Математический мир

Математическое доказательство: от поколения к поколению

К. Базард, Й. Байер, К. Бенцмюллер, М. Давид, Е. Зельманов, Л. Лэмпорт, Ю. В. Матиясевич, Л. Полсон, Б. Сток, Д. Шляйхер

Доказательство — одно из важнейших понятий в математике. Однако поражает разрыв между теоретическим определением понятия доказательства и тем, как оно практически используется. Это подвергает опасности уникальный статус математики как строгой науки. Может быть, настало время согласовать теорию и практику, т. е. точность и интуицию, благодаря появлению *средств компьютерного доказательства теорем*. Это служило предметом обсуждения в сообществах специалистов. Однако математические доказательства всё более усложнялись, выходя за границы человеческого понимания, так что ведущие математики заинтересовались формальной проверкой своих доказательств. В то же время важнейшие математические результаты в последние годы проверялись на компьютере людьми, не принадлежащими к сообществам специалистов, даже студентами младших курсов. В нашей статье исследуются различные определения доказательства, расхождения между ними и возможности навести мосты. Статья построена как *дискуссия* или *собрание мнений* участников профессиональных сообществ в математике и информатике, находящихся на разных этапах своей деятельности. Авторы бросают вызов распространённым предубеждениям и исследуют новые перспективы.

Jonas Bayer, *Christoph Benzmüller*, *Kevin Buzzard*, *Marco David*, *Leslie Lamport*, *Yuri Matiyasevich*, *Lawrence Paulson*, *Dierk Schleicher*, *Benedikt Stock*, *and Efim Zelmanov*. Mathematical proof between generations // Notices AMS. 2024. Vol. 71, № 1. P. 80–92.

1. Что такое математическое доказательство?

Дирк Шляйхер

Математика часто гордится тем, что она самая фундаментальная из всех наук: математическая истина, установленная единожды, будет верна всегда. Но что именно называется математическим доказательством? Вот один из возможных ответов — в терминах другого фундаментального математического понятия, определения.

Определение (Доказательство: формальное определение). Математическое доказательство — это последовательность рассуждений, которые основаны на заданном множестве аксиом и состоят в формальном выводе следствий согласно формальным правилам дедукции.

Это «идеалистичное» определение доказательства, возможно, появляется первым, и таким формальным способом нетрудно доказывать простые результаты, скажем в комбинаторике, элементарной теории чисел или о евклидовых треугольниках. Но рано или поздно оказывается, что это определение слишком неудобно и непрактично для любого глубокого математического результата. На практике математики пользуются совсем другим определением доказательства.

Определение (Доказательство: практическое определение). Математическое доказательство — это последовательность рассуждений, которая убеждает грамотного читателя.

Тут можно возразить, что математики систематически делают существеннейшую ошибку: определяют нечто одним способом, а потом используют его по-другому. Но разве это не умаляет все важнейшие достоинства математики?

Многие математики поспешат отметить, что типичное доказательство, построенное как убедительная последовательность рассуждений, может быть представлено в более подробном виде: в принципе про любой шаг можно спросить, почему данное утверждение выполнено, и вставить дополнительные шаги для объяснения. В свою очередь каждый из этих шагов можно при необходимости детализировать, пока в итоге не дойдём до аксиом. Можно ожидать, что некоторые этапы такой детализации будут выполнены автором доказательства — вплоть до достаточно фундаментального уровня, когда мы приходим к уже установленным утверждениям. У математиков есть общее понимание, что наиболее актуальные математические факты можно таким

образом обосновать — в принципе. Так что практическое определение доказательства можно истолковать в следующем смысле: доказательство — это последовательность рассуждений, которая убеждает грамотного читателя, что формальное доказательство в смысле первого определения «может быть построено».

Но, правда, случается, что математические «доказательства», написанные людьми, содержат ошибки. Многие из них тривиальны, и тогда можно вставить недостающие шаги или рассмотреть пропущенный частный случай аналогично остальным. Часто утверждают, что рано или поздно любые ошибки обнаружатся благодаря тщательности математического сообщества.

Но время от времени появляются теоремы, которые считаются доказанными, а много позже доказательство оказывается ложным или неполным, и даже сам результат может оказаться ложным. Как узнать про данную теорему, что «рано или поздно» не случится завтра, когда, может быть, откроется фундаментальный пробел? Как при этом поверить в математическую корректность доказательства?

Классический, достаточно известный пример — теорема о четырёх красках: эта теорема считалась доказанной уже в конце XIX века, когда было дано не одно, а два доказательства — и через одиннадцать лет в обоих нашлись ошибки.

Недавний пример из теории динамических систем. Доказательства, которые позже оказываются ошибочными, — не просто теоретическая возможность или случай из ранней истории математики (когда стандарты математической строгости могли быть не столь высоки). Покажем это на весьма недавнем примере из теории динамических систем. Утверждение формулируется совсем просто:

Пусть $f\colon \mathbb{C} \to \mathbb{C}$ *— голоморфная функция, но не многочлен* (*т. е. трансцендентная целая функция*). *Тогда* f *имеет не больше одной максимальной вполне инвариантной области*: *это такая область* U, *что* $f(U) = U = f^{-1}(U)$, *причём* $\mathbb{C} \setminus U$ *содержит не менее двух точек*; *максимальность означает, что* U *не содержится в строго большей области* U' *с теми же свойствами*.

До самого недавнего времени этот результат рассматривался как теорема, доказанная в 1970 г. Ноэлем Бейкером, видным исследователем в этой области [1]. Однако спустя примерно полвека Дюваль заметил, что доказательство содержит ошибку, а Ремпе и Сикссмит [19] показали, что её нельзя исправить применёнными методами. В частности, ключевой шаг в доказательстве состоит в том, что если $U, V \subseteq \mathbb{C}$ — непересе-

кающиеся односвязные области, то одна из областей $f^{-1}(U)$ и $f^{-1}(V)$ должна быть несвязна. Это утверждение, однако, ложно; как показали Рempe и Сикссмит, контрпримером служит даже столь простая функция, как $f(z) = e^z + z$: для неё существует бесконечно много непересекающихся односвязных областей со связными прообразами. Поэтому доказательство Бейкера нельзя исправить в рамках первоначального подхода. Основное утверждение остаётся открытым вопросом и сегодня — спустя полвека после того, как его сочли строго доказанной теоремой.

Можно спросить, насколько этот результат актуален — не осталась ли ошибка незамеченной потому, что он никого не интересовал? К сожалению, это не так. Рempe и Сикссмит [19, § 9] дают впечатляющий список результатов, которые так или иначе зависят от этой ошибочной статьи. В списке есть несколько публикаций разных авторов, которые опираются либо на ошибочную статью, либо на другие статьи, где использован ошибочный результат, но в которых можно доказать основные результаты другими методами, изложенными в статье [19]. Другой список содержит несколько работ, в том числе 35-летней давности, которые опираются на ошибочную статью и основные результаты которых снова надо считать открытыми вопросами. Ещё в одном списке — два обзора, давности между 25 и 30 годами, где упоминается результат из статьи [1]. И, что особенно обескураживает, в нескольких публикациях, иногда причём классических и много цитируемых, ошибочный метод доказательства был признан столь полезным, что он был модифицирован, развит далее и обобщён (а ошибка осознана не была). В результате целая область математики была вынуждена выяснять, насколько повлияла на неё столь «убедительная» ошибка, сделанная на полвека раньше.

Это совсем недавний пример, который сильно взволновал математическое сообщество и показал, что в любой момент могут обнаружиться проблемы даже в общеизвестных, казалось бы, результатах. Однако вполне возможно, что сейчас математика движется к тому, что два определения «доказательства» можно будет наконец согласовать. Как к этому можно прийти — одна из ключевых тем этой статьи.

Другая тенденция в современной математике, приводящая к сходным выводам, состоит в том, что доказательства становятся длиннее и сложнее, иногда настолько, что они уже не могут быть ни целиком загружены в человеческую память, ни проверены рецензентами. Например, Петер Шольце недавно попросил формально проверить один из его ключевых результатов по «сжатой математике», а именно проект под названием «Эксперимент по жидким тензорам» («Liquid Tensor

Experiment»), который привлёк широкое внимание и был успешно завершён в 2022 г. За много лет до этого Владимир Воеводский точно так же беспокоился о корректности результатов в его области, относящихся к разработке «теории гомотопических типов».

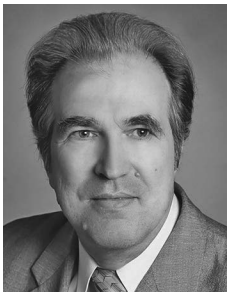
Одним из побудительных мотивов для нашей статьи послужил семинар на «Гейдельбергском форуме лауреатов» в 2018 г. под названием «Будущее математического доказательства», где присутствовал ряд соавторов этой статьи. Некоторые её темы обсуждались и разрабатывались на этом семинаре.

Статью надо воспринимать как калейдоскоп многих вопросов, относящихся к математическим доказательствам и соответствующим человеко-машинным системам. Её создали несколько исследователей разного профиля. В § 2 *Юрий Матиясевич* излагает свою точку зрения, что вскоре потребуется сопровождать математические публикации формальными доказательствами и это вполне может быть сделано уже следующим поколением. В § 3 *Ефим Зельманов* защищает важность смысла и объяснения в математике по сравнению с формальной проверкой. В § 4 *Лесли Лэмпорт* излагает способ представления математических доказательств, который «должен сильно затруднить публикацию ошибочных доказательств». В § 5 *Кристоф Бенцмюллер* обсуждает несколько крупных достижений в компьютерном доказательстве теорем и формулирует представление о том, как соединять формальные и традиционные доказательства. Затем в § 6 *Йонас Байер*, *Марко Давид* и *Бенедикт Сток*, в то время студенты-старшекурсники, описывают, как они обучались технике компьютерного доказательства с самого начала и осуществили одну из первых крупных формализаций, выполненных молодым поколением. В § 7 *Кевин Базард* объясняет, как он делает компьютерные доказательства «притягательными» для математиков, и рассказывает, как он включил их в учебные занятия. В § 8 *Лоренс Полсон* намечает открытые проблемы и перспективы на будущее.

Наши соавторы выражают в статье различные точки зрения. Так что при чтении вы, естественно, можете не согласиться с какими-то их утверждениями. На самом деле это относится также к отношениям соавторов с рецензентами — чтение их переписки побудило одного из нас заметить: «Это всё ещё математика или уже бытовая мыльная опера?». Надеемся, что калейдоскоп, представленный в этом тексте, вдохновит читателей и поможет им выработать свою собственную точку зрения.

Дирк Шляйхер, профессор математики университета Экс-Марсель
dierk.schleicher@univ-amu.fr

2. Зачем формализовать математические результаты? И почему десятая проблема Гильберта?



Юрий Матиясевич

Десятилетие назад Санкт-Петербургское математическое общество провело заседание, которое называлось «Математическое доказательство: вчера, сегодня, завтра». Как один из трёх докладчиков, я совершенно не согласился с предыдущим и осмелился [15] публично объявить следующее:

> ПРЕДСКАЗАНИЕ
> Через 25 лет математические журналы (если им суждено дожить до этого времени) не будут принимать к рассмотрению статьи, не сопровождаемые доказательствами, которые может проверить компьютер.

Уже в то время существовал по крайней мере один (и только один?) журнал, где доказательства проходили предварительную проверку на компьютере. Это был журнал «Formalized mathematics», основанный в 1990 г. За прошедшее десятилетие количество математических журналов продолжало расти, но я не знаю ни одного нового журнала, который бы требовал формализованные доказательства. Тем не менее я осмеливаюсь повторить своё предсказание verbatim et litteratim [дословно и буквально], т. е.: «Через 25 лет *с нынешнего момента*…».

В самом деле, прогресс в компьютерной проверке доказательств очень впечатляет — и в развитии программного обеспечения, и в количестве и значимости реально проверенных теорем. Сводку таких результатов можно найти в интернете [23]; список доказательств регулярно обновляется, но изначально ограничен выбором теорем из «Топ-100».

Но в чём могла бы состоять цель компьютерной проверки результатов, уже доказанных людьми? Один очевидный ответ таков: дополнительно удостовериться в корректности доказательства (есть много примеров важных и широко применяемых теорем с ошибками в доказательстве, которые десятилетиями оставались незамеченными). Однако некоторые люди считают, что сами компьютеры недостаточно надёжны из-за возможных ошибок в их конструкции/программном

обеспечении/функционировании. По моему мнению, «через 25 лет» прогресс в этой области полностью устранит такие возражения.

Формализованные доказательства жизненно важны для весьма масштабного проекта под названием «Всемирная библиотека цифровой математики» («World digital mathematics library» [11]). Конечная цель этого проекта — перенести всё *математическое знание* (не только аксиомы/определения/теоремы/доказательства) на компьютеры. Это потребует колоссальных усилий, но кто же захочет тратить своё драгоценное время на кропотливое оформление уже известных результатов? Математики предпочитают создавать новые идеи (определения, гипотезы, теоремы) и не ценят тяжёлую работу по написанию доказательств со всеми мелкими подробностями (к счастью, такое занятие ценят специалисты по информатике).

Давно предложено следующее средство решения этой проблемы: «ушедшие от дел» математики старшего поколения, которые уже не способны генерировать новые блестящие математические идеи, могли бы посвятить остаток жизни «обучению компьютеров математике». Но есть и противоположный вариант: это могла бы делать молодёжь, лишь начинающая осваивать математику. При этом они смогут остро почувствовать, что такое математическая строгость. Однако остаётся невыясненным следующее: как повлияло бы участие в такой деятельности на способность создавать новое в математике?

Так что, услышав, что группа студентов из университета Якобса в Бремене изучила под руководством профессора Дирка Шляйхера доказательство неразрешимости десятой проблемы Гильберта, я предложил им продемонстрировать понимание всей конструкции, создав полностью формализованное доказательство. Моя роль в проекте была весьма ограниченной: я снабдил студентов достаточно подробным (для человека) доказательством и отвечал за выбор системы Isabelle для верификации. Я был счастлив увидеть огромный энтузиазм, появившийся у студентов, — может быть, потому, что они (как и я) в тот момент недооценили объём необходимой работы.

Десятая проблема Гильберта не очень трудна для формализации. Немного странно, что вначале мы ждали её полвека, а потом за короткое время возникли четыре независимых верификации в системах Coq, Isabelle/HOL, Lean и Mizar [14, 16, 17, 21].

Юрий Владимирович Матиясевич, академик, советник РАН
yumat@pdmi.ras.ru

3. О доказательстве и прогрессе математики: взгляд математика-исследователя

Ефим Зельманов

Я добавлю свои 5 центов к замечательной дискуссии о компьютерной проверке доказательств, организованной студентами. Больше сорока лет я живу в мире доказательств, иногда сложных.

Если мне скажут, что доказательство корректно, поскольку так говорит компьютерная программа, но я не увижу больших идей, «вращающих колёса», то я, вероятно, продолжу думать над задачей, как будто не было компьютерного благословения.

Цель доказательства — *понимание*. Для математиков недостаточно знать, верно или нет то или другое утверждение. Они хотят знать, *почему* оно верно или неверно. Часто такое понимание приходит в виде связи с большими идеями, которые вступают в игру там и сям в разных контекстах. Не смогу выразить это лучше, чем У. Тёрстон в его прекрасной статье о доказательствах (см. [22]).

По моему мнению, надёжное компьютерное тестирование доказательств — замечательное достижение в области искусственного интеллекта. Оно ценно и интересно само по себе, не говоря о доказательствах как таковых. Оно может найти и другие применения.

Должен признать, что там, где идёт речь непосредственно о вычислениях, я доверяю компьютерам больше, чем людям.

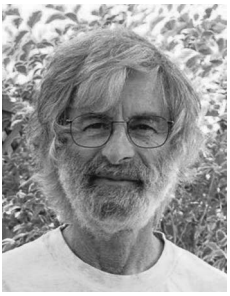
Трудные и важные доказательства часто излагаются на пределе интеллектуальных возможностей человека (вспомним о доказательстве гипотезы Пуанкаре, принадлежащем Г. Перельману). Следует ли ожидать, что авторы представят их в виде, удобном для компьютера? Опасаюсь, что мы хотели бы слишком многого.

Современная математика породила новую проблему: доказательства невероятной сложности. Я знаю доказательства некоторых глубоких результатов, которые появились довольно давно, но никто (кроме авторов) всё ещё не может сказать, что понимает все подробности. Единственная надежда на то, что хорошие доказательства подобны живым организмам. Они развиваются с течением времени и проходят естественный отбор. Появляются новые идеи и приносят новое понимание. И могут появиться новые доказательства.

Закончу неочевидным утверждением, что на практике доказательство — это такое рассуждение, которое все математики считают доказательством.

Ефим Зельманов, профессор и заведующий кафедрой,
Южный научно-технологический университет (Шэньчжэнь)
efim.zelmanov@gmail.com

4. Сделать математику более строгой



Лесли Лэмпорт

Математическая строгость в практике большинства математиков не очень высока. Есть свидетельства, что примерно треть всех опубликованных, прорецензированных математических статей содержит существенные ошибки — неправильные доказательства или теоремы, в правильность которых авторы верили. (Такое свидетельство я привёл в другом месте [12].) Можно сделать математику более строгой и математики будут делать меньше ошибок, если заменить архаичные привычки более разумными, и вот каким образом.

Формулы. Несколько веков назад формулы записывались прозой. Сегодня математики понимают преимущества записи формул в математических обозначениях: они короче, понятнее и удобнее в применении. Замена прозы математикой должна была уменьшить количество ошибок.

Математики думают, что они перестали писать формулы прозой. Они ошибаются. Они заменили лишь часть прозы в своих формулах математикой. Рассмотрим следующее определение того факта, что $\lim_{x\to a} f(x) = b$.

> Для любого $\varepsilon > 0$ существует такое $\delta > 0$, что при всех y,
> если $0 < |y - x| < \delta$, то $|b - f(y)| < \varepsilon$. (1)

Математик найдёт определение (1) совершенно нормальным, даже несмотря на то, что эта запись математической формулы содержит много слов. Вот как эта формула записывается без слов:

$$\forall\, \varepsilon > 0\colon\ \exists\, \delta > 0\colon\ \forall\, y\colon\ (0 < |y - x| < \delta) \quad \Rightarrow \quad (|b - f(y)| < \varepsilon. \quad (2)$$

Полагаю, что большинство математиков сочтут формулу (2) более непонятной и некрасивой, чем (1). Думаю, что несколько столетий

назад математики сочли бы непонятной и некрасивой формулу

$$0 < |y - x| < \delta.$$

Зачем писать формулу (2) вместо (1)? По той же причине мы не пишем «0 меньше, чем абсолютная величина…»: формула (2) короче, понятнее (когда вы освоились с обозначениями) и удобнее в применении. И она уменьшит количество ошибок. Покажите студентам, изучающим основы анализа, определение (1) и попросите их записать, что утверждение о равенстве $\lim_{x \to a} f(x)$ и b ложно. Сомневаюсь, что многие из них запишут правильно. Немного научите их элементарной логике, и они легко смогут вычислить отрицание формулы (2). Самое очевидное использование слов в формулах — выражать логические операции; но они используются и иначе, например для описания множеств и функций.

Формулы, записанные без слов, могут теперь обрабатываться компьютерными программами. Программа легко вычислит отрицание формулы (2). Но отрицание формулы (1) она вычислить не сможет[1)]. Такие программы помогут студентам освоиться с математическими понятиями, если эти понятия описаны математически, а не прозой.

Математики считают, что трудно записывать формулы полностью в математических обозначениях, без слов. Я спрашивал многих математиков, насколько длинным было бы вполне строгое, не содержащее слов определение риманова интеграла — если считать известными определения множества действительных чисел, арифметических операций над ними, а также основы теории множеств. Я получил ответы в пределах от 50 строк до 50 страниц. Они ошибочны.

Я разработал язык под названием TLA$^+$, на котором инженеры составляют полностью формальные математические описания компьютерных систем. В нём есть средства для проверки корректности используемой математики. Интеграл Римана определяется на языке TLA$^+$ примерно в дюжине строк.

Доказательства. Несколько веков назад доказательства писались прозой. И до сих пор пишутся. Математики даже не начали менять способ записи доказательств. Они думают, что их доказательства выражают строго логическое мышление. Они ошибаются. Их доказательства в прозе написаны в литературном стиле, который затемняет

[1)] Для записи формул вроде (1) предложены неестественные языки с ограничениями, которые могут пониматься компьютерной программой. Такие языки малополезны или совсем бесполезны для тех, кто не боится математики.

логику доказательства. Рассмотрим следующую фразу, с которой начинается доказательство в учебнике по основам анализа Майкла Спивака [20, p. 170] — а эта книга считается очень строго написанной.

Пусть a и b — две точки промежутка, причём $a < b$.

Это явно неверная формулировка, так как рассматриваемый промежуток может состоять из одной точки, и тогда невозможно выбрать a и b. На самом деле эта фраза — часть корректного доказательства, но читатель должен открыть для себя это доказательство, скрытое в прозе Спивака.

Запись доказательств прозой ведёт к ошибкам. Как их избежать? Большинство специалистов по математике и информатике считают, что единственный способ для этого — писать машинно-проверяемые доказательства. Это требует записи формул на формальном языке. Язык TLA^+ достаточно прост, чтобы его использовали инженеры, не продвинутые в математике, и достаточно близок к работающей математике, чтобы математики находили его вполне естественным. Но он слишком прост, чтобы годиться для записи таких доказательств, какие можно встретить в большинстве статей из математических журналов. Формализация таких доказательств требует языка, слишком сложного для большинства инженеров — и, как я полагаю, большинство математиков найдёт его совершенно непонятным. Лишь немногие математики приложат усилия, чтобы выучить такой язык, если только он не сделает запись их доказательств существенно легче. Сегодня он сильно затрудняет запись большинства доказательств. Стандартные машинно-проверяемые доказательства сейчас практичны лишь в некоторых ситуациях, включая некоторые критичные для безопасности. Не ожидаю, что это изменится в ближайшие несколько десятилетий.

К счастью, теперь есть простой метод, позволяющий любому писать доказательства с меньшим количеством ошибок. Он не может устранить все ошибки, но сильно уменьшает шансы на их появление. Его основная идея в том, чтобы линейный порядок обычной прозы заменить иерархической структурой и именовать гипотезы и доказанные факты так, чтобы далее в доказательстве можно было на них ссылаться. Ниже кратко объясняется этот метод; полное описание появилось в другом месте [13].

Теорема состоит из утверждения и его доказательства. Доказательство — либо короткий абзац, либо последовательность утверждений и их доказательств. В каждом пункте доказательства есть текущая цель и набор фактов, которые могут использоваться для достижения

⟨2⟩3. $A \wedge B \Rightarrow C$
Текущая цель: доказать, что $A \wedge B \Rightarrow C$
⟨3⟩1. ДОСТАТОЧНО ПРИНЯТЬ A, B
ДОКАЖИТЕ C
Доказательство: простая логика.
Тривиальное доказательство того факта, что, приняв A, B, а затем доказав C, мы достигаем текущей цели.
Текущая цель: C; утверждения A и B добавляются к используемым фактам.
⟨3⟩2. D
Доказательство утверждения D.
D добавляется к используемым фактам.
⟨3⟩3. E
Доказательство утверждения E.
E добавляется к используемым фактам.
⟨3⟩4. Что и требовалось доказать.
Доказательство утверждения C.
Текущая задача и используемые факты те же, что перед ⟨2⟩3, с добавлением $A \wedge B \Rightarrow C$.
⟨2⟩4. …

Рис. 1. Утверждение и его структурированное доказательство

этой цели. Эти утверждения могут быть записаны прозой или математически. В последнем случае логическая структура утверждения часто задаёт иерархическую декомпозицию его доказательства. На рис. 1 показана структура части доказательства, содержащего утверждение $A \wedge B \Rightarrow C$, где для доказательства утверждения C вначале доказываются D и E. Обычно из этих двух утверждений легко следует C, что делает лёгким доказательство завершающего шага ⟨3⟩4. Номер ⟨2⟩3 показывает, что данное утверждение — третье в доказательстве второго уровня для утверждения первого уровня.

Это доказательство прямолинейно, и из его представления на рис. 1 не видна необходимость структурирования. Но предположим, что это малая часть длинного доказательства, а доказательство каждого из утверждений D и E занимает полстраницы. Если доказательства написаны прозой, то как сможет читатель отследить, где закончилось действие допущений A и B и где нельзя больше применять утверждение D? Математики пытаются справиться со сложностью дока-

зательств, используя леммы; но так добавляется лишь один уровень иерархии, а это продвигает не очень далеко.

Сделать доказательство более строгим — значит закрыть все дыры, где могут скрываться ошибки. Это означает сделать его длиннее. Если удлиняется доказательство, написанное прозой, то станет труднее его читать. Но при иерархической структуре удлинение доказательства облегчает чтение. Дополнительные пояснения возникают на нижних уровнях иерархии, так что они не затемняют структуру доказательства. Это особенно верно, когда математики перестают создавать печатную продукцию на спиленных деревьях и начинают использовать гипертекст, так что нижние уровни доказательства можно скрыть, когда их не читают. Чтобы избежать ошибок, нужны более подробные доказательства, чем можно сейчас найти в журналах. Значит, пока журналы не используют гипертекст, нужно писать подробное доказательство, чтобы отловить ошибки, а затем сокращать его для публикации. Это легко делать со структурированными доказательствами: вы просто заменяете нижние уровни иерархии короткими набросками доказательств. (Можно даже написать латеховский макрос, чтобы единственный файл мог породить обе версии путём изменения немногих символов.)

Можно научить студентов писать структурированные доказательства на примере очень простых машинно-проверяемых доказательств определённой тематики. Любая хорошая система машинного доказательства должна допускать иерархическое структурирование доказательств. Язык для записи теорем должен быть простым — не похожим на сложный язык, нужный в серьёзной математике. Язык TLA^+ и его система доказательства не идеальны, но их можно использовать при неимении лучшего.

Студентам нужно понимать, что факты, изучаемые на занятиях по математике, могут в принципе быть формально доказаны на основе простых аксиом и правил вывода. На практике мы лишь приводим доказательства к такому уровню, что читатель, как мы верим, сочтёт все шаги очевидно верными. Этот уровень растёт с образованием и опытом. Иногда мы также для краткости записываем формулы словами. Но студенты и математики должны осознавать, что они могут сделать свои доказательства вполне строгими и приблизить их к исходным аксиомам насколько захотят.

Иерархически структурированная запись доказательств помогает избегать ошибок; но она не гарантирует, что их не будет. Нужно честно сказать себе, какие утверждения очевидны и какие нужно

доказать. Мой совет: подробно распишите доказательство до уровня, на котором, как вы думаете, всё очевидно, а затем спуститесь на один уровень глубже. Но если не заботиться о корректности доказательств, ничто не удержит от ошибок, кроме необходимости написать машинно-проверяемое доказательство.

Что следует сделать теперь? Если вы согласны, что глупо записывать формулы словами, а доказательства — прозой, немедленно перестаньте это делать. Не надо ждать, пока другие изменятся.

Формулы. Не требуется удалять все слова из ваших формул. Начните с использования кванторов $\forall$ и $\exists$. Затем попробуйте избавиться от «…», что не является математическим оператором. Последовательность $x_1, \ldots, x_n$ — не что иное как функция x с областью определения[2] $1..n$, отображающая каждое i из этой области в x_i. Часто, хотя не всегда, математические выражения становятся проще и изящнее, если устранить «…» и ввести функцию x — попробуйте. Опасайтесь подменять строгую запись словами и неаккуратными обозначениями. Если вы открыты для перемен, вы найдёте, что математическая строгость — часто самый простой подход. Если вы учитель, ваши ученики должны получить — или получать — простейшие сведения, необходимые для записи формул с меньшим количеством слов, чем сейчас. Помогите ученикам освоиться с правильными математическими обозначениями, используя такие обозначения на занятиях.

Доказательства. Нет причин не начать записывать структурированные доказательства уже сейчас. Нужна лишь одна-две фразы, чтобы объяснить читателям, как их надо читать. Я делаю так уже примерно 30 лет, и ни один редактор или рецензент не пожаловался на мои доказательства. Для начала прочитайте, как я записываю структурированные доказательства, но не стесняйтесь менять мой стиль так, как вам подходит. Лишь две черты необходимо сохранять: иерархическую структуру и возможность давать названия и ссылаться на предположения и уже доказанные утверждения.

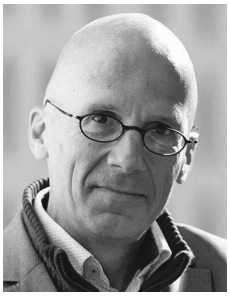
Если вы преподаватель, научите своих студентов структурировать доказательства так, как это делаете вы. Они ещё с этим не определились, и они оценят, насколько структурирование облегчает понимание ваших доказательств. Побуждайте их писать структурированные доказательства во всех их курсах. Другие профессора вряд ли станут

[2] Выражение «…» обозначает математический оператор со следующим определением: $i \ldots j := \{k \in \mathbb{Z} : i \leqslant k \leqslant j\}$, где $\mathbb{Z}$ — множество всех целых чисел.

жаловаться на слишком строгие доказательства; возможно, это даже побудит их самих так писать.

Лесли Лэмпорт, специалист по информатике, Microsoft Research

5. Что такое доказательство? Чем оно должно быть?



Кристоф Бенцмюллер

Относится ли понятие математического доказательства к строгому, но обычно не слишком наглядному *формальному выведению новых «истин» из их предпосылок посредством аккуратно определённых правил вывода*? Или это *искусный акт коммуникации*, при котором красивые структуры, порождающие новые математические идеи, открываются коллегам таким образом, что их можно легко *увидеть* и воспринять, и даже получить новое вдохновение?

Прежнее понятие *формального доказательства* связано прежде всего с логической строгостью и надёжностью. Наглядность и красота второстепенны, если вообще имеют значение. В последнее время формальные доказательства привлекают возрастающее, хотя весьма неоднозначное, внимание математиков, которое вызывается, например, успешным применением современных технологий доказательства теорем к трудным задачам, требующим математических рассуждений и их проверки. С помощью человеко-машинных систем были решены трудные задачи, с которыми человек иначе бы не справился. Приведём некоторые примеры.

(i) *Проблема четырёх красок*: эту знаменитую задачу решили Аппель и Хакен с помощью технологии автоматического доказательства уже в 1977 г., а позже Гонтье создал интерактивное формальное доказательство с помощью компьютерной системы Coq.

(ii) *Гипотеза Кеплера* (о наилучшей упаковке сфер в трёхмерном евклидовом пространстве): команда экспертов приложила огромные усилия для проверки решения, представленного Хейлсом в «Annals of Mathematics», но в итоге не смогла полностью это выполнить. В конце концов Хейлс и его команда осуществили это с помощью интерактивной системы HOL Light. Главный результат: теперь доступно формальное доказательство, которое можно проверять независимо — людьми и компьютерными программами.

(iii) *Проблема пифагоровых троек* (можно ли покрасить все натуральные числа в два цвета так, чтобы не было монохромных пифагоровых троек): эту трудную задачу решили Хойле, Кульман и Марек [9], используя технологию автоматического SAT-доказательства (от boolean SATisfiability — булева выполнимость). Формальное доказательство, сгенерированное компьютерной программой, имеет колоссальный размер (около 200 терабайт); тем не менее его ещё можно независимо проверить (по крайней мере на компьютерах). Продолжением этой линии исследований стало автоматическое решение проблемы о пятом числе Шура[3)] [8] и доказательство гипотезы Келлера[4)] [5].

(iv) Моя собственная текущая работа с коллегами сосредоточена на технологиях металогических рассуждений высших порядков [3], позволивших нам обнаружить и объяснить ошибки и проблемы в отрецензированных публикациях по математике, методологии вычислений и этике информатики. Среди них — обнаружение незамеченного противоречия в гёделевском современном варианте онтологического доказательства существования Бога, обнаружение и объяснение глубоко скрытого парадокса в статье Э. Н. Залта «Principia Logico-Metaphysica» и выявление некоторых небольших проблем в известном учебнике по теории категорий [4].

Здесь можно упомянуть и многие другие работы, например машинную проверку доказательства теоремы о нечётном порядке[5)].

Следует заметить, что формальная реконструкция работы математика, вообще говоря, требует много времени и сил, например из-за недостатка больших и удобных в использовании библиотек формализованной математики, а также нехватки программного обеспечения, которое не создавало бы по ходу дела дополнительных трудностей. С другой стороны, математические результаты технического характера, вроде упомянутых в п. (iii), не всегда опираются на наглядные и содержательные математические доказательства. В целом математика сталкивается с задачами возрастающей сложности, которые для своего решения и последующей оценки (например, рецензентами) требу-

[3)] Для какого наибольшего натурального числа n можно покрасить натуральные числа вплоть до n в пять цветов так, что уравнение $a + b = c$ не будет иметь монохромного решения? — *Прим. перев.*

[4)] В любом замощении n-мерного евклидова пространства одинаковыми гиперкубами найдутся два гиперкуба, соприкасающиеся грань к грани. — *Прим. перев.*

[5)] Теорема Фейта — Томпсона: любая конечная группа нечётного порядка разрешима. — *Прим. перев.*

ют методов, выходящих за традиционные рамки. Вышеприведённые примеры (i)-(iv) — лишь первые свидетельства этого. Когда доступны технологии, позволяющие обнаруживать ошибки в публикациях, их несомненно следует использовать для повышения научного качества последних. Поэтому формальные доказательства должны занять более важное место в математике и вокруг неё.

На самом деле я считаю долгом перед обществом принять этот вызов. Просто цепляться за традиционное понятие математического доказательства — непригодное решение в мире возрастающей технологичности. Подумаем о таких областях, как «проверка программ в информатике» или «надёжный ИИ», где мы в идеале хотим формальных гарантий, что реализованные сложные решения математически корректны, но где традиционные наглядные доказательства могут отсутствовать.

Тем не менее сами по себе формальные доказательства представляют ограниченный интерес и в идеале всегда должны будут соединяться с традиционными. Объяснимость, прозрачность и интуитивная ясность должны остаться достоинствами с высшим приоритетом, и не только в математике. В дальней перспективе возрастающая надёжность и красота комбинированного подхода, где оба вида доказательств соединены на равных, оправдают дополнительные ресурсы, которые надо вложить сейчас. Ошибки при публикации (даже мелкие) будут предотвращаться формальными доказательствами, а появлению непродуманных и малопонятных теорий будут препятствовать требования интуитивной ясности и красоты.

Итак, каким должно быть доказательство? Вывод таков, что доказательство в идеале должно быть и человеко-ориентированным традиционным, и машинно-ориентированным формальным. Традиционные математические доказательства создаются и используются людьми, а формальные преимущественно создаются и используются машинами. Однако при соединении этих антиподов они постепенно вовлекаются в гармоничный танец.

На следующем шаге, после *соединения*, можно мечтать о *слиянии* традиционных и формальных доказательств в нечто единое. Современные человеко-машинные системы, такие как Isabelle/HOL, обеспечивают возрастающую наглядность языков для построения и изложения доказательств, но достижение вышеназванной честолюбивой цели ещё требует значительного научного прогресса. Отыскание всеобъемлющего решения, которое включает оба понятия доказательства, можно даже рассматривать как вызов для ИИ, поскольку требует-

ся «бесшовное» семантическое соединение естественного языка, диаграмм, языка формул и т. д. вплоть до обмена содержательными рассуждениями между людьми и системами машинного доказательства.

Недавняя работа, которую выполнили Марко Давид, Бенедикт Сток, Йонас Байер и их товарищи, даёт основания для надежды. Их проект верификации решения десятой проблемы Гильберта значительно меньше по охвату, чем некоторые проекты, упомянутые выше, но отличается тем, что студенты-математики приступили к своему проекту формализации, поддержанному Матиясевичем, без предварительного знания технологии машинного доказательства. Тем не менее они приняли вызов и достигли большого прогресса, независимо работая с системой машинного доказательства. Это особенно примечательно, так как это новый пример (в дополнение, скажем, к [6]) впечатляющего проекта формализации, выполненного людьми не из профессионального сообщества формализованной математики. И это ещё одна великолепная демонстрация зрелости, которой достигла современная технология машинного доказательства.

Кристоф Бенцмюллер, заведующий кафедрой технологии систем ИИ
и профессор в Отто-Фридрих-университете (Бамберг),
профессор факультета математики и информатики
в Свободном университете Берлина
c.benzmueller@gmail.com

6. Машинное доказательство: правильный способ изучения математики?!

Йонас Байер

Марко Давид

Бенедикт Сток

Когда мы впервые встретились с Ю. В. Матиясевичем, мы не подозревали, что он создал «грандиозный план», чтобы утвердить своё видение будущего математических доказательств. Он посетил наш

университет ещё в 2017 г. и изложил идею о формализации ДПРМ-теоремы[6]. Это ключевой результат в его доказательстве неразрешимости теории диофантовых уравнений, что даёт отрицательное решение десятой проблемы Гильберта.

Не подозревали мы и о том, что играем роль «подопытных кроликов», которые должны в тщательно организованном эксперименте реализовать идею, что компьютерная проверка математических доказательств посильна даже для молодых и неопытных университетских студентов. Теперь мы глубоко благодарны за роль, которая нам была предоставлена. Ниже раскрывается, чему мы научились с тех пор; теперь в другом проекте мы сами руководим новым коллективом студентов, никогда раньше не занимавшихся компьютерной проверкой доказательств.

Матиясевич быстро обрисовал ситуацию; он изложил нам свою теорему и пригласил работать над её компьютерной проверкой. Ни он, ни мы не имели понятия, каких усилий это потребует. В то время нам казалось, что нужно, поняв доказательство ДПРМ-теоремы, просто «перевести» его на такой язык, чтобы система Isabelle могла его проверить. Но каждый, кто когда-либо работал с интерактивной системой машинного доказательства, знает, что слово «перевести» не очень оправданно по отношению к этому процессу. На самом деле он включает заполнение возможных дыр, которые часто обнаруживаются в математических статьях. Общие фразы типа «легко видеть, что…» при этом должны получить логический смысл. Вскоре мы осознали, что трудности часто кроются в леммах, выглядящих вполне невинно. В итоге нам пришлось пять раз пересматривать формализацию простого понятия регистровой машины, прежде чем наше формальное определение оказалось полезным.

Несмотря на встретившиеся трудности, мы настаиваем, что обучение работе с интерактивной системой машинного доказательства — вполне осуществимое, хотя ещё не лёгкое дело. В предыдущей статье [2] мы размышляли над своими ошибками и встретившимися трудностями. Ключевой вывод состоит в том, что продвижение почти невозможно, если рядом нет специалиста, отвечающего на вопросы. Поэтому, чтобы популяризовать систему машинного доказательства среди следующего поколения, мы теперь сами активно обучаем формализации математики новую группу студентов. Текущая

[6] ДПРМ-теорема утверждает, что любое рекурсивно перечислимое множество диофантово. Названа в честь своих авторов: Мартин Дэвис, Хилари Патнем, Джулия Робинсон и Юрий Матиясевич.

необходимость лично передавать опыт отличает системы машинного доказательства от большинства языков программирования и систем компьютерной алгебры вроде Mathematica. При этом онлайн-форумы вопросов и ответов, например[7] Stack Overflow, дают свободный доступ к поисковым базам данных почти по всем мыслимым вопросам об этих средствах, тем самым обеспечивая их доступность для широкой аудитории. Интерактивные системы машинного доказательства и связанные с ними сообщества, ныне замкнутые, должны последовать этому примеру, чтобы стать полезными типичному математику!

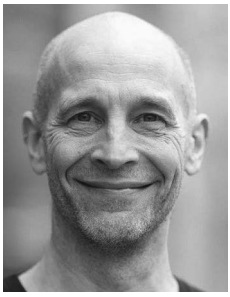
Помимо технических навыков и результатов, этот проект породил изменение парадигмы, в которой мы рассматриваем «обычную» математику. В наших университетских курсах мы больше не спрашиваем, убедительно ли доказательство для нас, а интересуемся, можно ли его формализовать в системе Isabelle. Поскольку компьютер часто выявляет допущения или особые случаи, которые человек пропускает, такое направление мысли порождает более строгий подход к (неформальным) рассуждениям. Наши новые, более точные представления в большой мере выковывались во взаимодействии с компьютером и его специфической логикой. Стало видно, как следующее поколение математиков сможет примирить два конфликтующих определения понятия «доказательство».

Йонас Байер, аспирант Кембриджского университета
jcb234@cam.ac.uk

Марко Давид, аспирант Калифорнийского университета (Беркли)
marco.david@berkeley.edu

Бенедикт Сток, аспирант Оксфордского унивиерситета
stock@maths.ox.ac.uk

7. Некоторые мысли о формализации математики



Кевин Базард

Сделать формализацию привлекательной.

В математике существует мода. Пожалуй, в течение нескольких последних десятилетий в моде была программа Ленглендса. Его гипотезы доказаны, выдвинуты новые гипотезы, построена классическая теория, затем теория для модуля p (важнейшая при

[7] https://stackoverflow.com.

доказательстве последней теоремы Ферма), а теперь — ошеломляющая *p*-адическая теория, а также геометрические и теоретико-категорные версии. Могу признаться без смущения, что моя работа с Йоханом Коммелином и Патриком Массотом, когда мы перевели строку «Пусть X — перфектоидное пространство» на язык системы Lean, была попыткой *рекламировать* системы машинного доказательства перед математиками. Мы хотели показать им, что такие системы уже готовы работать с актуальной современной математикой. Нет, мы ещё не можем предъявить ничего столь впечатляющего, как непостижимое для человека доказательство гипотезы Римана в миллиард строк (на самом деле компьютеры, которые генерируют доказательства трудных гипотез, находящихся на острие современной математики, остаются научной фантастикой на сегодня и вполне могут остаться таковой ещё на десятки лет). Однако давайте подумаем об этом.

Если мы смиримся со статус-кво (когда по существу никто не загружает привлекательные математические факты в системы машинного доказательства), то эти системы *никогда* не смогут такие факты доказывать, поскольку они просто не будут о них знать! Компьютеры заведомо не могут прочесть тот бред, который мы пишем в наших статьях (да часто и люди не могут его прочесть). Люди должны делать ручной перевод — и чем раньше, тем лучше. Так чья же *обязанность* загружать *утверждения* глобальных гипотез Ленглендса в эти системы? Кто сделает это, дав человечеству шанс создать компьютерные ресурсы обучения и доказательства теорем, учебные данные по большим языковым моделям для студентов, которые учатся и работают по программе Ленглендса? Несомненно, это наша обязанность как математиков. Никто больше не собирается этого делать — нельзя ожидать, что специалисты по информатике превратятся в специалистов по программе Ленглендса; гораздо легче математикам выучить какой-то язык программирования. Если работающие математики не станут выполнять перевод, давайте предоставим его аспирантам-математикам. Если вы недавно защитили диссертацию по чистой математике, сможете ли вы — более того, сможет ли человечество — *сформулировать* в системе Lean основные теоремы, доказанные вами в этой диссертации? При нынешних технологиях практический ответ всё ещё сильно зависит от тематики. В отношении моих магистрантов и аспирантов я убеждён на 100 %, что мы сможем не только сформулировать, но и доказать основные результаты в системе Lean.

Заглянем в будущее: а что если математики массово займутся формализацией и мы вдруг найдём перфектоидные пространства и гипо-

тезы Ленглендса в системе Lean, а также в системах Arend (для гомотопической теории типов), Isabelle/HOL (для теории простых типов), MetaMath (для теории множеств), Coq и HOL 4, HOL Light и Mizar, cubical Agda и… Что тогда? Я уже высказал мнение, что было бы научной фантастикой ожидать, что эти системы начнут *доказывать* программу Ленглендса.

Но близко к реальности следующее. Системы вроде Lean можно применять как дополнительный интерактивный ресурс для аспирантов, изучающих алгебраическую геометрию или иную густонаселённую область. Как только создана база данных из *формулировок* множества теорем, фигурирующих в проектах Stacks Project или Kerodon (базах данных по алгебраической геометрии и иной привлекательной математике), можно передать управление специалистам по информатике. У них есть средства, известные как молотки (hammers), позволяющие строить доказательства или контрпримеры к загруженным в систему утверждениям с помощью базы данных, которую создали мы, математики. Отметим, что такие средства *не требуют формализации доказательств*, так что даже огромную базу данных вроде формулировок теорем из Stacks Project можно построить вручную за несколько человеколет — в идеале руками молодых алгебраических геометров, которых интересуют новые методы изучения этой области. Разумеется, формализация доказательств — увлекательное дело, так что они, без сомнения, займутся этим. Другая возможность — обучать коду Lean большие языковые модели вроде ChatGPT. Это направление находится ещё во младенчестве, но может оказаться решающим.

Сделать формализацию интересной. Приведённые выше примеры показывают, что математическое сообщество вполне может выиграть от формализации серьёзной математики в системе машинного доказательства. Опыт показывает, что формализацией занимаются в основном молодые люди. Поэтому важно учить их формализации; однако это другое дело, чем учить их математике. Игра в натуральные числа (the Natural Numbers Game) — это игра в системе Lean с использованием браузера, которая возникла в Империал-колледже из многих часов написания случайных задач в системе Lean для студентов-математиков и дальнейшего наблюдения за тем, как студенты их решают. Идея выводить факты о натуральных числах из исходных принципов была очень удачной, и она возникла отсюда. Студенты говорят: «Я закончил игру в натуральные числа, что дальше?» Правильный ответ: «Инсталлируйте Lean согласно инструкциям на сайте сообщества».

Но когда они выполнили и это, что дальше? Это зависит от их математических интересов и способностей. Часто я побуждаю увлечённого студента формализовать математические факты, уже хорошо знакомые ему, в качестве первого проекта в системе Lean. Правило таково: если это компилируется, ты выиграл. После этого можно начать разговор о том, как написать код, отвечающий высоким стандартам.

Для аспирантов и более опытных в математике людей, которые интересуются происходящим в этой области, всегда полезно указать текущие математические проекты, реализуемые в системе Zulip на `leanprover.zulipchat.com` — платформе с доступом из чата сообщества Lean. Мы всё время ищем новых людей для помощи в разных проектах, и мы рады «взять на борт» новичков. Вот некоторые из главных проектов, активных на сайте в настоящий момент: формализация доказательства великой теоремы Ферма для регулярных простых чисел (руководитель Риккардо Браска) и формализация некоторых основных результатов «сжатой математики» (руководитель Адам Топаз). В 2024 г. начинается проект под моим руководством при поддержке Исследовательского совета по инженерным и физическим наукам (Великобритания) по формализации полного доказательства Последней теоремы Ферма. Ранее на сайте разрабатывались проекты по выворачиванию сферы (руководитель Патрик Массот), в котором формализовалось современное доказательство возможности вывернуть сферу в трёхмерном пространстве, и «эксперимент с жидким тензором» (руководитель Йохан Коммелин), где формализуется доказательство теоремы Клаузена и Шольце 2019 г. В 2020 г. Шольце предложил сообществу формализаторов проверить ключевую теорему, которую он анонсировал совместно с Клаузеном, и сообщество Lean приняло вызов. Интересна предыстория этого проекта; Шольце считал, что при существующей в математике процедуре рецензент не углубляется в подробности работы, и Шольце хотел увидеть, сможет ли это сделать система машинного доказательства. Оказывается, это действительно возможно. Шольце был консультантом при реализации проекта, занявшей 18 месяцев; попутно сообщество построило ряд других формализованных теорий (например, теорию гомологической алгебры), и некоторые из них теперь используются в других проектах.

Обучение навыкам формализации. Научить студентов формализовать математику — значит научить их переводить математические идеи с естественного языка на язык Lean и обратно. Как и при изучении естественного иностранного языка, вы начинаете переводить

текст с одного языка на другой и спрашиваете, если что-то непонятно. В курсе языка Lean, который я веду на математическом факультете Имперіал-колледжа, мы работаем с математикой, уже изученной студентами на естественном языке, и учимся разговаривать на языке Lean. В дальнейшем, когда студенты уже отчасти овладели этим языком, я ввожу новые математические факты. Афина Тома и Паола Янноне научили меня, что параллельное изучение в первые годы отношений эквивалентности и языка Lean обычно кончается плохо. Однако для студента, который знает то и другое, формализация биекции между отношениями эквивалентности и разбиениями — превосходный опыт освоения языка Lean. Требуется искусство для выбора нужного проекта, и какой проект подходит — обычно зависит от студента.

Поправки в текущих исследованиях. В интервью для журнала Wired мне однажды приписали утверждение, что вся математика ложна. Я знаю, что это неверно — часть математики определённо верна. Эти суждения (по крайней мере в классической логике) противоположны друг другу. Однако я действительно сказал, что некоторые наши башни, возможно, построены на песке, а теперь я думаю, что даже это слегка наивно. После бесед с Давидом Рабуэном, историком и философом математики, я теперь понимаю, что передний край математики всегда выглядит примерно так: есть подробности, ещё не проверенные до конца, но все знают, что обычно получится хорошо, по крайней мере настолько, чтобы основная теорема выполнялась.

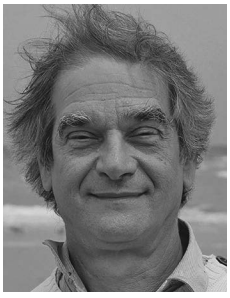
Однако ошибки в математике иногда случаются. В этом столетии лишь в моей области (теория чисел) авторитетные математики анонсировали доказательства гипотезы Леопольдта и ABC-гипотезы; последняя работа была даже опубликована в солидном математическом журнале. Однако не видно, чтобы наше сообщество признало эти доказательства строгими. К сожалению, Lean не избавит нас от этих проблем, по крайней мере сейчас. Пока что Lean доказывает теоремы путём их перевода людьми с естественного языка, и если никто не готов взяться за грандиозную работу по переводу доказательства ABC-гипотезы, которое предложил Мотидзуки (а почему кто-то возьмётся? Математическое сообщество в прошлом не так обращалось с опубликованными доказательствами), то я не вижу, как выйти из тупика.

Некоторые доказательства уже вышли за пределы, доступные единичному мозгу, — ни один человек не понимает полностью все их идеи. Области, считавшиеся модными, могут отмереть, если будут доказаны великие гипотезы, которые их движут. Всё, что не документи-

ровано должным образом, в действительности рискует быть потеряно. Можно надеяться на появление новых, более простых доказательств, но история показывает, что это не всегда оправдывается: иногда сложные вещи так и остаются сложными. Независимо от того, обращаемся ли мы к системам машинного доказательства, математикам необходимо более внимательно, чем раньше, следить за тщательным документированием того, что мы считаем уже известным, — чтобы иметь ответ на технические вопросы от будущих поколений математиков.

Кевин Базард, профессор чистой математики
в Империал-колледже (Лондон)
k.buzzard@imperial.ac.uk

8. Когда компьютерное доказательство станет повседневным в математике?



Лоренс Полсон

Введение и предварительные сведения. Идея применять технические средства в математических рассуждениях начала осуществляться в 1960-х годах. Система Н. Г. де Брёйна AUTOMATH — это теория типов для выражения математических определений и доказательств. А. В. Трыбулец включил в систему Mizar удобочитаемый формальный язык для абстрактной математики. Оба автора были профессиональными математиками и хотели, чтобы их работа была полезна в самой математике. Но технология не была готова, тем не менее они достигли существенного прогресса. Разработка системы AUTOMATH привела к сегодняшним теориям зависимых типов, к исчислению индуктивных конструкций. В системе Mizar собрана содержательная и обширная библиотека формализованной математики, а её удобный структуризованный язык — всё ещё лучший из существующих.

Интерактивные системы машинного доказательства возникли в середине 1970-х годов. Первой была, возможно, Mizar, но наибольшее влияние бесспорно оказала эдинбургская LCE. В ней реализована логика для вычислимых функций (a Logic for Computable Functions), которая быстро устарела, но также введена архитектура, которую восприняли наиболее успешные системы, в том числе Coq, HOL, Isabelle и Nuprl [7]. Эти средства предназначались для верификации в информатике. Система HOL (от high-order logic — логика высших порядков)

была выбрана, чтобы реализовать определённые методы проверки оборудования. Эти задачи проверки редко требовали какой-то математики кроме целочисленной.

В 1994 г. была обнаружена ошибка в процессоре с плавающей точкой на пентиуме (фирма Интел), и это драматическое событие привлекло внимание специалистов по верификации к действительным числам. Джон Харрисон формально доказал корректность алгоритма для вычисления экспоненты, учтя все особенности арифметики с плавающей точкой. В дальнейшем Харрисон сыграл главную роль в проекте *Flyspeck* по формальной проверке доказательства гипотезы Кеплера, принадлежащего Томасу Хейлсу. Он формализовал многие основополагающие математические результаты, например теорему о распределении простых чисел.

Гонтье, формализовав доказательство теоремы о четырёх красках, уже продемонстрировал, что интерактивная система машинного доказательства (в данном случае Coq) может помочь в решении значимого математического вопроса. Эта работа была аналогична проекту Flyspeck в том отношении, что включала формальную проверку большого количества вычислений.

Система Isabelle. Система Isabelle [18] возникла на основе LCF для поддержки множества формализмов, в том числе теории множеств. Однако в сфере верификации господствует логика высших порядков, так что господствующей версией этой системы стала Isabelle/HOL. В ней расширен формализм различных логик высших порядков с классами аксиоматических типов, что позволяет многократно использовать формальный материал на основе единого аксиоматического базиса [10]. В систему Isabelle включён язык структурированных доказательств Isar, основанный на математическом языке системы Mizar. На языке Isar можно выразить вложенные доказательства, в которых явно описаны их этапы — промежуточные утверждения и их доказательства. В системе Isabelle осуществляется автоматизация доказательств (чтобы вызывать мощные внешние системы доказательства, применяется *sledgehammer* — «кузнечный молот») и опровержений — в виде *поиска контрпримеров*. Пользовательский интерфейс — уникальная интегрированная среда для текущего редактирования доказательств.

Между этими компонентами существует мощное взаимодействие. Текст структурированного доказательства легко использовать повторно. Скопированный в новый контекст, он тут же будет проверен, а любые ошибки отмечены. К структурированным доказательствам удобно

применять sledgehammer: если данное утверждение слишком трудно для автоматического доказательства, пользователь может предложить достаточно лёгкое для этого промежуточное утверждение, в итоге приводящее к доказательству первоначального утверждения.

Эти мощные средства сильно уменьшили трудности, первоначально связанные с формальными доказательствами. Они также стимулировали рост значительных библиотек формализованной математики.

Системы машинного доказательства: час настал? Сегодня у нас широкий набор систем: семейство HOL для логики высших порядков; Isabelle/HOL для логики высших порядков с классами аксиоматических типов; Coq и Lean, реализующие исчисление конструкций. Такое разнообразие вызвано конфликтом приоритетов — например, простотой реализации и её семантики в противовес выразительной силе логического исчисления. Соревнование исследовательских групп стимулировалось онлайн-листом, в котором Фреек Видийк фиксировал, какие теоремы из «топ-100» [23] доказаны в той или иной системе. Только одна из ста теорем не доказана ни в одной, и это Последняя теорема Ферма!

Существующими средствами машинной проверки охватывается многое в современной математике. Ныне существуют огромные библиотеки формализованной математики. «Архив формальных доказательств» системы Isabelle[8)] содержит свыше 650 ссылок на множество базовых математических фактов — из линейной алгебры, многомерного анализа, теории вероятностей, комплексного анализа, топологии — и свыше 3 миллионов строк доказательств. Библиотека *mathlib* в системе Lean — огромный и быстрорастущий массив материала из всех областей математики[9)].

Готовы ли в итоге эти системы оказывать помощь математикам? Всё ещё существует много препятствий.

- Формальный синтаксис выглядит искусственным и часто едва понятен. Чтобы увидеть трудности, связанные с традиционными обозначениями, сравним x^2, $\nabla^2 f$, $\sin^2\theta$, $f^2(x)$. В теории групп G обозначает группу, но также и множество; ab — произведение элементов a и b, но HK — совсем другая вещь, просто потому, что мы использовали другую часть алфавита. В теории множеств $\lambda < \aleph_1$ имеет иной смысл, чем $\lambda < \omega_1$, даже притом что $\aleph_1 = \omega_1$.

[8)] `http://www.isa-afp.org`.

[9)] `https://leanprover-community.github.io/mathlib-overview.html`.

- В библиотеках формализованной математики ещё много пробелов, а то, что имеется, бывает трудно найти. Названия теорем часто неоднозначны (что такое теорема Рота?), и это верно, например, и для понятия предела (которое может относиться к анализу, топологии или даже теории множеств). Но нужна возможность искать «предельные теоремы» и найти что-нибудь подходящее. Ещё лучше, чтобы система могла сама что-то предлагать.
- Часто бывает слишком трудно доказать очевидные утверждения. Например, показать, что множество конечно, что функция непрерывна, вычислить производную или оценить предел. В некоторых случаях бывает эффективным квалифицированное применение существующих средств автоматизации. В других случаях нужно писать специализированные решающие процедуры. Наиболее известны решающие процедуры для линейной арифметики; недавнее достижение здесь — система Мануэля Эберля для отыскания пределов.

Навстречу будущему. В последнее время при поддержке Европейского исследовательского совета[10)] мои коллеги и я занимались исследованием и расширением границ сегодняшней технологии. Мы формализовали сравнительно недавние (после 1970 г.) и глубокие результаты[11)]. Последние включают аддитивную комбинаторику, теорию экстремальных графов и теорию комбинаторных схем. Мы даже формализовали некоторые сложные определения, в частности определение схемы Гротендика (необходимое в исследованиях на переднем крае алгебраической геометрии). До сих пор считалось, что эти определения не охватываются теорией простых типов системы Isabelle/HOL.

Математикам также нужна помощь в навигации по нашей огромной библиотеке формализованной математики. Бывает трудно выяснить, формализован ли нужный результат: многие теоремы существуют под несколькими названиями и, наоборот, одно и то же название (например, неравенство Юнга) может применяться к семейству различных результатов. С каждым известным результатом могут быть связаны дюжины технических лемм, по большей части очевидных и однако необходимых, чтобы что-нибудь доказать в сегодняшних системах. Мои коллеги создали экспериментальную поисковую машину SErAPIS, сделавшую возможным поиск по всей библиотеке с помощью огромного словаря математических понятий[12)].

[10)] Проект ALEXANDRIA, GA 742178.

[11)] https://www.cl.cam.ac.uk/~lp15/Grants/Alexandria/.

[12)] https://behemoth.cl.cam.ac.uk/search/.

Привлекательна и идея, что можно использовать доказательства из нашей библиотеки для автоматического порождения новых доказательств. Это ещё один способ помочь извлечь пользу из наших 3 миллионов строк формальных доказательств. Многообещающие результаты как раз начинают появляться в нескольких исследовательских группах.

В более отдалённом будущем можно надеяться автоматизировать математическую интуицию. Она сообщает нам, что данная функция заведомо непрерывна или что некоторая формула не может порождать только простые числа. Таким путём мы узнаём, что некоторое утверждение не может быть дословно верным или допускать доказательство объявленными методами. Формализация демонстрирует снова и снова, что, хотя опубликованные доказательства часто содержат ошибки, сами теоремы обычно верны. Математики могут усмотреть истину, не всегда умея записать корректное рассуждение. Придать такую интуицию компьютеру — значит преобразовать нашу сферу деятельности. Эта проблема останется открытой для одного-двух следующих поколений.

Лоренс Полсон, профессор вычислительной логики
в Кембриджском университете
lp15@cam.ac.uk

Благодарности

Фотографию Кристофера Бенцмюллера предоставил Доминик Шрайнер, Pressestelle Erzbistum Bamberg.

Фотографию Кевина Базарда предоставил Томас Аргус, Имперал-колледж, Лондон.

Фотографии всех других авторов предоставлены авторами.

Список литературы

[1] *Baker I. N.* Completely invariant domains of entire functions // Mathematical Essays Dedicated to A. J. Macintyre. 1970. P. 33–35.

[2] *Bayer J.*, *David M.*, *Pal A.*, *Stock B.* Beginners' quest to formalize mathematics: A feasibility study in Isabelle // Intelligent Computer Mathematics. C. Kaliszyk, E. Brady, A. Kohlhase, and C. Sacerdoti Coen, editors. Cham: Springer International Publishing, 2019. P. 16–27.

[3] *Benzmüller C.* Universal (meta-)logical reasoning: Recent successes // Science of Computer Programming: methods of software design: techniques and applications. March 2019. Vol. 172, № 1. P. 48–62.

[4] *Benzmüller C.*, *Scott D. S.* Automating free logic in HOL, with an experimental application in category theory // J. Automat. Reason. 2020. Vol. 64, № 1. P. 53–72.

[5] *Brakensiek J.*, *Heule M.*, *Mackey J.*, *Narvez D.* The resolution of Keller's conjecture // J. Automat. Reason. 2022. Vol. 66, № 3. P. 277–300.

[6] *Buzzard K.*, *Commelin J.*, *Massot P.* Formalising perfectoid spaces // CPP 2020. J. Blanchette and C. Hritcu, editors. ACM, 2020. P. 299–312.

[7] *Harrison J.*, *Urban J.*, *Wiedijk F.* History of interactive theorem proving // Computational logic. Amsterdam: Elsevier/North-Holland, 2014. (Handb. Hist. Log.; Vol. 9). P. 135–214.

[8] *Heule M. J. H.* Schur number five // Proc. 32nd AAAI Conf. AAAI Press, 2018. P. 6598–6606.

[9] *Heule M. J. H.*, *Kullmann O.*, *Marek V. W.* Solving and verifying the Boolean Pythagorean triples problem via cube-and-conquer‘ // Theory and applications of satisfiability testing — SAT 2016. Cham: Springer, 2016. (Lecture Notes in Comput. Sci.; Vol. 9710) P. 228–245.

[10] *Hölzl J.*, *Immler F.*, *Huffman B.* Type classes and filters for mathematical analysis in Isabelle/HOL // Interactive theorem proving. Heidelberg: Springer, 2013. (Lecture Notes in Comput. Sci.; Vol. 7998). P. 279–294.

[11] IMU Committee on Electronic Information and Communication (CEIC), World digital mathematics library (WDML). https://www.mathunion.org/ceic/library/world-digital-mathematics-library-wdml.

[12] *Lamport L.* Errors in proofs — a correction and further data. https://lamport.azurewebsites.net/tla/proof-statistics.html.

[13] *Lamport L.* How to write a 21st century proof // J. Fixed Point Theory Appl. 2012. Vol. 11, № 1. P. 43–63.

[14] *Larchey-Wendling D.*, *Forster Y.* Hilbert's Tenth Problem in Coq // 4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019). H. Geuvers, editor. Vol. 131 of Leibniz International Proceedings in Informatics (LIPIcs). P. 27:1–27:20. Dagstuhl, Germany, 2019. Schloss Dagstuhl — Leibniz — Zentrum fuer Informatik.

[15] *Матиясевич Ю. В.* Математическое доказательство: вчера, сегодня, завтра // Компьютерные инструменты в образовании. 2012. Вып. 6. С. 13–24. http://ipo.spb.ru/journal/index.php?article/1532/.

[16] *Pak K.* Diophantine sets. Preliminaries // Formalized Mathematics. 2018. Vol. 26, № 1. P. 81–90. http://dx.doi.org/10.2478/forma-2018-0007, DOI 10.2478/forma-2018–0007.

[17] *Pak K.* The Matiyasevich Theorem. Preliminaries // Formalized Mathematics 2018. Vol. 25, № 4. P. 315–322.

[18] *Paulson L. C.* Isabelle: A generic theorem prover. Berlin: Springer-Verlag, 1994. (Lecture Notes in Comput. Sci.; Vol. 828). With contributions by T. Nipkow.

[19] *Rempe-Gillen L.*, *Sixsmith D.* On connected preimages of simply-connected domains under entire functions // Geom. Funct. Anal. 2019. Vol. 29, № 5. P. 1579–1615. DOI 10.1007/s00039-019-00488-2. MR4025520.

[20] *Spivak M.* Calculus. New York: W. A. Benjamin, Inc., 1967.

[21] *Aryal D.*, *Bayer J.*, *Ciurezu B.*, *David M.*, *Deng Y.*, *Devkota P.*, *Dubischar S.*, *Hassler M. S.*, *Liu Y.*, *Oprea M. A.*, *Pal A.*, *Stock B.* Hilbert Meets Isabelle: Formalisation of the DPRM Theorem in Isabelle // EasyChair, 2018.

[22] *Thurston W. P.* On proof and progress in mathematics // Bull. Amer. Math. Soc. (N. S.). 1994. Vol. 30, № 2. P. 161–177.

[23] *Wiedijk F.* Formalizing 100 theorems. `http://www.cs.ru.nl/~freek/100/`.