



ФОРМАЛИЗОВАННАЯ ДИСКРЕТНОСТЬ В МАТЕМАТИКЕ КАК ОСНОВА ОБУЧЕНИЯ ИНФОРМАТИКЕ

Аннотация

Затрагиваются революционные тенденции XX века в формализации математики. Уточняется влияние информатики на преподавание некоторых разделов формализованной математики студентам-информатикам. Предлагаются желательные изменения читаемых математических курсов для студентов-информатиков. Характеризуются специальность и направление «Математическое обеспечение и администрирование информационных систем», разработанные и реализованные, прежде всего, преподавателями математико-механического факультета Санкт-Петербургского государственного университета. Перечисляются учебные пособия, излагающие опыт автора в преподавании некоторых разделов формализованной математики для студентов-информатиков.

Ключевые слова: формализация, дискретное моделирование, непрерывное моделирование, алгоритм, сложность алгоритма, FP-SPACE, математическое обеспечение, информатика.

1. РЕВОЛЮЦИОННЫЕ ТЕНДЕНЦИИ XX ВЕКА В ФОРМАЛИЗАЦИИ МАТЕМАТИКИ

XX век разработал формализацию математики. Г. Фреге изобрёл понятие квантора. Дж. Пеано предложил обозначения для них. Как следствие, те теоремы, которые раньше казались экзотическими, сейчас привлекают больше математического внимания. Например, первая теорема Гёделя (о неполноте формальной арифметики), которая состоит в том, что не существует полной системы для доказательства утверждений, использующих, кроме логических связей и кванторов, только полиномы над натуральными числами. Вторая теорема Гёделя (о неполноте формальной арифметики) говорит о том, что если непротиворечивость формальной ариф-

метики можно доказать средствами самой формальной арифметики, то формальная арифметика противоречива. Из других интересных математических теорем можно вспомнить теорему о независимости континуум-гипотезы от аксиом теории множеств Цермело-Френкеля, теорему Гёделя о полноте исчисления предикатов, предложенного Д. Гильбертом. После математической формализации понятия алгоритма стало ясно, что исчисление предикатов слишком богато, и удалось доказать, что нет и не может быть алгоритма проверки выводимости формул в исчислении предикатов. Не может быть аналогичного алгоритма проверки выводимости формул для любой математической теории, которая формализована на основе исчисления предикатов и содержит формальную арифметику. Всё это было установлено до 60-х годов XX века.

Следующим этапом (60–70-е годы) широко исследовался вопрос о существовании алгоритмов для решения различных задач. Например, Ю.В. Матиясевич доказал невозможность существования алгоритма проверки наличия целых неотрицательных корней у полиномов с целыми коэффициентами, решив тем самым Десятую проблему Гильберта.

Начиная с 70-х годов XX века выяснилось, что этого недостаточно. Выяснилось, что для решения некоторых дискретных задач есть решающие их алгоритмы, но они затрачивают слишком большое число шагов, чтобы с их помощью можно было проводить вычисления на современных компьютерах или компьютерах ближайших поколений. Возникла такая область исследований, как теория сложности алгоритмов, в которой нас интересует не просто существование алгоритма, но существование алгоритма с заданной верхней границей его сложности (например, с числом выполненных шагов, ограниченным сверху полиномом от длины записи исходных данных). Впрочем, удобнее и проще доказывать, что объём памяти, используемой в процессе работы алгоритма, не превосходит полинома заданной степени от длины записи исходных данных.

Проблема в том, что не все алгоритмы, которые принципиально решают ту или иную задачу, могут быть полиномиально масштабированы относительно времени и, тем более, относительно памяти. Язык формальной дискретно-конечной математики более удобен для описаний, которые может преобразовывать компьютер в тех или иных (даже достаточно больших) объёмах исходных данных. Например, целочисленные арифметические операции сложения, вычитания и умножения выполняются в IBM-совместимых компьютерах по модулю 2^{16} для целых чисел из отрезка $[-2^{15}, 2^{15} - 1]$. Вместо рациональных чисел полезно использовать двоично-рациональные числа, то есть имеющие знаменатель, представляющий собой неотрицательную степень двойки. Такие числа являются математическим аналогом компьютерного понятия двоичного числа с фиксированной точкой.

Как говорил Гаусс, «Математика – это язык». А раз это язык, то какие средства используются в языке формальной дискретно-конечной математики? Если мы планируем использовать компьютер, то желателен язык программирования, который тесно связан со сферой искусственного интеллекта, которая постоянно развивается. Языки рефал и пролог из некоторых относительно последних версий полезны и удобны при реализации алгоритмов решения задач искусственного интеллекта.

На мой взгляд, в дальнейшем многие математические вопросы, многие вопросы физических и инженерных приложений будут решаться с применением пакетов компьютерной алгебры. Но при этом людей ждут некоторые сложности. Например, возникает вопрос: как дифференцировать условные выражения? В точке ветвления нужно дополнительно обращать внимание на гладкость и другие особенности поведения функции. Но пакеты компьютерной алгебры это иногда не учитывают и в результате разные пакеты по-разному дифференцируют одну и ту же кусочно заданную функцию. Это один из примеров того, с чем приходится сталкиваться человеку, который вроде бы решил задачу с помощью компьютера, но выясняется, что математический язык человека и язык программного обеспечения компьютера не являются согласованными. Поэтому на современном этапе использования компьютерной техники язык непрерывной математики следует преобразовать. Дискретность приобретает в математике большую важность, и, как следствие, должно быть больше исследований в этом направлении. Алгоритмов в непрерывной математике практически нет, поскольку традиционные точные математические понятия алгоритма использует только дискретные данные.

К тому же бывают разные взгляды на ценность теорем в различных математических школах. Есть существенная разница в том, в каких странах каким разделам математики уделяют больше внимания. Есть много разных других сложностей. Приведу пример из математической логики. Конечнoзначная логика лучше соответствует описаниям

в условиях неопределённости, когда наша точка зрения очень часто меняется. Поэтому «чёрно/белую» логику (да/нет), конечно, должна сменить конечнозначная логика. Например, возможные варианты ответа в социологическом опросе: да, нет, не знаю. Здесь полезно применение трёхзначной логики.

Пока эти тенденции ещё не полностью захватили многие приложения, поэтому мне кажется, что в XXI веке сделают много открытий и найдут оттенки, о которых мы сейчас и не догадываемся. Например, как должна быть написана программа для мобильного телефона, чтобы свести к минимуму потребление энергии? Это пример практической задачи, для которой теоретические вопросы раньше не разрабатывались. Есть много моментов, связанных со сложностью алгоритмов, и я думаю, что у нас появится ещё много неожиданного.

Последняя треть XX века – победное развитие математических конструкций, используемых в информатике.

Я наблюдал три революции использования вычислительной техники. Первая связана с появлением компьютеров, когда маститые математики говорили: «Что программист может сделать? Только запрограммировать алгоритм, который я для него придумал». Потом математики всё-таки осознали, что работа с компьютером требует значительных интеллектуальных усилий. Затем появились персональные компьютеры. После этого уже маститые программисты говорили: «Я уже реализовал этот алгоритм. А что принципиально нового сделал программист на персональном компьютере?». Дело в том, что многие разработчики программного обеспечения для персональных компьютеров не могли толково объяснить маститым программистам, работающим на больших мощных компьютерах, как и какие трудности они преодолели при решении поставленной задачи. Третья революция – появление встроенных приложений мобильных устройств. Выяснилось, что их функционирование должно укладываться в узкие временные рамки.

Я думаю, что общее впечатление по поводу тенденций формальной дискретной математики второй половины XX века я изложил. Основная идея в том, что язык описания задачи должен быть дискретно-конечным. Для практического решения задач на компьютере все решаемые задачи должны находиться в классе **FP-SPACE**, то есть в классе функций, вычисляемых на машине Тьюринга, использующей ленту с числом ячеек, не превосходящим некоторого полинома от длины записи исходных данных. Более того, современным гарантом отсутствия полиномиальной масштабируемости (по числу шагов решения) задачи является доказательство NP-полноты этой задачи.

2. ВЛИЯНИЕ ИНФОРМАТИКИ НА ПРЕПОДАВАНИЕ НЕКОТОРЫХ РАЗДЕЛОВ ФОРМАЛИЗОВАННОЙ МАТЕМАТИКИ СТУДЕНТАМ-ИНФОРМАТИКАМ

Многие тенденции в преподавании формализованной математики можно наблюдать уже сейчас. Ещё 30 лет назад курсы дискретной математики и математической логики не читались для студентов инженерных специальностей. Сейчас эти курсы читаются студентам практически всех специальностей и направлений, тесно связанных с использованием вычислительной техники.

Есть некоторые общематематические утверждения, недоказуемые в формальной арифметике, и примеры таких утверждений можно демонстрировать при обучении студентов. Кроме того, можно ввести ограничение – какими средствами при доказательстве разрешено пользоваться. Например, широко известно, что трисекция угла невозможна с помощью циркуля и линейки. В то же время её можно осуществить другими простыми дополнительными средствами.

Д. Гильберт предполагал, что всю математику можно формализовать в виде большой игры, аналогичной игре в шахматы, где играют по заранее оговорённым правилам. Но первая теорема Гёделя о неполноте формальной арифметики показала, что это невозможно. При этом её можно доказать всем

студентам и даже школьникам, хорошо знакомым с компьютером.

Для практических целей часто достаточно формальная арифметика для целых чисел, представимых в компьютере средствами одинарной или двойной точности. Возможно, что использование очень больших чисел – это язык, который позволяет уйти в некоторые почти бесконечности, и, как следствие, получаются очень долго работающие алгоритмы или алгоритмы вообще не возможны. На мой взгляд, многие условия можно описать на языке формальной арифметики конечных чисел компьютерного типа *integer* (особенно для целых чисел из отрезка $[-2^{15}, 2^{15} - 1]$). Тожественная истинность формул такой формальной арифметики может быть проверена алгоритмом, принадлежащим классу **FP-SPACE**. Более того, задача такой проверки является **P-SPACE**-полной.

Понятие числа абстрактное, но школьники его запросто воспринимают. Здесь дело не в абстрактности, дело в привычке. Если школьника обучают только численным алгоритмам, то, конечно, теорему Гёделя ему сложно осилить. Но если ему рассказывают о компиляторах и интерпретаторах, о возможности рассматривать и обрабатывать программу как строку, то он поймёт гораздо больше. Если школьник в состоянии понять, что существует однозначное соответствие между записями алгоритмов и числами, то доказательство теоремы Гёделя проводится довольно наглядно. Важно, что формализованные математические доказательства также построены на средствах математической логики.

Язык математической логики сложился к 1900 году, когда Д. Гильберт сформулировал свои знаменитые проблемы и заложил основы языка исчисления предикатов. Далее осталось формализовать аксиомы арифметики, а это вполне понятно и доступно даже современным школьникам, ориентированным на точные науки и знающим математическую индукцию. Дети, хорошо знакомые с работой на компьютере, знакомы и с математическим понятием алгоритма. Поэтому им можно более кратко излагать то, что для

человека, незнакомого с компьютером, может вызвать существенные проблемы понимания.

Вспоминаю анекдотический случай про кванторы. Однажды при занятиях с учителями математики выяснилось, что им инспектор запрещал в преподавании школьникам использовать кванторы, потому что дети и так перегружены, и новые значки им становятся непонятными. Но школьники могут пользоваться этими знаками как стенографией. При использовании условного знака легче понять, о чём идёт речь, когда вместо разных длинных слов используется один и тот же символ.

По существу, математическая логика всё-таки начинается не там, где используется только булева алгебра, а там, где используются кванторы. Без кванторов мало что можно записать. Например, когда формулируется закон коммутативности сложения в виде $x + y = y + x$, то здесь неявно присутствует квантор всеобщности по x и y (для всех чисел x и y).

Есть в формализованной математике и другие совсем парадоксальные вещи. Например, университетский курс математического анализа можно сократить раза в два, используя нестандартный анализ, в основе которого лежат бесконечно малые и бесконечно большие числа. Эта идея восходит ещё к Г. Лейбницу, который создал математический анализ, не имея точного математического понятия (которое было сформулировано в 60-е годы XX века) как бесконечно малого, так и бесконечно большого числа. Если мы не выходим за пределы некоторого специального элементарного языка, то можем обращаться с бесконечно малыми так же, как с рациональными числами. Есть числа натуральные, а есть гипернатуральные – объединение натуральных с бесконечно большими натуральными числами. При этом если мы обозначим бесконечно большое число посредством w (удовлетворяющее аксиоме, что оно больше каждого натурального числа), то $w - 1 < w$. Это можно доказать. То есть получается другая арифметика, оперирующая как с конечными, так и с бесконечно большими натуральными числами. Имеет

смысл рассматривать не все вещественные числа, а только гиперрациональные, то есть числа, которые входят в формализованную аксиоматику рациональных чисел и их множество является нестандартным расширением множества рациональных чисел. И e , и π являются как раз такими гиперрациональными числами.

В основе современных точных описаний мира лежит формализованный логико-математический язык и, в первую очередь, конечно, алгоритмический. Но здесь есть один существенный момент. В алгоритмическом языке ограниченной сверху сложности желательно использовать кванторы только по конечным множествам, явно задаваемым списками.

3. ЖЕЛАТЕЛЬНЫЕ ИЗМЕНЕНИЯ ЧИТАЕМЫХ МАТЕМАТИЧЕСКИХ КУРСОВ ДЛЯ СТУДЕНТОВ-ИНФОРМАТИКОВ

Должен быть большой вес у информатики, дискретной математики, математических основ информатики. Но традиционно широко распространены концепции, связанные с непрерывными объектами. Например, концепция о том, что мир непрерывен и его можно описать с помощью вещественных чисел. Однако, компьютер не может работать с произвольными вещественными числами. Желательно определять понятие конструктивного вещественного числа как алгоритмической последовательности двоично-рациональных чисел, абсолютная величина разности членов которой, начиная с n -го, меньше, чем 2^{-n} .

Раньше говорили, что в любой науке столько науки, сколько в ней математики. Поэтому в астрономии и механике много науки. Теперь следует говорить, что в любой науке столько фундаментальных компьютерных приложений, сколько в ней математической (а не инженерной) информатики.

Приведу пример из физики. Возьмём то, что физики называют «термояд». Сколько лет обещают запустить реактор, вырабатывающий энергии больше, чем в нём тратится? Проинтегрируем сумму затраченных

усилий – сколько людей над этим работало, сколько конференций прошло, сколько грантов получено, сколько книг было выпущено, сколько учёных приобрели мировые имена, сколько кандидатов и докторов наук приобрели учёную степень – и мы почувствуем, что это направление само в себе. Такие разделы есть в любой науке – будь то физика, химия или математика. Единственный способ окончания этого – сокращение финансирования таких разделов науки. Но это социальный вопрос. Кроме того, а вдруг когда-нибудь что-нибудь получится? Пусть не то, что искали, но что-то полезное.

Иногда мне хочется покритиковать какую-нибудь науку. Проще критиковать людей, которых я лучше знаю, например, тех, которые занимаются в области, близкой к прикладной математике. Но после того, как у меня возникает такое желание, я вдруг вспоминаю, что есть огромное количество политических, психологических, юридических, филологических и разных других наук и их разделов, которые, как правило, используют математику в значительно меньшей степени, потому что в принципе не могут её использовать, поскольку их наука находится на стадии, как сейчас это принято говорить, гуманитарного оформления. Ещё не выработаны математически чётко, даже не очерчены понятия в этих науках. Кроме того, разные авторы под одним и тем же словом (например, под словом «революция») понимают разные вещи. И желание критиковать своих коллег сразу пропадает, когда я понимаю, что есть много других наук, которые в ещё большей степени заслуживают существенной критики. Я не хочу сказать, что не нужно заниматься гуманитарными науками, но там удаётся получать результаты намного туманнее, противоречивее и слабее, чем в науках, тесно связанных с математикой. Очень часто в гуманитарных науках результаты оказываются такими: может быть так, а может быть и не так. Получается, что такая наука заключается только в коллекционировании. Мне кажется, что должна быть предсказательная функция в каждой науке.

В принципе, способов мышления может быть много, и, если заложить в человека

слишком много способов мышления, то у него может наступить, как раньше говорили программисты, переполнение. Дело в том, что школьнику или студенту трудно выбрать, что перспективно, а что нет в ближайшие 10–20 лет. Часто студента привлекает какое-то популярное понятие, хочется его изучить. А то, что это никакого практического или теоретического применения не находит и в ближайшем будущем не будет находить, ему не видно. Для таких студентов обучение – это своего рода игра, по ходу которой им рассказывают что-то новое. Но когда студент приходит в аспирантуру, игра кончается, поскольку ему много нового не рассказывают, но от него требуют создать что-то своё новое. А он иногда уже не может что-то это своё новое придумать.

Вопрос о том, чему учить студентов, всегда сложный. Я сторонник такой либерализации образования, чтобы студент мог выбрать курс между двумя преподавателями, а ещё лучше – между тремя. Дело не в том, что один преподаватель лучше, а другой хуже. У каждого преподавателя и у каждого студента есть свои особенности, и надо дать студенту возможность почувствовать себя более комфортно. Однако, всё это невозможно, если экономить на обучении математических основ информатики, что часто делается в инженерных вузах.

Наконец, нужно отметить, что доценту кафедры информатики В.А. Костину удалось разработать программу обучения информатике и реализовать её на математико-механическом факультете Санкт-Петербургского государственного университета. Разумеется, в этом принимали участие его коллеги, прежде всего из отделения информатики, а особенно кафедры информатики, а также математиков-преподавателей математико-механического факультета СПбГУ. Эта программа обучения послужила основой программ, реализованных во многих городах Российской Федерации по специальности, а последние 2 года и по направлению «Математическое обеспечение и администрирование информационных систем». Студенты, обучившиеся этой специальности, способны анализировать достаточно детально описан-

ные математические алгоритмы, выбирать наиболее эффективные из них применительно к конкретной задаче и разрабатывать на их основе эффективные программы. Это связано с тем, что студентам математически грамотно читаются основные математические дисциплины, связанные с теми или иными описаниями математических алгоритмов. Более 60 вузов Российской Федерации выпускают студентов по этой специальности несколько лет подряд, что говорит о существенной востребованности этой специальности, готовящей математиков-программистов.

Упомянутые специальность и направление были созданы и функционируют в рамках Учебно-методического объединения (УМО) при Санкт-Петербургском государственном университете.

Следует отметить, что на отделении информатики математико-механического факультета СПбГУ в настоящее время (как правило, много лет) ведётся обучение по следующим специальностям (5 лет обучения) и направлениям (4 года обучения), тесно связанным с информатикой: «Математическое обеспечение и администрирование информационных систем», «Программная инженерия», «Информационные технологии», «Фундаментальная информатика и информационные технологии», «Прикладная информатика в области социологии», «Прикладная информатика в области международных отношений». На другом отделении того же факультета СПбГУ обучаются по специальности и по направлению «Прикладная математика и системное программирование». Наиболее длительным по годам выпусков (среди упомянутых специальностей) было обучение по специальности «Математическое обеспечение и администрирование информационных систем», в значительной степени посвящённой обучению разнообразному программированию математически точно сформулированных задач.

Значительный опыт автора преподавания ряда современных дисциплин, связанных с затронутыми в статье вопросами, студентам-информатикам отражён в учебных пособиях [1–5].

Литература

1. Бабаев И.О., Герасимов М.А., Косовский Н.К., Соловьев И.П. Интеллектуальное программирование. Турбо-Пролог и Рефал-5 на персональных компьютерах. СПб.: Изд-во СПбГУ, 1992.
2. Косовский Н.К. Элементы математической логики и её приложения к теории субрекурсивных алгоритмов. Л.: Изд-во ЛГУ, 1981.
3. Косовский Н.К. Основы теории элементарных алгоритмов. Л.: Изд-во ЛГУ, 1987.
4. Косовский Н.К., Тишков А.В. Логика конечнозначных предикатов на основе неравенств. СПб.: Изд-во СПбГУ, 2000.
5. Яхонтов С.В., Косовский Н.К., Косовская Т.М. Эффективные по времени и памяти алгоритмические приближения чисел и функций. Учебное пособие. СПб.: Изд-во СПбГУ, 2012.

Abstract

Revolutionary trends of the XXth century in mathematics formalization are mentioned. The influence of computer science upon the teaching of some formalized mathematical themes to the computer science students is specified. Desirable changes of mathematical courses for computer science students are offered. Speciality and direction «Mathematical support and administration of information systems» developed and implemented, first of all, by the professors of faculty of mathematics and mechanics of St. Petersburg State University are described. Some text-books describing the author's experience in the teaching of some formalized mathematical themes to the computer science students are listed.

Keywords: formalization, discrete simulation, continuous simulation, algorithm complexity, **FP-SPACE**, mathematical support, computer science.



Наши авторы, 2012.
Our authors, 2012.

*Косовский Николай Кириллович,
доктор физико-математических
наук, профессор, заведующий
кафедрой информатики
математико-механического
факультета СПбГУ,
kosov@NK1022.spb.edu*