогда это число-строка. Но соглашения о записи чисел-строк могут быть разные и для них должны доказы-

ваться разные способы проверки строки на то, что она является числом-строкой. Пусть предикатом проверки будет $N(a)$, например. Тогда

$A(i)$

Обозначает формулу вида:

$N(i) \Rightarrow A(i)$

Похожее соглашение есть в теории множеств, где строковые буквы используются для обозначения переменных множеств (частный случай классов), а для обозначения переменных классов используются прописные буквы.

Надо немного «перевести дух», взглянув обзорно и несколько отстранённо на то, что сделано до настоящего момента в данной статье, и какие задачи в данной статье и за её рамками предстоит решить.

Сначала определимся с тем, как мы будем называть ту теорию для строк и вычислимых натуральных чисел-строк, построение которой мы пытаемся начать в данной статье. Возьмём «арифметику строк» рабочим названием для данной теории на нынешнем начальном этапе её построения. Арифметику Пеано и её арифметические расширения (включая арифметику второго порядка, которая ещё будет упомянута в данной статье) будем называть «абстрактная арифметика».

Вместо названия «абстрактная арифметика» воздержимся от использования каких-либо уничижительных названий вроде «наивная арифметика», «не вычислимая арифметика» и т.п. Мне не понятно, кстати, что такого «наивного» было в первоначальной теории множеств Кантора? Он был гениален и шёл впереди всех математиков своего времени. Да, его теория была ещё не формальная, так математики тысячи лет работали с числами тоже без теории первого порядка, но это не значит, что всех предшественников надо называть «наивными». Пифагор, Евклид, Ньютон, Эйлер, Ферма, Лейбниц, Гаусс и т.п. – наивные? Скорее те, кто стал бы их так называть – показали бы своё не умное высокомерие.

Не вижу проблем, чтобы называть первоначальную теорию множеств Кантора – «первоначальная теория множеств Кантора». Как есть в физике «модель атома Резерфорда», «модель атома Бора» и т.п. – уважительные, без всяких уничижающих эпитетов названия для тех умственных обобщений, которые строили великие учёные по имевшимся у них к тому моменту фактам.

И парадоксы, которые были обнаружены при исследовании первоначальной теории Кантора – вовсе не часть его теории. Частью теории было требование того, чтобы противоречивые конструкции не являлись частью этой теории. Да, их порой было трудно выявить, так это нормально для не формальной теории со значительными (и в чём-то избыточными) «степенями свободы». И ведь все эти противоречия выявили, разобрали, а затем и предложили тщательно проверенные формализованные способы построения теории множеств. Но – в той логике, и в той «Вселенной», которую открыли Кантор и его первоначальная теория множеств

Во-первых,

В арифметике строк нам потребуется построить аналог частичных рекурсивных функций – теперь для строк (включая числа-строки). Очевидно, что для этого нет никаких принципиальных трудностей, потому что арифметика строк выразительней абстрактной арифметики и должна бу-

дет (после своего построения) обладать необходимыми для вычислимости свойствами. Эта задача не может быть решена в данной статье из-за своей громоздкости, но при её решении можно использовать готовые методы и архитектуру доказательств для частичных рекурсивных функций абстрактной арифметики.

Во-вторых,

Для арифметики строк нет теперь необходимости построения какого-либо «соответствия» между числами и строками, которое строили между числами абстрактной арифметики и записями чисел/строк на ленте Машины Тьюринга. Теперь числа-строки и строки являются объектами одной и той же теории.

В абстрактной арифметике есть свойства для подсчёта количества объектов. Слишком абстрактные и невычислимые в силу этого свойства, но зато они общие для самых разных способов подсчёта. А в арифметике строк подсчёт оказывается применимым ещё и к размерам самих объектов теории. И это открывает доступ (через конечность алфавита, из которого составлены строки и числа-строки) к конечности объектов, к их конкретизации и их вычислимости.

За счёт конкретизации в арифметике строк удастся, видимо, заменить аксиомы для суммы и произведения на определения. Ведь теперь есть потенциальная возможность доказать, какой будет число-строка i после применения к ней j раз операции инкремента. Это и будет определение суммы $i + j$ на базе доказательства (притом конструктивного доказательства) «существует и единственно» для результата операции. Аналогично – и для произведения.

Намеченное в предыдущем абзаце тоже – не для данной статьи. Ведь тогда пришлось бы в аксиомах о взятии символа с соответствующего места строки (они ещё будут приведены в данной статье) ограничиться операцией инкремента. Но гораздо проще и наглядней использовать в нашем случае первоначального построения теории готовые арифметические операции при написании аксиом. Поэтому переход от аксиом к определениям для арифметических операций мы вынесем за рамки данной статьи, что вовсе не отменяет факта, что между арифметическими операциями и некоторым строковыми функциями, аналогичными «сложению в столбик» или «умножению в столбик» имеется однозначное доказуемое соответствие в арифметике строк.

Интересно, что некоторые строки не являются числами-строками. Например, если алфавит для данного типа содержит не только цифры. Возможна и ситуация, в которой для данного типа строк вообще не заданы ни способ записи нуля, ни изменение при инкременте. Тогда индукция и другие числовые аксиомы не применимы к подобным строкам.

Такие «не числовые» строки соответствуют, по сути, логике отрицания числовых свойств. И в этом отрицании нет никакого «интуиционизма» Брауэра или каких-то иных надуманных ограничений на «допустимые» в математике методы. Просто «в природе» не всё является числами.

Всегда верны только общелогические аксиомы и теоремы. Что касается любой другой непротиворечивой логики (аксиоматики из замкнутых утверждений 1-го порядка, например), то имеются как соответствующие ей интерпретации, так и такие, которые соответствуют отрицанию этой логики. Сказанное касается и чисел, и множеств, и любых других не общелогических объектов. Тут я неявно отвечаю на один мой спор о том, почему я не использую теорию множеств в построении

теории для (вычислимых) натуральных чисел, и почему в теории о (вычислимых) натуральных числах имеется, якобы, «интуиционизм». Вот почему.

В-третьих,

Для арифметики строк есть логика (притом конструктивная логика) для конечных вычислений с тем, что мы называем «натуральные числа». Я тут отделяю «натуральные числа» от «чисел абстрактной арифметики». Первые мы считаем вычислимыми, а у вторых нет некоторых необходимых для вычислимости свойств.

Сейчас мы раскроем необходимые для вычислимости свойства, о которых упоминалось при обсуждении аксиомы (S_1) в предыдущем разделе.

Логика конечности в разбираемом случае состоит в том, что строка одномерна и у нас есть функция взятию символа с любого места строки. А вопрос о (не)равенстве строк сводится к вопросу о (не)равенстве символов на соответствующих местах этих строк. А символов – конечный набор. И равенство между строками (или отрицание равенства) должно решаться в рамках конечного размера (конечного количества мест с символами в строке), иначе про вычислимость речь вообще не идёт. Это всё отличает вычисляемые числа-строки от чисел абстрактной арифметики, которых – бесконечное количество и они не имеют (не аксиоматизирована) одномерной конечной структуры внутри числа абстрактной арифметики для его конечного сравнения с другим числом абстрактной арифметики.

Что касается бесконечного времени сравнения двух символов из конечных алфавитов (такое тоже можно представить), то логика тут не отличается от конечного времени сравнения 2 символов. Они всё равно либо равны, либо не равны. Поэтому у нас всегда есть конечный (вычисляемый) эквивалент для сравнения 2 символов из конечного набора – так устроена наша реальность.

То есть, сравнение между объектами конечного набора всегда имеет интерпретацию с конечным временем на операцию сравнения без потери общности с точки зрения логики теории. В этом, можно считать, и состоит искомый результат при построении логики конечности, а не в том, чтобы были невозможны невычисляемые (с бесконечным временем сравнения между объектами из конечного набора) интерпретации.

Можно, конечно, утверждать, что для чисел абстрактной арифметики тоже всегда есть «конечный эквивалент». Но проблема в том, что этот «конечный эквивалент» неотличим от «бесконечного эквивалента» в рамках абстрактной арифметики. Потому что в абстрактной арифметике нет той логики, которая позволила бы отделить интерпретации с логикой конечных (или как у конечных) интерпретаций от тех, которые не могут иметь вычисляемых аналогов. И мы это видели, когда рассматривали интерпретации абстрактной арифметики с бесконечными числами-строками.

И ещё один вывод можно сделать из вышесказанного: мы строим теорию для вычисляемых чисел-строк и строк, но она недостаточно выразительна для теории алгоритмов. Мы можем утверждать, что у нас есть «конечный эквивалент» для вычисления – когда необходимое вычисление для строк и/или чисел-строк действительно имеет вычисляющий их (за конечное время) алгоритм в реальном мире. Но мы не можем рассчитывать или оценивать необходимое для вычисления время в рамках арифметики строк. Если, конечно, мы не расширим нашу теорию соответствующим образом. Но в

этой статье мы этого делать не будем.

В-четвёртых,

Нет никакой необходимости в «гёделизации». Нет никакой нужды в том, чтобы сводить строки к числам абстрактной арифметики. Что не отменяет доказательств Гёделя – они опираются далеко не только на арифметику, а с опорой на язык и теорию для строк и чисел-строк построения Гёделя должны стать значительно проще. В арифметике строк строки присутствуют наравне с числами-строками и нет нужды записывать одни при помощи других.

Идея «гёделизации» в том, чтобы реализовать исследование свойств текстов формул и текстов доказательств средствами математики, для чего записать тексты исследуемых формул и тексты доказательств посредством таких математических объектов, для которых есть теория и которые вычислимы – потому что необходимы финитные (по размеру и времени проверки) объекты. Поэтому Гёдель «кодировал» тексты формул и тексты доказательств посредством их «гёделевых номеров» – которые являются числами абстрактной арифметики.

Сейчас понятно, что числа абстрактной арифметики не годятся, потому что не обладают свойствами финитности. Но принципиальные идеи доказательств Гёделя от этого никак не нарушаются – меняется лишь легко заменяемый технический метод на менее сложный. Надо использовать арифметику строк и записывать тексты формул и тексты доказательств как есть – при помощи символов.

Было даже странно, что для представления текстов формул используются не тексты формул. Но работа по переписыванию многих доказательств с «гёделизацией» – весьма громоздкая и не может быть выполнена в рамках данной статьи.

В-пятых,

Необходимо будет переписать доказательства о «взаимозаменяемости» рекурсивных функций с вычислимостью на абаке (на бухгалтерских счётах) и т.п. В данном случае частичные рекурсивные функции строк (включая числа-строки) можно будет построить так, чтобы они выполняли те же вычисления, что и их аналоги на абаке (на бухгалтерских счётах) и т.п. Притом для любого способа исполнения алгоритма он будет одинаково вычислимым, либо не вычислимым для любого из рассматриваемых способов. Вот только про машину Тьюринга придётся сказать отдельно в следующем пункте.

В-шестых,

Придется отказаться от машины Тьюринга как модели вычислений и заменить её моделью вычислений, которая гарантированно обладает необходимыми для вычислимости свойствами, и которая в гораздо большей степени соответствует нынешним вычислительным машинам.

Дело в том, что программа машины Тьюринга – в отличие от её данных – хранится не на ленте машины Тьюринга, а задаётся (в частности, но не исчерпываясь) некоторым «перечислением» множества состояний q_i и «набором правил» перехода от каждой возможной конфигурации

$\langle q_i, a_j \rangle$, где a_j – «символ» данных, записанных в текущей ячейке данных,

К состоянию

$\langle q_{i1}, a_{j1} \rangle$, где a_{j1} – «символ» данных, который теперь записан в текущей ячейке данных вместо

a_j .

При этом происходит либо переход от текущей ячейки данных: либо к следующей за ней, либо к предыдущей за ней. Либо текущая ячейка данных не меняется.

Если же нет правила по переходу от данной конфигурации к следующей, то машина останавливается и тем заканчивает вычисление.

Возникает вопрос: а что при такой конструкции машины Тьюринга гарантирует, то, что она обладает необходимыми для вычислимости свойствами?

Вопрос, как мы выяснили выше, совершенно не праздный. Потому что если «перечень множества состояний» и/или «набор правил перехода» опирается на числа абстрактной арифметики, то необходимых для вычислимости свойств у такой машины нет.

Нет смысла дискутировать о том, кто что имел в виду в формулировках про «перечни» и «наборы» – раз это не практически понятный (притом однозначно понятный в принципиальном отношении) способ построения вычислительного устройства, либо не некоторые объекты теории первого порядка. Притом, что теорий первого порядка с вычислимыми числами, например, просто не существовало (как теперь ясно), когда писались книги с описанием машины Тьюринга как модели для вычислений.

Методически правильно – на мой взгляд – относиться к программе с той же «практичностью», что и к данным. В возможностях своих теорий мы можем ошибаться, а вот пригодный для практической реализации «инженерный» проект можно реализовать в «железе» и проверить хотя бы в рамках некоторых ограничений. И «мысленные эксперименты» с тем, что можно реализовать на практике, гораздо в большей степени поддаются проверке и обнаружению ошибок, чем какие-то абстрактные умственные конструкции.

У меня есть готовая модель для вычислений в несколько десятков страниц, но пока она выложена в ЕДРИДе и нуждается в рецензировании. Программа в предлагаемой Машине – это данные на одном из «лучей данных». Программа представляет собой программный текст, состоящий из команд с разделителями. На другом «луче данных» есть счётчик, соответствующий номеру ячейки, с которой начинается текущая команда. В процессе выполнения программы этот счётчик меняется. Остальные данные расположены на других «лучах данных». Все «лучи данных» сходятся в одной «точке» – в которой расположен процессор – заменяющий собой «тележку Тьюринга» и гораздо больше соответствующий распространенной сейчас архитектуре вычислительных машин, чем «тележка».

При построении данной Машины пришлось решить вопрос неограниченной адресации ячеек – ведь для математических нужд нам необходимы бесконечные лучи данных, а на практике размер памяти ограничен максимальной разрядностью в номере адреса ячеек памяти для данной машины. Решить это было интересно и не слишком сложно – теперь есть способ на практике так организовать архитектуру вычислительной машины, что можно неограниченно наращивать её память без изменения «предельной разрядности» в нумерации адресов.

Ещё интересное наблюдение в том, что времени для сортировки на «луче данных» требуется не меньше, чем порядка N^2 единиц (где N – количество сортируемых элементов) даже для быстрых

алгоритмов сортировки – если учитывать ограниченность скорости света. Это расходится с тем, что пишут в учебниках по программированию. Но в рамках ограниченной памяти, где время доступа к любой ячейке данных можно считать ограниченным некоторой константой – в таких рамках действительно получаются оценки для времени быстрой сортировки в духе $N \times \lg(N)$. Но для больших данных предположение «доступ к любой ячейке данных требует примерно одного и того же времени» является слишком грубым и надо учитывать «геометрию» расположения данных и скорость сигнала для доступа к ним.

В-седьмых,

Тезис Тьюринга-Чёрча остаётся в силе, но теперь – применительно к частичным рекурсивным функциям арифметики строк и новому варианту модели вычислительной машины.

3 Статья «String Theory»

Что же так долго (примерно 150 лет) сбивало с толку математическое сообщество и не позволяло ему увидеть недостаточную выразительность своего понимания (формального и неформального) в отношении вычислимости?

В качестве хорошего примера возьмём статью [String Theory. John Corcoran; William Frank; Michael Maloney. The Journal of Symbolic Logic, Vol. 39, No. 4 \(Dec., 1974\), 625-637.](#)

В статье String Theory разобраны 2 попытки аксиоматизации теории строк – Гермеса и Тарского. Обе теории оказались эквивалентными («синонимичными») арифметике.

Желающие могут посмотреть саму статью подробно, но тут мы остановимся на аксиоматике Гермеса. Аксиоматика Тарского эквивалентна, поэтому нет принципиальной необходимости в её рассмотрении. Аксиоматика Гермеса:

§2. Successor axioms S_n [Hermes].

<Аксиомы Гермеса построения строк-потомков>

SA1: $[&i < j] (s_i x \neq s_j y)$

The successors have disjoint ranges.

SA2: $[&i] (s_i x \neq 0)$

SA3: $[&i] (s_i x = s_i y \Rightarrow x = y)$

SA4: $\forall P [(P0 \& \forall x (Px \Rightarrow [&i] Ps_i x)) \Rightarrow \forall y Py]$ Successor induction.

Конец цитаты.

Отметим, что в теории Гермеса вместо конкатенации символа со строкой используется функция s_i , дописывающая символ с номером i из алфавита размером n символов к значению аргумента этой функции – строке x . Дописывает к началу, или концу, или ещё каким-то стандартным образом – не имеет значения. Способ всегда один и тот же – большего из логики данных аксиом получить нельзя, потому что им соответствуют и те, и другие интерпретации. Обозначается это так:

$s_i x$

Число ноль в обеих теориях соответствует пустой строке, и в приведённой аксиоматике это видно.

Было доказано (не в самой статье String Theory, но там даны ссылки на соответствующие работы), что теория $S1$ «синонимична» арифметике второго порядка. В конце раздела будут приведены некоторые технические подробности – в том числе и об арифметике 2-го порядка. Но арифметика второго порядка – это всё равно абстрактная арифметика. И, наконец, оставалось доказать «синонимичность» произвольного S_n с $S1$, после чего логика строк свелась бы (по мнению авторов статьи) к абстрактной арифметике. Был один технический усложняющий нюанс, который решился при помощи переписывания обозначений в теории $S1$ и получения теории $S0$ без усложняющего нюанса. Мой слабый перевод с английского из параграфа 4 статьи String Theory об этом:

Проблема заключается в том, что поскольку $S1$ и S_n ($n \geq 2$) разделяют функциональный символ $s1$, используемый "в разных смыслах они совместно несовместимы. Например, в $S1$ каждый ненулевой объект имеет форму $s_1 x$, но в S_n ($n \geq 2$) некоторые ненулевые объекты не имеют этой формы (например, $s_2 0$). Это препятствие решается очевидным образом, используя традиционное

понятие логической формы $\langle \dots \rangle$. Две теории T и T' определяются в одной и той же (логической) форме, если существует взаимно однозначное, сохраняющая категорию соответствие от множества значимых слов одной к множеству значимых слов другой теории, которое переводит теоремы одной теории в теоремы другой и наоборот.

Затем определим, как одна теория может быть интерпретирована в другой, если некоторая теория в той же логической форме, что и первая, может быть получена в дефиниционном $\langle$ используя только определения $\rangle$ расширении другой.

Для выполнения вышеуказанной цели мы предполагаем новую теорию S_0 , которая вытекает из S_1 путем замены всех вхождений s_1 новым функциональным символом s .

4.1. S_0+n : интерпретация S_n в S_0 . Для целей терминологии думайте о S_0 как об арифметике. Конец цитаты.

По аксиоматике сразу видим принципиальное упущение – нет логики нарушения равенства от строковых свойств. Строки разных типов не рассматриваются как всегда отличные (не равные) друг другу. В первом разделе тут уже было обосновано, почему в арифметике нет необходимых для вычислимости свойств. Обосновывалось это для любых случаев – включая канторовскую нумерацию пар, когда одно число абстрактной арифметики в паре отвечает за тип, а другое – за собственно число. А в статье String Theory построения гораздо проще, чем в разделе 1 выше, что делает статью String Theory ещё более показательным примером в плане упущенных свойств вычислимости, так как авторы считают, что разные типы «синонимичны» друг другу на уровне арифметики даже при взаимно-однозначном соответствии чисел-строк данного типа с числами абстрактной арифметики.

Это гораздо проще опровергать, чем то, что рассматривалось в 1 разделе выше.

Итак, идея доказательства String Theory в том, что за основу берутся одни те же числа-строки (притом все числа-строки) данного «базового» типа S_0 , «синонимичного» арифметике. Из них «назначены» тоже все числа-строки типа A (взаимно-однозначно с «базовым» типом) и типа B (тоже взаимно-однозначно с «базовым» типом). Поэтому имеется бесконечное количество пар строк типа A и типа B , которые равны друг другу – так как являются одной и той же строкой «базового» типа. Но это противоречит аксиомам нарушения равенства из-за строковых свойств. Противоречит тому, что никакая строка типа A не может быть равна никакой строке типа B кроме случая, когда типы A и B полностью совпадают.

Соответственно, нет возможности корректно получить из такой «эмуляции» типа A и типа B инкапсулированные в настоящие типы A и B свойства – размер алфавита для данного типа, зависящий от этого размер строки, правильную последовательность чисел-строк от нуля с переходами при инкременте от текущего к следующему числу-строке и т.п.

Особенно показательно (чуть ниже поясню, почему) то, что в аксиоматике Гермеса нет инкремента (функции увеличения на 1). То есть, последовательность чисел совершенно не однозначна. Можно, например, изменив порядок расположения разрядов по старшинству на противоположный – когда цифры расположены по нарастанию соответствующей им разрядности и число сто двадцать три будет записываться как 321, а не обычным – 123 – способом. И такой способ записи тоже соответствует всем аксиомам Гермеса.

Такая неоднозначность порядка имеет место и при «обычной» аксиоматике абстрактной арифметики – о чем было сказано тут в первом разделе. Но она совершенно неприемлема для логики вычислимости – где необходим детерминированный алгоритм получения следующего числа-строки из заданного числа-строки. И именно на противопоставлении неоднозначности абстрактной арифметики и детерминированности вычислений строится одно из доказательств недостаточной выразительности абстрактной арифметики для рассмотрения вопросов вычислимости выше (в первом разделе).

Тут мы видим особенность мышления математиков – которое, как правило, полностью опирается на принятые в математическом сообществе теории (формальные и неформальные) при игнорировании реальности, если она расходится с этими теориями. Впрочем, проверка на «расходится» даже не проводится, потому что имеет место полное доверие к устоявшимся в математическом сообществе (которые уже не обсуждаются) теориям и пониманиям.

Похоже, что логика строк без аксиом нарушения равенства совпадает (или близка) с абстрактной арифметикой, если считать (и это ошибка) ноль общим у разных типов «чисел-строк», и, если каждая «строка» является ещё и «числом-строкой». А индукция в статье String Theory используется та же (помимо квантора по предикатам – но это лишь способ перейти к логике второго порядка), по сути, что и в арифметике Пеано – перебор ведётся по всем «числам-строкам» (что означает и по всем «строкам» в данном случае).

Изложенное в предыдущем абзаце – это некоторая «эмуляция» чисел-строк. Недостаточную выразительность этой эмуляции при помощи абстрактной арифметики мы уже обсудили в первом разделе, начиная с абзаца «В чём же принципиальное упущение арифметики Пеано. . . » и несколько дальше (там мы рассматривали пример с записью числа в десятичном $1 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$ и двоичном виде $1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$). И построения в статье String theory вполне соответствуют упомянутому обсуждению с некоторыми дополнительными – но не принципиальными – нюансами.

Правильный же подход в том, что когда мы рассматривает не один, а хотя бы два разных типа строк (и чисел-строк), то необходимо добавлять логику неравенства чисел-строк разных типов. Нули разных типов при этом не будут равны. И аксиома индукции тогда оказывается другой – с типированными кванторами вместо «обычных».

Аксиома индукции с типированными кванторами – отличается от того, что считалось правильным в математическом сообществе со времен арифметики Пеано. Разумеется, математикам с их способом мышления, было практически невозможно подумать о том, что принятый ими способ понимания индукции чисел несовместим с вычислимостью, если выйти за рамки одного способа записи чисел.

Хотя, надо отметить, что публикация теории Пеано (1889 г.) по меркам истории математики случилась почти вчера. Поэтому добавление и смену некоторых аксиом, чтобы получить необходимые для вычислимости свойства, можно считать продолжением работы по аксиоматизации для работы с числами (а теперь ещё и со строками), которая проходила с 1860-х годов и лишь ненадолго (по меркам истории математики) прервалась из-за экстремальных нематематических событий в

окружающем математиков мире.

Ещё я обещал пояснить некоторые «технические моменты» из статьи String Theory.

Интересен способ такой записи чисел, чтобы каждая строка была ещё и числом-строкой. Рассмотрим для алфавита из цифр 0-9. За ноль берется пустая строка, а для цифр 0-9 строка «0» будет 1, а строка «9» будет означать 10. Тогда «00» будет 11, а «0000» будет означать 1111 (везде без кавычек мы пишем обычную десятичную запись числа) и т.д. Если же вместо числа из цифр «0» стоят ещё и другие цифры в конце данного числа, то мы просто добавляем число, начинающееся не с «0» как десятичное. Так «00123» будет $11111 + 123$. Например, переход от «0000» к «00000» это $1111 + 9999 + 1 = 11111$.

Приведём информацию об арифметике 2-го порядка из «Вычислимость и логика» Булоса и Джеффри. Обозначим через Ind следующее предложение 2-го порядка:

$$\forall P ([P(0) \wedge \forall x (P(x) \Rightarrow P(x'))] \Rightarrow \forall z P(z))$$

Теперь образуем конъюнкцию этого предложения со всем аксиомами Пеано кроме аксиомы индукции и (q_3) – так как Ind заменяет собой индукцию 1-го порядка, а (q_3) вытекает из аксиомы индукции. Полученную конъюнкцию обозначим PA . И это предложение есть аксиоматизация арифметики, притом это полная теория для языка абстрактной арифметики, но, увы – это теория 2-го порядка со всеми её минусами – включая отсутствие финитных доказательств для бесконечного количества истинных утверждений, и даже отсутствие эффективной аксиоматизации с точки зрения аксиом 1-го порядка. Подробности и доказательства сказанного о предложении PA – в книге «Вычислимость и логика» Булоса и Джеффри.

В статье String Theory представлена информация о том, что арифметика 2-го порядка (то есть PA) «синонимична» теории Гермеса S_n – при любом n . И мы видим, что в теории Гермеса нет ни инкремента, ни операций сумма с произведением. Что лишний раз подчёркивает, насколько эти арифметические операции не конкретно заданы в абстрактной арифметике и насколько они в абстрактной арифметике далеки от необходимых для вычислимости свойств.

В заключении попробую обобщить те причины, по которым математикам не удалось в процессе формализации арифметики в 19 веке и позже – при исследованиях о вычислимости – формализовать или хотя бы не формально, но верно уловить логику вычислимости.

Я приводил доводы в первом разделе, что период логических открытий был прерван внешними – по отношению к математике – событиями насильственного характера, и у меня нет сомнений, что достигнутый перед этим творческий подъём немецкого математического сообщества многое ещё дал миру, если бы он не был прерван. Но я сильно сомневаюсь, что даже при сохранении того уровня математической силы, которого достигли математики до прихода Гитлера к власти, они смогли бы сами обнаружить отсутствие необходимых для вычислимости свойств в построенной ими арифметике.

По моему субъективному мнению, у многих (возможно, практически у всех) сильных математиков есть профессиональная деформация «формализма» (назовём это так), когда математик видит актуальный для него вопрос в разрезе той математической теории, в рамках которой он этот вопрос исследует. Даже если формальной теории нет (для арифметики несколько тысяч лет не было фор-

мальной теории), математик все равно действует в рамках тех умственных конструкций, которые приняты в математическом сообществе и сознательно, или в силу незаметного (из-за его постоянства), но очень мощного влияния математического сообщества, изгоняет из своих умственных конструкций «лишние» - пусть и соответствующие реальности – особенности.

Довольно хорошо известны примеры разных подходов к исследованию физических вопросов физиков и математиков в теории относительности, например. И хоть можно показывать пальцем на формулы математиков, но заявить о их соответствии реальности (выбрав среди разных формализмов то, что соответствует реальности) выпало на долю физика Эйнштейна. Оставим популярные у любителей споры – никаких претензий о приоритете к Эйнштейну ни Лоренц, ни Пуанкаре не предъявляли, и ничто не мешало после вполне дружескому общению Эйнштейна и Лоренца, а формулы Эйнштейна (иногда называемые формулами Эйнштейна-Гильберта) по смыслу опять же принадлежат Эйнштейну – что публично признавал сам Гильберт.

И Эйнштейн был далеко не лучшим студентом у Минковского (о чём позже с удивлением писал Минковский своему другу Гильберту), пространство имени которого Эйнштейн позже использовал в теории относительности. А вот сам Минковский в переписке с Гильбертом шутил, что после общения с физиками ему потребуется время, чтобы «очиститься» и вернуться к математическому способу мышления. Я это понимаю, как возврат к более «формальному» мышлению – основанному на принятых у математиков умственных конструкциях и «очищенному» от всякого «реализма» – который либо должен быть формально выражен в умственных конструкциях, либо просто не нужен в процессе математического исследования.

Вычисления – это процессы вполне реальные. И нет ничего удивительного, что математики не смогли сами «вытащить» необходимую логику из реальности, чтобы «переселить» её в свои умственные конструкции. Это не их компетенция. Другое дело, что логика вычислимости необходима как часть математики (в отличие от той же теории относительности) и математики «заполнили пустоту» ошибочными умственными конструкциями.

Проблема в том, что способность достаточно глубоко понимать математические конструкции и их язык с одновременным приоритетом реальности (а не умственных конструкций) в мышлении – это «пограничное» сочетание того, что с трудом сочетается в одной голове. Но есть «пограничные» вопросы вроде вычислимости, которые будут ждать своего решения до тех пор, пока соответствующий «пограничник» не появится и (что ещё менее вероятно) не сможет реализовать свои возможности, встретив ожидающий его вопрос.

Поэтому вопрос о математической формализации вычислимости решается сейчас – в случайный момент, а не тогда, когда этот вопрос был поставлен впервые в математическом сообществе.

4 Остальные аксиомы для арифметики строк

Если у нас задан тип числа-строки, то за счёт изменения построения чисел (изменив порядок расположения разрядов по старшинству на противоположный, например) при «том же» алфавите мы можем получить новый тип чисел-строк (и строк). Даже при алфавите из единственного символа (не пустого элемента алфавита) мы можем создать новую последовательность чисел-строк из исходной. Но это будет уже не определение в математическом смысле. Ведь появились новые объекты, отличающиеся от исходных (в том числе и алфавит теперь считается другим для другого типа). В программировании (в объектно-ориентированном программировании) подобный способ создания новых сложных типов (классов) из других сложных типов (классов) является обычным. И называется определением (что не соответствует математическому пониманию этого термина) объектов класса – притом типичным приёмом является создание новых классов на основе других – ранее созданных – классов.

И, поскольку возможность создания дополнительного построения чисел-строк – при наличии хотя бы одного варианта построения чисел строк – является частью «природы» строк, то мы должны обеспечить в арифметике строк такую логику, чтобы в рамках условия (дедуктивной посылки) «если бы у нас был ещё один тип» у нас была та же логика, которая необходима, когда этот тип уже есть.

И строить свою аксиоматику мы должны с точки зрения её полной готовности для её расширения дополнительными типами чисел-строк (и строк), созданных «по образцу» (но не идентично, конечно) с имеющимися до этого типами (или типом) чисел-строк (и строк).

Сказанное означает, что с точки зрения готовности логики теории к событию «появился ещё один тип строк/чисел-строк» аксиоматика абстрактной арифметики является ошибочной. Ошибочной не в том смысле, что там есть противоречие (она не противоречива, как известно), а в том, что в аксиомах абстрактной арифметики используются «обычные», а не типированные кванторы. И в рамках одного типа чисел-строк (для интерпретаций с одним типом) логика таких аксиом не отличается от логики тех же аксиом, но с типированными кванторами. Но – при расширении количества типов в теории за пределы единственного типа – аксиомы с «обычными» кванторами перестают соответствовать верной логике аксиом с типированными кванторами. А «готовность» к такому расширению – должна быть в логике теории, обладающей свойствами вычислимости чисел-строк. И в этом смысле абстрактная арифметика является ошибочной.

Поэтому будем стараться формулировать аксиомы так, чтобы при появлении нового типа они соответствовали логике нарушения равенства из-за строковых свойств и другим необходимым для вычислимости свойствам, которые удастся заметить для чисел-строк отличающихся типов в реальном мире.

Пустая строка и концы строк

(1.1) $\forall a \exists \ominus_U s(a, 0) = \ominus_U$ – символы строки начинаются только с первой позиции. А нулевая позиция – «техническая», там всегда пустая строка (пустой элемент алфавита) того же типа, что и вся строка. Можно было бы располагать не пустые символы строки, начиная с нулевой позиции. В ИТ-сфере используют оба эти варианта в разных языках программирования. Но в любом случае

надо аксиоматизировать наш выбор. И несколько удобней для построения теории, как оказалось, начинать нумерацию значимых позиций строки с 1.

Напоминаю, что числовые значения без индекса типа в функциях означают, что значение функции не зависит от типа числа-строки, а зависит только от «числовой величины» данного аргумента. У нас есть аксиомы (среди аксиом нарушения равенства из-за строковых свойств), обеспечивающие преобразования между числами-строками разного типа с сохранением числовых величин преобразуемых значений. Это и создаёт «единый масштаб» для всех типов чисел-строк и строковых операций с использованием числовых значений.

Что касается конструкции вида:

$$\exists \ominus_U P(\ominus_U)$$

То из-за свойств пустой строки произвольного типа (что видно из аксиомы 1.1 и будет видно из дальнейших аксиом) она эквивалентна «обычной» формуле следующего вида:

$$\exists x (x = s(x, 0) \wedge P(x))$$

У этой формулы есть частный случай, для которого легко вывести:

$$(\exists \ominus_U f(x) = \ominus_U) \Leftrightarrow f(x) = s(f(x), 0)$$

Действительно, из рассмотренного соответствия для $\exists \ominus_U P(\ominus_U)$ имеем:

$$(\exists \ominus_U f(x) = \ominus_U) \Leftrightarrow \exists y (y = s(y, 0) \wedge f(x) = y)$$

Осталось доказать эквивалентность:

$$\exists y (y = s(y, 1) \wedge f(x) = y) \Leftrightarrow f(x) = s(f(x), 0)$$

Но последняя эквивалентность – известная в логике тавтология для подстановки в предикат нужного значения ($f(x)$ в нашем случае) свободной переменной, путём связывания этой переменной в конъюнкции с нужным значением под квантором существования:

$$Q(f(x)) \Leftrightarrow \exists y (y = f(x) \wedge Q(y))$$

Только вместо $Q(y)$ у нас здесь $y = s(y, 1)$

Казалось бы, в связи с доказанной эквивалентностью $(\exists \ominus_U f(x) = \ominus_U) \Leftrightarrow f(x) = s(f(x), 0)$ можно переписать аксиому (1.1) в виде:

$$\forall a s(a, 0) = s(s(a, 0), 0)$$

Но это было бы ошибкой. Потому что такая «аксиома» была бы верной и в том случае, если бы функция $s(a, 0)$ получала первый значимый символ с нулевого места. И в этом случае мы как раз не смогли бы получить те логические выводы, которые получили только что. Но из других аксиом (которые будут дальше) видно, что длина любого элемента алфавита как строки не больше 1. Функция $s(a, i)$ извлекает всегда некоторый элемент алфавита с места i строки a , поэтому, взятие 2-го символа в извлеченном элементе алфавита даст пустую строку. Это как раз способ получать пустую строку нужного типа независимо от того, с нулевой или 1-й позиции начинаются значимые символы строки для функции $s(a, i)$. Поэтому эквивалентной альтернативной записью для аксиомы (1.1) будет следующая:

$$(1.1') s(a, 0) = s(s(a, 0), 0'')$$

Рассмотрим заодно и конструкции вида:

$$\forall \ominus_U P(\ominus_U)$$

Она эквивалентна «обычным» формулам следующего вида:

$$\forall x (x = s(x, 0) \Rightarrow P(x))$$

Как видим, «обычные» формулы, соответствующие кванторам по пустым строкам разного типа, не содержат даже индекса типа.

(1.2) $\forall a \forall i ((\exists \ominus_U s(a, i') = \ominus_U) \Rightarrow s(a, i'') = \ominus_U)$ – если на не нулевой (i' -ой) позиции строки находится пустой элемент алфавита, то на следующей позиции (на i'' -ой) тоже будут тот же пустой элемент. И дальше – как легко догадаться – то же самое.

Из выведенного выше $(\exists \ominus_U f(x) = \ominus_U) \Leftrightarrow f(x) = s(f(x), 0)$ и того, что $s(s(u, i'), 0) = s(u, 0)$ можем переписать аксиому (1.2) в альтернативном виде:

$$(1.2') \forall a \forall i (s(a, i') = s(a, 0) \Rightarrow s(a, i'') = s(a, 0))$$

(1.3) $\forall i \forall \ominus_U i \neq \ominus_U$ – никакое число (включая ноль) не является пустой строкой. Эта аксиома тоже не является безальтернативной, но в ИТ-сфере принято именно так. Технически удобно иметь пустую строку в дополнение к обычной «нумерации» – есть, что поставить в соответствие бесконечности в качестве результата некоторых функций, например. Это мы увидим уже в следующем определении.

Если заменить часть аксиомы $\forall \ominus_U i \neq \ominus_U$ на эквивалентную $\exists \ominus_U i = \ominus_U$, то можно записать аксиому без не-числовых кванторов и индексов типа. По аналогии с (1.2') перепишем аксиому (1.3) в виде:

$$(1.3') \forall i (i \neq s(i, 0))$$

(1.4) Определение для длины строки $l_9(u)$ даём для десятичного представления результата – без ограничения общности:

$$(l_9(u) = 0_9 \wedge (\exists \ominus_U s(u, 1) = \ominus_U))$$

$$\vee (\exists \ominus_U s(u, l_9(u)) \neq \ominus_U \wedge s(u, l_9(u)') = \ominus_U \wedge s(l_9(u), 0) = \ominus_9)$$

$$\vee (l_9(u) = \ominus_9 \wedge \forall i \forall \ominus_U s(u, i') \neq \ominus_U)$$

Как видим, аксиома (1.3) позволяет нам для строки с бесконечной длиной иметь значение, которое отличается от любого числа, а для конечной длины иметь ровно то значение, которое равно количеству непустых символов в строке.

Если заменить часть аксиомы $\forall \ominus_U s(u, i') \neq \ominus_U$ на эквивалентную $\exists \ominus_U s(u, i') = \ominus_U$, то можно дать альтернативное определение без не-числовых кванторов и индексов типа.

(1.4') Определение для длины строки $l(u)$ переписанное по аналогии с (1.2')

$$(l_9(u) = 0_9 \wedge s(u, 1) = s(u, 0))$$

$$\vee (s(u, l_9(u)) \neq s(u, 0) \wedge s(u, l_9(u)') = s(u, 0) \wedge s(l_9(u), 0) = \ominus_9)$$

$$\vee (l_9(u) = \ominus_9 \wedge \forall i s(u, i') \neq s(u, 0))$$

Мы не будем в данной статье давать доказательство «существует и единственно» для результата функции $l_9(u)$. Мы пока переносим логику реальности в теорию, а выводится эта формула из других аксиом (является ли данная формула определением) или её логика независима, в силу чего данная формула является аксиомой – вопрос второстепенный на данном этапе исследования. Но логика данной формулы должна быть частью нашей теории – поэтому мы её записываем среди исходных формул теории.

Структура строк

Любая строка состоит из элементов алфавита в каждой своей позиции:

$$(2.1) \forall a \exists \Theta_U \forall i \exists j s(a, i) = e(\Theta_U, j)$$

Переписать эту аксиому по аналогии с (1.2') невозможно, потому что в области действия квантора $\exists \Theta_U$ есть ещё кванторы. Дело в том, что все элементы алфавита для данной строки должны быть из одного и того же алфавита.

Но, с другой стороны, очевидно, какого именно нужного типа «существует» пустая Θ_U строка в аксиоме (2.1). Она равна $s(a, 0)$. Поэтому и в данном случае имеется альтернативный вид у аксиомы (2.1) без кванторов по разным типам пустых строк:

$$(2.1') \forall a \forall i \exists j s(a, i) = e(s(a, 0), j)$$

Строка – это конкатенация своих символов:

$$a = s(a, 1) \cdot \dots \cdot s(a, i) \cdot \dots$$

Но мы запишем это бесконечное выражение как критерий равенства строк:

$$(2.2) a = b \Leftrightarrow \forall i s(a, i) = s(b, i)$$

Но эта аксиома ничего не говорит о том, как соединяются строки – а тем более символы, а в силу этого мы ещё не достигли логики для равенства:

$$a = s(a, 1) \cdot \dots \cdot s(a, i) \cdot \dots$$

как конкатенации своих символов. Поэтому нам необходимы ещё и аксиомы о конкатенации.

Соединение строк

В аксиомах о соединении строк везде в посылке нужно делать проверку на совпадение типов этих строк, потому что «нормальная» конкатенация – это операция для операндов с одинаковыми типами. Проверка совпадения типов операндов ниже – это сравнением символов с нулевых мест этих строк.

(3.1) Аксиома о результате конкатенации строк одинакового типа при конечной 1-й строке:

$$s(a, 0) = s(b, 0) \wedge l_9(a) \neq \Theta_9 \Rightarrow$$

$$(i_9 \leq l_9(a) \Rightarrow s(a \cdot b, i_9) = s(a, i_9)) \wedge (i_9 > l_9(a) \Rightarrow s(a \cdot b, i_9) = s(b, i_9 - l_9(a)))$$

Напоминаю, мы условились, что знаки неравенства (кроме знака $\neq$) – это сравнение числовых величин, поэтому от типа чисел-строк они не зависят. Поэтому вместо i_9 можно было бы писать i , везде, кроме вот этой части аксиомы (3.1):

$$i_9 - l_9(a)$$

Но и в этой части можно обойтись без типа числа строки i , если для записи разницы между числами использовать явное приведение типа:

$$((e_9(0_9)) i) - l_9(a)$$

Однако и эта аксиома не позволяет ещё нам утверждать, что равенство:

$$a = s(a, 1) \cdot \dots \cdot s(a, i) \cdot \dots$$

является конкатенацией своих символов.

Нам ещё надо обеспечить логику тождественного отображения элементов алфавита на самих себя функцией $s(a, 1)$. Такое тождественное отображение вовсе не возникает от того, что оно «должно

быть». Нам надо обеспечить это «должное», а не надеяться на то, что оно возникнет как-то, «само собой».

Аксиомой о равенстве мы внесли логику взаимно-однозначного соответствия для результатов функции $s(a, 1)$ с элементами алфавита любого типа. Потому что у нас будут заданы аксиомы для равенства для элементов алфавита, на которые опирается аксиома равенства для строк. Но любой элемент алфавита – это ещё и строка. Поэтому не должно быть противоречий между аксиомой равенства для строк, которая опирается на $s(a, i)$ и аксиомами равенства для элементов алфавита – а для этого результаты функции $s(a, 1)$ должны быть взаимно-однозначными на этих элементах. То, что результаты $s(a, i)$ всегда являются элементами алфавита – мы задали аксиомой (2.1), но мы никак не задали связь (кроме взаимной однозначности) между элементами алфавита и результатом применения к ним $s(a, 1)$.

У нас нет такой информации даже о пустых строках. Что означает, что мы пока что никак не позаботились («нарочно») даже о том, чтобы исключить хотя бы такие интерпретации, в которых «пустая строка» может иметь не нулевую длину. Потому что определение функции длины тоже строится на функции $s(a, i)$.

Следующую группу аксиом мы начнём с решения данного вопроса, а пока закончим запись аксиом в текущей группе.

(3.2) Аксиома о конкатенации при бесконечной 1-й строке или при разных типах строк:

$$s(a, 0) \neq s(b, 0) \vee l_9(a) = \Theta_9 \Rightarrow a = a \cdot b$$

Из критерия равенства строк и аксиом о конкатенации очевидна теорема об ассоциативности конкатенации:

$$(т.3.1) \quad (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

Алфавит

Аксиомы об «атомах», из которых образованы строки.

Аксиомы об элементах алфавита:

(4.1) $s(e(\Theta_U, i), 1) = e(\Theta_U, i)$, на 1-й позиции строки, соответствующей элементу алфавита (речь о любом алфавите), находится данный элемент алфавита.

Но даже теперь мы не можем быть уверены, что равенство:

$$a = s(a, 1) \cdot \dots \cdot s(a, i) \cdot \dots$$

является тем же, что конкатенация символов, из которых можно составить строку a .

При построении предыдущих аксиом мы такую цель обсуждали и ставили, но только после следующей аксиомы «нарочно» сделаем всё необходимые для истинности этой формулы и её понимания как конкатенации символов. То разумное, что необходимо и чего ещё нет – обычно само не возникает и его надо создавать «специально», опираясь на разум (известный, проверенный способ), а не полагаясь на какие-то «новые сущности» вопреки «бритве Оккама» (кстати, это один из аргументов против атеизма).

Примером неправильного подхода при работе (точнее, при отказе от такой работы) с реальностью служит статья String theory и разбираемые в ней необоснованные аксиоматизации, которые не были сопоставлены с реальностью и не были построены так («специально»), чтобы соответ-

ствовать реальности. Разумеется, эти теории реальности и не соответствуют. Понятно, что это не претензии к математикам, которых работе с реальностью в приоритете над теорией не учат – и не будут учить, потому что подобные знания практически всегда мешают математическому образу мысли.

Но сейчас надо подчинять теорию реальности и изменять её в соответствии с реальностью. Поэтому так делаем – и продолжим делать – до конца данной статьи, по крайней мере.

И вопрос о том, можно ли вывести некоторые аксиомы из других имеющихся аксиом, показав тем избыточность некоторой части аксиоматизации – напоминаю – для нас не приоритетен. Безусловным приоритетом для нас на данном начальном этапе построения теории является перенесение логики реальности в теорию. И если вывод некоторых свойств не очевиден, но они есть в реальности – то наш долг аксиоматизировать эти свойства. А окажется ли эта аксиоматизация избыточной или нет – не имеет принципиального значения на данном этапе исследования практики вопроса и первоначального построения теории. Все необходимое мы записываем. Если не можем сразу записать как теорему – записываем, как аксиому. Главное – отразить найденное и существенное из реальности в теории, чтобы после в теоретическом исследовании не упустить чего-нибудь принципиально необходимого для соответствия с реальностью.

(4.2) $s(e(\Theta_U, i), 2) = \Theta_U$, на 2-й позиции строки, соответствующей элементу алфавиту, находится пустой элемент алфавита, то есть – длина этой строки не превышает 1. Иначе может быть такой алфавит, например: «123», «23», «3». Тогда при конкатенациях и операциях $s(\dots)$ будет соответствие с аксиомами до данной группы включительно, но без (4.2).

(4.3) $\forall i \leq E_9(\Theta_U) : e(\Theta_U, i) \neq \Theta_U$, не пустые элементы алфавита нумеруются от нуля до $E_9(\Theta_U)$ включительно.

Главное в этой аксиоме (вместе со следующими в данной группе) с точки зрения вычислимости – наличие ограничивающего числа для размера алфавита любого типа. В первом разделе было разобрано, как связана ограниченность алфавита с возможностью конечных вычислений.

Выше уже упоминалось, что проверки на неравенство сравнивают числовые величины между собой независимо от их типа (можно считать, что оба типа приводятся к типу десятичной системы счисления, например), поэтому у переменной i не указан тип.

Для сравнения строк уместно определить функцию $\text{Comp}_9(a, b)$, которая для одинаковых значений – возвращает 0_9 , а для значений разных типов возвращает Θ_9 . А для значений одного типа, но не равных, возвращает 1_9 , когда значение 1-го аргумента меньше второго из-за меньшего номера в алфавите самого раннего (с наименьшим местом) не равного символа, чем у 2-го аргумента, либо из-за того, что в первом аргументе – пустая строка, в отличие от второго аргумента. В остальных случаях (2ой аргумент «меньше» 1-го) функция возвращает 2_9 .

Но подобные определения даны тут не будут из-за громоздкости как соответствующих формул, так и доказательств «существования и единственности». Для определения $\text{Comp}_9(a, b)$ потребовалось бы, например, определять ещё и $\text{CompIn}_9(a, b)$, которая ищет номер места, где символ в 1-м аргументе отличается от символа во 2-м аргументе. Еще – развивая тему поиска и сравнения – пришлось бы определять функцию поиска $\text{find}_9(in, what, i)$, и так далее.

(4.4) $\forall i \leq E_9(\Theta_U) \forall j \leq E_9(\Theta_U) : e(\Theta_U, i) = e(\Theta_U, j) \Rightarrow (e_9(0_9))i = (e_9(0_9))j$ – непустые элементы данного алфавита не повторяются в нём.

В данной аксиоме пришлось использовать явное приведение типов в заключительном равенстве, потому что равенство (в отличие от знаков неравенства) имеет фиксированный смысл и сравнивать числа-строки на равенство их числовых величин мы можем только при совпадении их типов.

(4.5) $\forall i > E_9(\Theta_U) : e(\Theta_U, i) = \Theta_U$, любой номер элемента алфавита, превышающий $E_9(\Theta_U)$, соответствует пустому элементу алфавита;

«Сплошные части» строк

«Сплошная часть» строки может быть конечной:

$$S(a, i, j) = s(a, i) \cdot s(a, i + 1) \cdot \dots \cdot s(a, i + j - 1), \text{ где } i > 0, j > 0.$$

Структура конечной (ограниченной аргументом длины) «сплошной части» строки описывается определением для $S(a, i, j)$ из трёх следующих частей:

$$(5.1) \ 0 < i \wedge 0 < k \leq j \Rightarrow s(S(a, i, j), k) = s(a, i + k - 1)$$

$$(5.2) \ 0 < i \wedge 0 < j < k \Rightarrow s(S(a, i, j), k) = s(a, 0)$$

Для удобства зададим результат функции $S(a, i, j)$ для нулевых значений:

$$(5.3) \ S(a, 0, j) = S(a, i, 0) = s(a, 0)$$

Структура неограниченной аргументом длины «сплошной части» строки описывается следующими двумя аксиомами:

$$(5.4) \ 0 < i \wedge 0 < k \Rightarrow s(C(a, i), k) = s(a, i + k - 1)$$

$$(5.5) \ s(C(a, 0), i) = s(C(a, i), 0) = s(a, 0)$$

Возможно, надо ещё добавить аксиому о существовании бесконечных строк, либо отрицание подобного утверждения. Но и того, что изложено – более чем достаточно для проверки и изучения логических следствий на начальном этапе построения арифметики строк. Пока в приоритете – сравнение с реальностью на соответствие и отсутствие принципиальных упущений, а вопрос о бесконечных строках едва ли уместен на этом этапе. К тому же теория первого порядка всегда не полна и надо в какой-то момент остановится при её построении. Остановимся сейчас, а что и как расширять (если потребуется) – узнаем уже после данной статьи.

5 Заключительные замечания и благодарности

Невозможно сделать сразу всё от формулировки задач по построению логики вычислимости до полного создания теории. Арифметика Пеано создавалась (выделяя только самый активный период) примерно 30 лет – от 1860-х, когда вопрос о формализации арифметики стал активно обсуждаться, и до публикации Пеано в 1889 г. Логика вычислимости сложнее, и идёт в конфликте с тем, что было создано ранее. Неужели один человек в одной статье в состоянии поставить и решить сразу все возникающие, в связи с этим, вопросы? Нет, конечно, а попытка исполнить подобное с моей стороны было бы сродни замысловатому суициду.

Гильберт ставил такие задачи (формальное обоснование непротиворечивости математических теорий, например), до решения которых (порой отрицательных) проходили десятилетия. И никаких логических выводов, а тем более решений поставленных задач, в подобных публикациях не делал. А я далеко не Гильберт, чтобы пытаться быть большим математиком, чем был он.

Можно было бы рассмотреть, например, в предыдущем разделе аксиомы без типов (оставив только один тип) – так проще выявлять упущения в логике и громоздкости в формулировках. Но подобные детали – для других статей, для работы разных математиков на годы, если не на десятилетия исследований. В этой же статье я обозначил принципиальный кризис в понимании вычислимости в математике, сделал этот кризис осознанным, наметил те фундаментальные свойства (которые смог обнаружить) в логике вычислимости, которые следует – на мой взгляд – включить в создаваемую теорию. И сделал «первую итерацию» в формировании языка теории для работы с числами-строками и строками.

Даже кризис понимания вычислимости ещё не признан математическим сообществом и мой подход не прошёл проверку в принципиальном отношении. И приводить на этом этапе строгие логические выводы теорем на многие десятки страниц из возникшей «ниоткуда» теории – это как искать 5-й знак после запятой в расчётах, где нет ясности даже с тысячами.

К тому же иные способы написания статьи я уже испытывал в других обстоятельствах и вкратце изложу эту историю, чтобы заранее ответить на некоторые советы о том, как «правильно» готовить математическую статью к публикации – которые не учитывают того, что данный случай является особым.

Я не сомневался в том, что арифметика Пеано (и даже ее аналог без аксиомы индукции) имеют необходимые для вычислимости свойства. Мысли проверять арифметику Пеано (и любые ее арифметические расширения) на отсутствие таких свойств – даже не возникало. И доказательства о сводимости вычислений на МТ к частично рекурсивным функциям арифметики представлялись мне корректными с точки зрения сохранения свойств вычислимости при переходе к арифметике.

Но, с другой стороны, я видел, что практики пользуются строками – включая числа-строки (операции с числами на бумаге и в памяти компьютера, например). И логику чисел-строк на практике не получается заменить на арифметику Пеано.

Важным условием для проведения расчетов на практике является приемлемая скорость этих расчетов. И вот на эту тему арифметика заведомо не говорит ничего. Более того, очевидно, что ее разные интерпретации на практике будут соответствовать разной тяжести (по времени и ресурсам

памяти) проведения вычислений.

Таким образом, арифметика недостаточно выразительна для решения вопросов теории алгоритмов, в рамках которых используемые объёмы времени и памяти являются принципиальными характеристиками вычислений. И вот для теории алгоритмов развивать теорию строк было очевидно необходимо.

При этом вопрос о выразительных возможностях арифметики в плане вычислимости – то есть, о принципиальной возможности провести вычисления за конечное время – такой вопрос даже не ставился и выразительность арифметики в этом отношении не подвергалась сомнению с моей стороны в начале исследований.

Таким образом, я пришел к некоторому паллиативу на пути выявления ошибок в понимании и применении арифметики Пеано. Я не оспаривал доказательства теорем о вычислимости частичных рекурсивных функций арифметики, но считал невозможным подобные доказательства в отношении оптимальной (не допускающей хотя бы экспоненциально тяжёлого по объемам используемых времени и памяти варианта вычислений вместо оптимальных в некоторых своих интерпретациях) работы рекурсивных функций.

Да, это был паллиатив, но при скрупулезной обработке результатов – на пути исследования выразительных отличий арифметики Пеано и логики чисел-строк для вычислений – было неизбежно столкновение с принципиальной ограниченностью (недостаточной выразительностью) арифметики Пеано (и любых её арифметических расширений) даже в вопросе вычислимости, а не только в плане решения задач теории алгоритмов. И отсутствие необходимых свойств вычислимости в арифметике Пеано (и в любых её арифметических расширениях) было действительно обнаружено. Но – спустя многие годы после начала исследований.

В соответствии со своим первоначальным планом я собирался сформулировать доказательство недостаточной выразительности арифметики для решения задач теории алгоритмов – пусть не формальное доказательство (ещё без готовой теории первого порядка), а параллельно развивать теорию для строк и чисел-строк – до уровня, когда формализм охватит вопросы времени. Но разработку формализма я не собирался выполнять в одиночку – я собирался лишь начать этот процесс, а математическое сообщество уже вытянуло бы дальнейшее развитие общими усилиями. Ключевым же должно было стать неформальное доказательство недостаточной выразительности арифметики для теории алгоритмов – оно бы и стало обоснованием развития теории для строк и чисел-строк независимо от арифметики Пеано и её арифметических расширений. И такое доказательство удалось найти:

Допустим, мы берём некоторое десятичное число, но записываем его на ленту Машины Тьюринга в его «стандартной интерпретации». Раз годится натуральное число, то не имеет значения, какую правильную интерпретацию для записи этого числа мы используем – в нем есть вся логика натуральных чисел. Желательно брать самую «примитивную» (но полностью соответствующую арифметике) интерпретацию, чтобы избежать посторонних (не арифметических) эффектов.

А теперь представим, что алгоритму надо узнать 1-ю цифру исходного десятичного числа, имея в качестве входных данных это же число, но записанное в стандартной интерпретации (размер

такой записи числа равен самому числу плюс один и состоит из одних и тех же цифр – «счётных палочек»). Если бы число было записано в десятичном виде, то на практике достаточно было бы прочитать первый символ на ленте Машины Тьюринга для решения поставленной задачи. И время на работу алгоритма не будет зависеть от длины «входных данных» на ленте МТ. А вот если пытаться так же «частично» прочитать число в стандартной интерпретации, соответствующее исходному десятичному числу?

Допустим, можно, прочитав N (фиксированное количество) «счётных палочек», выяснить, что соответствующая этому числу строка (десятичное число в нашем случае) начинается на цифру «2» алфавита десятичных цифр. Тогда это будет означать, что все числа от $N - 1$ (такое число в стандартной интерпретации как раз состоит из N «счётных палочек») и до бесконечности начинаются на цифру «2» в своей десятичной записи. А на все остальные строки остаётся только $N - 1$ вариантов (где «счётных палочек» меньше, чем N) разных чисел, включая ноль. Но $N - 1$ вариантов нельзя поставить в соответствие бесконечному количеству вариантов десятичных чисел, начинающихся не на цифру «2» алфавита десятичных цифр. Поэтому в рамках теории алгоритмов не может быть соответствия строк многосимвольного алфавита и натуральных чисел, которое обеспечивало бы схожую скорость вычислений. Ведь в разобранным примере различие времени на вычисления может быть сколь угодно велико между чтением одной первой цифры при десятичной записи числа и чтением всего числа в стандартной интерпретации. Нет вообще никаких ограничений – даже экспоненциальных – между затратами времени на вычисления в этих разных интерпретациях.

В рассмотренном примере есть нюансы – мы полагаем, что на ленте МТ в качестве «входных данных» может быть только конечное количество не пустых «клеток», но эти нюансы легко устранить, либо исходя из того, что бесконечные входные данные невозможны и тогда числу-строке с десятичной записью чисел соответствуют тоже конечные записи чисел в стандартной интерпретации.

Либо мы допускаем бесконечные строки, но в таком случае у нас такие объекты, которые нельзя сопоставить с числами арифметики и арифметика заведомо недостаточно выразительна – в отношении строк многосимвольного алфавита, по крайней мере. В самом деле, Кантор доказал, что действительные числа (от 0 до 1) невозможно «пересчитать» при помощи натуральных чисел – и тут аналогичный случай.

Итак, есть доказательство – довольно красивое, на мой вкус – недостаточной выразительности абстрактной арифметики (арифметики Пеано и любых её арифметических расширений) для решения вопросов теории алгоритмов, но не удаётся пока сформулировать отличия искомой теории чисел-строк и строк от абстрактной арифметики.

Понимая, что едва ли эта задача по силам мне одному (в истории математики на решение подобных проблем уходили иногда века, а в некоторых случаях и тысячелетия), я строю теорию для строк и чисел-строк в той мере, в которой способен понять логику реальных вычислений и формализовать их, добавляю к приведенному неформальному доказательству и в конце апреля 2020 г. обращаюсь (как позже выяснилось) к Сергею Витальевичу Знаменскому с просьбой помочь

с публикацией.

Как теперь понятно, абстрактная арифметика не просто недостаточно выразительна для теории алгоритмов, но с ней дело обстоит гораздо хуже – она в принципе не может быть расширена дополнительными аксиомами так, чтобы числа абстрактной арифметики после её логического расширения обрели необходимые для вычислимости свойства.

Но при подготовке первоначального варианта статьи я ещё не знал ни об аксиомах нарушения равенства чисел-строк из-за строковых свойств, ни о необходимости менять аксиомы арифметики с учётом индексов типов, ни о необходимости замены кванторов в аксиомах арифметики на типированные кванторы – после чего мы получаем другие формулы, с другой логикой, чем в абстрактной арифметике.

В 2020 году я полагал, что потребуется лишь расширить арифметику новыми аксиомами – когда удастся понять логику вычислений в реальности и формализовать «недостающую» в арифметике часть этой логики, придав числам арифметики необходимые новые свойства для вычислимости.

Но интуитивно я выбрал правильный путь – не пытался получать из арифметики аналоги строковых свойств, но аксиоматизировал эти свойства «с нуля». В таком случае нет нужды обосновывать их выводимость из арифметики – а нам ведь нужно рассматривать реальность как безусловный приоритет. И мы рассматриваем именно и только то, что нам надо для исследования реальных вычислений – что менее громоздко, чем при опоре ещё и на арифметику Пеано, поэтому «чище» и легче «вылавливать» проблемы и/или упущения при сравнении с реальностью.

Сейчас формулировка и доказательство теорем в данной статье не является актуальным. Сейчас сформулированы аксиомы с логикой, заведомо отличающейся от абстрактной арифметики, и стоит вопрос об их истинности и признании математическим сообществом, а логические выводы из аксиом никак не могут подтвердить их истинность – ведь из противоречивого утверждения, например, тоже можно доказать нужные теоремы (да и что угодно другое тоже можно доказать). Но если сформулированные аксиомы будут признаны, то можно переходить к второму по приоритету занятию – логическим выводам из найденных аксиом и дальнейшему расширению теории независимыми от найденных аксиом новыми аксиомами – если в подобном расширении возникнет необходимость. Но вернемся к той ситуации, которая была в начале исследования.

Итак, у меня была – с первой стороны – заготовка теории о строках и числах-строках со всякими доказанными теоремами, обоснованными определениями и прочими выразительными инструментами, у меня было – со второй стороны – доказательство недостаточной выразительности арифметики для строк и чисел-строк применительно к вопросам теории алгоритмов. И я был намерен сравнивать первую сторону с реальностью в поисках упущений для того, чтобы сравнить найденные упущения (их логику, записанную в виде некоторых аксиом) с абстрактной арифметикой и обнаружить такие упущения, которые будут соответствовать второй стороне – то есть, окажутся выходящими за рамки арифметики.

Но эту программу действий я планировал для всего математического сообщества после публикации, а не так, чтобы тащить всё самому. Публикации в тот раз не случилось, но программу удалось выполнить – с помощью Сергея Витальевича Знаменского в существенной степени, хотя и

с ним наши пути исследования разошлись в итоге.

Без малого два года переписки, он дал ссылку на статью String theory и многое другое. Был, фактически, моим научным руководителем при обсуждении того, какой должна была бы быть теория чисел-строк и строк для работы с задачами теории алгоритмов и как написать эту теорию («первую сторону» статьи) в соответствии со стандартами математических изданий. Его интуиция не раз помогла сузить область поиска ответа до вполне обозримой.

Кстати, в статье String theory в качестве аксиомы берут некоторую вариацию аксиомы индукции 2-го порядка. На каком основании? Тут видна разница подходов с тем, что использую я. Но результаты доказательств в String theory не вызывали возражений – я же считал арифметику достаточно выразительной для вопросов вычислимости. Возражения вызывал метод аксиоматизации для неисследованных объектов и само доказательство с опорой на эти неубедительные исходные посылки. Впрочем, они оказались ошибочными для чисел и чисел-строк – эти посылки – как теперь видно.

Я очень благодарен С.В. Знаменскому и за то, что важно почувствовать интерес компетентного собеседника к твоим исследованиям, особенно если он тратит на понимание твоих идей и их развитие не шуточные усилия.

Тем не менее рецензент (это не С.В. Знаменский) дал отрицательное заключение в отношении предлагаемой публикации – спустя более, чем год рассмотрения. То есть, он не понял всю принципиальность данного исследования для математики и даже не попытался выяснить у меня то, что я, возможно, сумел бы ему разъяснить.

Я бы сказал, что есть арифметика Пеано – формальная теория, а есть практика и неформальная теория, в рамках которой доказано, что арифметика Пеано не отсекает (в принципе не способна отсечь) даже «неполиномиально тяжёлые» альтернативные способы вычислений. В силу чего логика арифметики Пеано недостаточно выразительна для теории алгоритмов и, с одной стороны, надо искать формализацию теории, которая фактически используется при работе с числами-строками, а – с другой стороны – использовать пока неформальную теорию для работы с числами-строками. То есть – отдавать себе отчёт в том, что «сводимость» к арифметике – это фикция, потому что арифметика Пеано не обладает некоторыми принципиальными свойствами, необходимыми для работы с числами-строками, но формально фиктивность «арифметизации» мы пока обосновать не можем, потому что соответствующая формальная теория ещё не построена.

Примечательно, что математики, фактически, именно так и делают в рамках задач теории алгоритмов – опираясь на неформальную теорию чисел-строк и строк. Есть даже понятие «длинная арифметика», которая вовсе не является арифметикой Пеано. И не формализована, потому что иначе выявилась бы несовместимость абстрактной арифметики с этой теорией – притом на уровне вычислимости (как теперь ясно), что идёт дальше и глубже, чем теория алгоритмов. Но я не видел доказательств (кроме своих собственных неформальных) о недостаточной выразительности арифметики для вопросов теории алгоритмов. Нет даже оговорок в доказательствах о «вычислимости» частично рекурсивных функций арифметики, что эти доказательства не распространяются на вопросы теории алгоритмов.

А в моем доказательстве недостаточной выразительности арифметики для теории алгоритмов есть информация о нюансах, из-за которых арифметика не справляется. И если люди действительно готовы искать вариант построения теории «выйдя за рамки» арифметики – то они должны хвататься за такие «точки входа» в решение проблемы. Отказ публиковать равносильно нежеланию признавать необходимость поиска такой логики теории, которая выходила бы за рамки абстрактной арифметики. Притом отказ при наличии доказательно убедительной необходимости искать логику, которая выходит за рамки арифметики.

Более того, аномальные с точки зрения арифметики факты должны разбираться ещё и с точки зрения проверки – не противоречат ли они арифметике при попытке расширить логику её объектов для придания им свойств вычислимости, что сделало бы необходимым выйти за рамки применения прежних объектов арифметики для придания им свойств вычислимости в расширенной теории. А ведь так и оказалось – хотя в то время я не предполагал такого положения дел.

Но для математиков приоритетом являются признанные ими теории и постановка вопроса о необходимости приоритета реальности (которая обычно даже не входит в круг профессиональных интересов математика) над теорией – с возможностью получить противоречие между реальностью и теорией – идёт вразрез с математическим способом мышления и вызывает, видимо, ужас у математиков.

Поэтому, возможно, я увидел – фигурально выражаясь – спины убегающих математиков, а не их поддержку. Но сказанное не относится к Сергею Витальевичу, который предложил мне сильно сократить статью, чтобы не перегружать рецензента, и после вернуться к попытке публикации.

Но долгое рецензирование тормозило мое продвижение вперед – не менять же существенно статью, пока её проверяют. А после отказа в публикации я не мог не идти дальше, и добавил то, что понял за время ожидания.

К началу января 2021 года я сильно сократил статью и добавил упоминание, что для чисел-строк принцип индукции не верен – потому что он не распространяется на некоторые случаи работы с входными данными, например, при произвольном доступе. Например: чтение первого символа с ленты входных данных, когда ещё нет никакой другой информации помимо этого символа.

То есть, я формулирую, что индукция может быть нарушена. На это следует неожиданно (для меня) сильная реакция со стороны Сергея Витальевича. Он предлагает мне дать пусть неформальное, но убедительное доказательство нарушения индукции для какого-то случая. К этому моменту есть практически всё для такого доказательства, и я его легко строю. Сергей Витальевич пишет, что оно убедительно. Но без Сергея Витальевича или иного человека «в теме» я до этого не додумался бы – чтобы искать и давать доказательство случаев нарушения индукции. А никого другого «в теме» кроме Сергея Витальевича не осталось – остальные убежали.

В результате я понимаю, что речь идет не о расширении свойств чисел арифметики, а о необходимости заменить их объектами другой теории, и в начале февраля 2021 г. предлагаю новый вариант статьи с найденным (с «подачи» С.В. Знаменского) доказательством нарушения индукции и предложением соавторства С.В. Знаменскому. В ходе острой дискуссии наши отношения портятся, и он отказывается считать верным нарушение индукции (тогда ещё не было ни индексов

типа, ни типированных кванторов, ни переделанных аксиом арифметики). Всё же верность математика признанным математическим теориям перевесила и в данном случае. Одно дело ожидать расширения свойств чисел арифметики дополнительной логикой и другое – прийти к выводу о невозможности расширения свойств чисел арифметики Пеано необходимыми для вычислимости свойствами.

Убедительных доводов в защиту своей позиции и в опровержение доказательства нарушения индукции (притом это его доказательство – в большей мере, чем моё) он не привёл, на мой взгляд, но я не собираюсь приводить тут написанные им слова и легко их опровергать – в совместном творчестве до публичных результатов часто идет обмен интуитивными доводами, которые при взгляде на них с точки зрения публичных споров (без понимания взаимного контекста и условностей) будут полной ерундой и такую «ерунду» он мог бы приводить, цитируя и мои слова из наших неформальных личных обсуждений. Но факт в том, что наши пути в середине февраля 2022 года разошлись. Однако я сделал ещё одну попытку убедить Сергея Витальевича.

К началу августа 2022 года я переписал статью с использованием индексов типов и переписал аксиомы арифметики соответствующим образом – в том числе переписав аксиому индукции с использованием типированных кванторов. Это новое построение теории было вовсе не формальной игрой символами, а результатом долгих рассматриваний свойств реальных вычислений и поиска способа изложить логику этих вычислений в математическом формализме. Однако до аксиом нарушения равенств чисел-строк из-за строковых свойств было ещё далеко. И этот вариант статьи я использовал, пытаясь убедить Сергея Витальевича рассмотреть вычислимость в реальном мире и сопоставить её с предлагаемой теорией. Но, похоже, он твёрдо решил не идти по этому пути. Возможно, разочаровался во мне – не знаю.

А уже осенью 2022 года удалось сформулировать аксиомы нарушения равенства из-за строковых свойств. Но это – как и индексы типов строк – я делал уже в одиночестве.

Теперь вопрос, в состоянии ли математическое сообщество в обозримой перспективе признать кризис математики в вопросе понимания вычислимости и заняться построением соответствующей теории. Это ведь будет откат на примерно 150 лет назад и пересмотр того, что было сделано для формализации теории натуральных чисел.

А, возможно, более реалистично подтянуть уровень программистов в вопросе формальной логики и подготовить, таким образом, математиков (часть из таких программистов станут математиками), для которых теория для строк и чисел-строк будет изначально исходной и принятой без мучительного преодоления предыдущей теории – арифметики Пеано и прежних представлений о вычислимости. Программные тексты уже давно не проще сложных логических формул и поэтому фундаментальные теории и язык математики пора перестать рассматривать, как нечто доступное только немногим профессионалам. И нам теперь нужны математики, являющиеся в равной степени и специалистами в ИТ-сфере.

Понимание вычислимости на основе правильной математической теории даёт гораздо более фундаментальное и гибкое образование, чем одно только практическое освоение программирования. И формирует среди наиболее способных к математике обучающихся тех, кто будет и дальше

углублять свое математическое образование и решать в дальнейшем сложные в математическом отношении задачи программирования и даже фундаментальные вопросы теории алгоритмов. Путь освоения математики, включая логику, характерный для классического образования (начиная с гимназий) Российской империи и уничтоженный – не на всегда, надеюсь – коммунистическими временщиками.

К тому же массовость математиков в ИТ-сфере будет ещё и экономической основой жизни резко увеличившегося математического сообщества России, что даст фундамент для всей иерархии обучения, распространения, администрирования, консультирования и развития научных знаний в области точных наук. Нынешняя зависимость от Запада в этом вопросе крайне архаична и может быть теперь относительно легко преодолена. С развитием необходимых инструментов обеспечения интеллектуального труда – авторских прав, патентных отчислений, взаимодействия с бизнесом и т.п. инструментов, которые уничтожались и подавлялись со времен бездарного – в плане развития научного сообщества и знаний – социалистического «эксперимента». Без всех этих инструментов материальной поддержки науки у нас не может быть самостоятельных и способных на систематические собственные масштабные достижения точных наук.

Тем более самостоятельные точные науки необходимы России во время разворачивающейся сейчас войны с Западом. Самостоятельность российских точных наук будет чрезвычайно полезна и для мировой науки – которую Запад в огромной степени уничтожил за время своего доминирования ради торжества собственных научных институтов и монополии на технологии.

К тому же начать выстраивать полноценное математическое сообщество России удобно в ходе реализации наиболее значительного расширения математической теории со времен Евклида в наиболее используемой на практике области математики. Ведь именно с числами-строками и строками на практике работают больше, чем с любыми другими математическими объектами, и именно числа-строки и строки являются объектами той теории, построением которой мы заняты в данной статье и будем заняты (я про математическое сообщество, к которому здесь нескромно причислил и себя), конечно, во многих других статьях после этой.

Разумеется, насчёт «наиболее значительного расширения математической теории со времен Евклида» - можно и поспорить. Самым значительным расширением теории была логика – но это до Евклида, если ориентироваться на Аристотеля (они почти современники). Совершенно гениальным стало применение букв в качестве переменных, функций и (позже) предикатов – но это не из реальности, а обобщение – как и правила оперирования со скобками, композиции, кванторы и т.п. Числа были раньше геометрии Евклида. Сам Евклид внёс свой вклад – «алгоритм Евклида», например. Отрицательные, мнимые числа, кватернионы – создавались внутри математики. Формализация арифметики – это, скорее, систематизация того, что было наработано за тысячи лет, но уже не внесение внутрь математики из реальности. А вот вычислимость – не исследовалась так, чтобы избавиться от ошибочных представлений хотя бы и – как теперь видим – внутри математики про неё представления оказались не вполне корректные. Но я не настаиваю, просто любопытно, что ещё в математику вносили из реальности масштабного, если я что-то упустил.

Я не сторонник того, чтобы «переламывать» нынешних математиков и убеждать их принять

новую теорию вместо прежних (арифметики Пеано, прежних взглядов на вычислимость и т.п.). Ведь математики и без того одной ногой стоят в могиле, чтобы требовать от них ещё большего ума, открытости новому и т.п. Требовать подобное могут только люди, которые никогда не мыслили по-настоящему напряженно, то есть – когда ты ощущаешь, что ещё немного – и ты расстанешься если не с жизнью, то со здоровьем – либо навсегда, либо потребуются многие годы восстановления и ограничения умственной активности. При том, что остановиться в своем сомнении и умственном поиске ты не можешь.

И твоё самочувствие напрямую зависит от твоих умственных представлений и приоритетов, а крушение какой-то важной части из них отняло бы существенную часть твоего здоровья и даже поставило бы под угрозу твою жизнь.

В этом смысле все люди с доминированием точной (в плане требования к итоговому результату) умственной деятельности над остальным в жизни – все они живут в «стеклянных домах». А выясняя истину – мы вынуждены «бросаться камнями» между собой. Поэтому этика тут особенно важна. Ты можешь опровергать идеи визави, но только логично и отстраненно от него самого. Ставить под сомнение направление его исследования или его способности у тебя нет оснований. Визави должен сам делать выводы – если ты опроверг некоторые его идеи – опровергает ли это направление его исследования, или это всего лишь некоторые отрицательные результаты, которые в большом количестве встречаются на пути всякого серьёзного исследования.

И тебе может быть очень тяжело из-за того, что само его исследование является «камнями» в твой «стеклянный дом». И ты можешь очень хотеть, чтобы данный исследователь исчез, можешь желать надавить на него, оскорбляя или принижая его и его исследования. Но подобное поведение является грубым нарушением этики, а в некоторых случаях оно должно рассматриваться как угрожающее чужому здоровью и жизни преступление. Даже если от самого факта продолжения его исследования ты погибнешь. Тогда – умри достойно, а не переходи на личности. Разумеется, если в споре твой оппонент первым грубо перешёл на личности, то я умываю руки со своими призывами к миролюбию.

Многие «стеклянные дома» представлений математиков о числах рассыпятся из-за брошенных мной в этой статье «камнях». И жизнь многих симпатичных (пусть и заочно) мне людей будет испорчена этим. Но, увы – «истина дороже».

Поэтому усугублять эти жертвы – контрпродуктивно, на мой взгляд. Раз предстоит очень серьёзное расширение математического сообщества, то работать надо с новыми математиками в плане новых теорий. А нынешние математики и без этого заняты какими-то своими делами или найдут себе такие, которые не потребуют от них пересматривать свои взгляды. Посмотрите на продолжительность жизни замечательных математиков – типичным является смерть в пределах 55 лет. Пуанкаре, Эмми Нётер, Минковский, Абель, Галуа и т.д. Минковский не дожил и до 45 лет, Абель и Галуа погибли до 30. Если твой мозг загружен и перегружен умственной работой – то он не способен в полной мере координировать работу твоего организма, а порой – и твое социальное поведение в сложных обстоятельствах. Да и сама нагрузка на мозг резко увеличивает риск утраты каких-то его функциональных возможностей из-за «сверхнормативных» режимов его работы.

Немало выдающихся математиков потеряли здоровье и/или математические способности – слепли, теряли силы, заканчивали свои дни в психиатрических клиниках и т.д. А людей – тем более талантливых – надо беречь. Надо создавать благоприятные условия с минимумом дискомфортных и вредных факторов, страховать на случай утраты здоровья и т.д. Что не отменяет необходимости контроля за тем, чтобы к этим благам не «пролезали» бесталанные паразиты.

В нашей стране, например, угробили математические способности Григория Перельмана, а из лично знакомых мне примеров – вынудили уехать Владимира Воеводского (выгнали его из МГУ за академическую неуспеваемость! Но его охотно приняли в аспирантуру Гарвардского университета, никакой диплом не просили и к постоянным нарушениям им формальных регламентов не «цеплялись»), с которым мы учились в старших «математических» классах школы 330 Москвы и приятельствовали в те школьные годы. Тоже умер рано – в 51 год, что я узнал, когда пытался возобновить знакомство ради размещения своей публикации на портале научных препринтов.

И в процессе переписки с С.В. Знаменским я в какой-то момент испугался – когда он несколько дней не отвечал (что контрастировало с предыдущей перепиской). Потому что я отлично понимаю, что, нагружая его своими вопросами и исследованием, я увеличиваю риск для его здоровья и жизни. И мне было страшно, что я стал фатальным дополнительным фактором. К счастью, та тревога оказалась ложной.

Лозунг победивших в СССР бездарей: «талант всегда пробьёт себе дорогу» надо сменить после бездарного развала страны с такими лозунгами на: «бездарь как-нибудь выживет, а вот талант угробить легко, поэтому его надо защищать, хорошо обеспечивать и давать работу».

В самом конце не могу не поблагодарить свою коллегу по бухгалтерскому учёту Светлану Васильевну Литвинову, благодаря которой я несколько лет имел существенную по доходам и не трудную для меня подработку – за счёт чего мог тратить гораздо больше времени на свои математические исследования, чем при обычной нагрузке бухгалтера. Без этой её помощи я бы на несколько лет позже додумался до идей, изложенных в данной статье – или, не исключено, что никогда не додумался бы.

Список литературы

- [1] *John Corcoran; William Frank; Michael Maloney.* String Theory. The Journal of Symbolic Logic, Vol. 39, No. 4, Storrs, USA. 1974
- [2] *Дж. Булос, Р. Джефффри.* Вычислимость и логика. Мир, М. 1994
- [3] *Э. Мендельсон.* Введение в математическую логику. Наука, М. 1984
- [4] *А.С. Герасимов.* Курс математической логики и теории вычислимости. Лань, СПб. 2014
- [5] *А.И. Мальцев.* Алгоритмы и рекурсивные функции. Наука, М. 1986
- [6] *В.И. Игошин.* Теория алгоритмов. Инфра-М, М. 2013