С.Ю.Знатнов

О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ КОМПЬЮТЕРНЫХ ДОКАЗАТЕЛЬСТВ

Abstract. *In clause the review and classification of the software is made for carrying out of computer proofs.*

Попытки автоматизации вычислительного процесса известны еще издревле. В Средневековье, Новом времени и позже они даже получили техническую реализацию – в качестве примера можно привести логическую машину Луллия, счетные машины Лейбница и Паскаля, “фортепиано” Джевонса. В рамках нашей статьи мы рассматриваем современную ситуацию в области автоматизации доказательств, возникшую примерно в середине прошлого столетия и до сих пор находящуюся в процессе становления. Определяется начало этой ситуации появлением электронной вычислительной техники и ее использованием в различных отраслях знания. Более всего нас будет интересовать использование ЭВМ в решении логических и математических задач.

ЭВМ в настоящее время достаточно часто используют для доказательства теорем. Рассмотрим некоторые способы классификации программ, производящих указанные вычисления. Во-первых, их можно разделить на два класса: доказательства уже доказанных теорем (на примере работы программы Логик-Теоретик) и доказательства (или опровержения) теорем, впервые полученные с помощью компьютера. Это один вариант классификации.

Другой вариант классификации - по возможности реально перепроверить результат, полученный компьютером в процессе вычислений. И здесь мы будем применять термины “обозримая” и “необозримая” машинная процедура и – соответственно – теорема. Ситуации, когда некоторыми достижениями мы обязаны исключительно возможностям компьютера, становятся достаточно привычными; список таких задач постоянно пополняется. Один из последних примеров подобного рода – бурный компьютерный поиск простых чисел (подробнее ниже). Но то, что нам кажется привычным, в данном случае должно подвергаться строгой критике и тщательному анализу.

Еще один признак для классификации - автономность программы в поисках результата. Принимает ли участие человек в рассматриваемой процедуре или программа работает по заданному

алгоритму, не прибегая к его помощи и оценке возникающих ситуаций.

Другая классификация - по типу результата: что мы получаем на выходе? Доказательство теоремы или опровержение исходного математического утверждения?

Кроме того, программа может являться обширной системой дедуктивного вывода или применяться для доказательства лишь одной теоремы.

Можно найти и другие - менее существенные - признаки для классификации, как-то: среда программирования, модель ЭВМ, время работы программы. Но эти признаки будут уже не столь значимыми.

Повторные доказательства и новые. Теперь опишем подробнее различные виды классификации, обозначенные выше, с указанием их характерных особенностей и примеров соответствующих программ.

Итак, (пере)доказательство уже доказанных теорем или доказательства, впервые полученные на ЭВМ. К числу программ первого типа относится уже ставшая классической программа Логик-Теоретик[1]. Авторы программы Г. Саймон, А. Ньюэлл и Дж. Шоу, используя достижения в области информатики и психологии, разработали систему, которая передоказала многие теоремы математической логики. «С помощью программы Логик-Теоретик заново доказано 38 из 52 теорем одного из разделов математической логики - исчисления высказываний, содержащихся в книге Б. Рассела и А.Уайтхеда "Principia Mathematica". В дальнейшем на ЭВМ с большим быстродействием удалось вывести все 52 теоремы»[2]. Доказательства, найденные системой эвристического поиска, в некоторых случаях были достаточно оригинальными и получили высокую оценку одного из авторов фундаментального труда: «Б. Рассел, ознакомившись с этими доказательствами, выразил восхищение полученными результатами»[3]. В основе метода доказательства программы Логик-Теоретик лежали «психологиче-

---

[1] *Ньюэлл А., Шоу Дж., Саймон Г.* Эмпирические исследования машины «Логик-Теоретик»; пример изучения эвристик // Вычислительные машины и мышление. М.: Мир, 1967. Надо заметить, что программы, передоказывающие уже существующие доказательства, относятся к раннему этапу становления компьютерных программ для доказательства теорем. Они (практически) не давали новых математических знаний, но были необходимы для накопления соответствующего опыта.

[2] Логика и компьютер. Вып. 1. М.: Наука, 1990. С. 101.

[3] *Рузавин Г.И.* Интуиция и понимание в математике // Интуиция, логика, творчество . М., 1987. С.87.

ские идеи, связанные с поиском решений человеком в ситуациях, похожих на блуждание в лабиринте»[4].

Рассматривая метод полного перебора в некотором дереве поиска, можно произвести сравнение в природе поиска нового знания человеком и компьютерной программой. У человека часто изначально возникает некоторая интуитивная догадка, гипотеза, и затем производится ее обоснование. Появляется цель и производится поиск рационального пути к ней.

При работе компьютерной программы производится слепой поиск в некотором пространстве, блуждание по темному лабиринту, приводящее, возможно, к никем не поставленной цели (к тому, что древние греки называли поризмами). Если же цель задана изначально, то программа определяет (опять же методом перебора), существует ли путь к этой цели. В.П.Оревков в приложении к работе . «Математическая логика и автоматическое доказательство теорем»[5] так оценивает эффективность работы методом полного перебора, называя его алгоритмом Британского музея (АБМ): «...алгоритм Британского музея последовательно порождает все возможные выводы в каком-нибудь варианте исчисления предикатов и прекращает работу в тот момент, когда *наткнется* (курсив мой. - С.З.) на вывод исходной формулы. АБМ совершенно непригоден для фактического осуществления поиска вывода даже с использованием любых физически возможных ЭВМ»[6]. Другие примеры подобных программ, связанных с машинным доказательством уже известных теорем геометрии, можно найти в работах Гелернтера[7] и Невинса[8]. В работе последнего методика доказательства теорем в программе сводится к тому, что к базе данных добавляется новое утверждение по допустимым правилам вывода. Если это утверждение совпадает с заданной целью, то считается, что доказательство проведено и программа останавливается. Если совпадения не происходит, то рассматриваются новые следствия из полученного утверждения. «Программа была написана на языке ЛИСП. Был доказан ряд геометрических теорем на плоскости на машине PDP... Она доказала все теоремы,

---

[4] Информатика / Сост. Д.А. Поспелов. М.: Педагогика-Пресс, 1994. С.215.

[5] *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Мир, 1983.

[6] Там же. С. 314.

[7] *Гелернтер Г.* Реализация машины, доказывающей геометрические теоремы // Вычислительные машины и мышление. М.: Мир, 1967.

[8] *Невинс А.* Доказательство теорем планиметрии с использованием прямых рассуждений // Кибернетический сборник. Новая серия. М.: Мир, 1979. Вып. 16. С. 145-170.

приведенные Гелернтером, и ни для одной из них не потребовалось более 5 секунд»[9]. Постановка задачи и вывод результатов работы программы производится в виде текста, представленного на формальном языке, разработанном для данной системы вывода. Например:

Дано: (PR(DC)(AB)), ((CQRB)(DA))

Доказать: (ES PQPB)

Тем не менее, доказательство представлено в виде *текста*, который может быть приведен к привычному виду (на формальном языке математической теории или на каком-либо естественном языке[10]), и само доказательство может быть проверено.

Теперь рассмотрим примеры применения ЭВМ для *доказательства или опровержения* математических утверждений, впервые проведенные с помощью компьютера. По своей сути это программы, производящие полный конечный перебор всех случаев. Но количество этих просматриваемых случаев настолько велико, что человеческих ресурсов для проведения подобных вычислений просто не хватит. Рассмотрим более подробно некоторые примеры таких программ.

Первый пример - опровержение общих математических утверждений с помощью компьютера. Опровержение строится путем приведения контрпримера. Такие программы работают (назовем условно) в зоне, недосягаемой для человеческих возможностей.

Одним из полученных таким образом результатов является опровержение гипотезы Л.Эйлера:

Для любого показателя r≥3 диофантово уравнение $n^r = n_1^r + n_2^r + n_3^r + \ldots + n_s^r$ не имеет решений в натуральных числах, если s<r.

«В течение более 200 лет никому не удавалось ни доказать, ни опровергнуть эту гипотезу. И только уже в наши дни группа американских ученых с помощью мощного компьютера получила ... всего один-единственный результат $-144^5=27^5+84^5+110^5+133^5$»[11].

Если рассмотреть полученный результат с точки зрения его обозримости, то получается следующее. Произвести те же вычисления вслед за компьютером абсолютно невозможно. Поэтому проведенный процесс можно считать необозримым. С другой стороны, полученный результат вполне отвечает свойству проверяе-

---

[9] Там же. С. 165.

[10] Кстати, с такой задачей трансляции вполне справится несложная компьютерная программа.

[11] *Зенкин А.А*. Когнитивная компьютерная графика // Материалы XI Межд. конф. по логике, методологии и философии науки. Москва - Обнинск, 1995. Том 2. С. 129.

мости - достаточно вычислить выражения в обеих частях равенства, чтобы убедиться, что гипотеза Эйлера неверна:

61917364224 = 14348907 + 4182119424 + 16105100000 + 41615795893.

Реализация подобного рода алгоритмов возможна по следующей структурной схеме:

Repeat
<вычисления>
Until <проверяемое условие>[12].

Цикл подобного рода не является арифметическим (или счетным) - нам не известно заранее, сколько раз будут произведены нужные вычисления. Это цикл по условию, где в качестве условия, прерывающего цикл, поставлено проверяемое выражение. Конечен этот цикл или нет? В приведенном примере - конечен, если достигнут требуемый результат. Пока же результат не достигнут, программа будет продолжать свой счет. Но даже если она будет работать очень долгое время, нельзя будет сделать никакого позитивного вывода о том, когда она закончит счет. Нам не известно, к сожалению, была ли остановлена программа, достигшая вышеназванного результата. Но ничто не мешает нам предположить, что если работа программы продолжается, то она может выдать еще один результат, нарушающий гипотезу Эйлера. А может быть, и не один…

Теперь о доказательстве истинности общих математических утверждений с помощью компьютера. В качестве примера доказательств подобного рода известно доказательство Аппелем и Хакеном проблемы четырех красок - любую карту можно раскрасить, используя 4 цвета, причем, любые две соседние страны должны быть окрашены в разные цвета. Данное утверждение было доказано для общего случая при n>2000. 2000 же видов карт требовали отдельного перебора, который и был организован с помощью ЭВМ. «Мы использовали 1200 часов на 3 различных компьютерах. Заключительная процедура разгрузки отличалась от нашего первого приближения почти 500 модификаций, следующими из открытия критических окрестностей. Развитие процедуры требовало ручного анализа приблизительно 10000 соседств и машинного анализа более чем 2000 конфигураций. Значительная часть этого материала, включая сокращение до 1482 конфигураций, использовалась в заключительном доказательстве»[13], - пишут о

[12] Синтаксис языка Pascal. Повторять <вычисления> до тех пор, пока не станет истинным <условие>.

[13] *Appel K., Haken W.* The solution of the four-color-map problem // Scientific American.1977, ¹ 10. P.121.

своей работе Аппель и Хакен. Если бы те же вычисления проводились людьми, то на это пришлось бы потратить «140000 полномасштабных человеческих жизней без сна и перерывов на трапезу»[14].

Обозримые и необозримые процедуры. Приведенный выше пример как раз и принадлежит к тем случаям доказательств, которые мы условились называть необозримыми процедурами. Если представить структурную схему подобных доказательств на некотором языке программирования, то она будет выглядеть следующим образом:

```
        For n:=p to k do
        <вычисления и проверка>
или
        For n:=k downto p do
        <вычисления и проверка>
```

В отличие от цикла по условию при опровержении утверждений здесь, как мы видим, используется арифметический цикл, в котором переменная цикла пробегает все натуральные значения на интервале от p до k (или в обратном порядке). Если цикл будет отработан полностью, то, следовательно, для любого n из указанного интервала проверено сформулированное утверждение. Остановку работы программы можно организовать в том случае, если для некоторого значения n это предложение ложно. А можно организовать работу программы так, что она зафиксирует все случаи, когда при заданном n исходное утверждение ложно. Тогда формулировку данного утверждения можно будет произвести с поправкой на эти случаи.

Подведем некоторый итог по этому поводу: компьютерные доказательства теорем, требующие огромных человеческих временных ресурсов и вычислительных способностей, реально существуют и не удовлетворяют свойству проверяемости обычных "человеческих" доказательств. Кстати, формально никак не определен критерий для названных нами "огромных человеческих временных ресурсов и вычислительных способностей". Только соображения здравого смысла и желание (или нежелание) потратить время на рутинную работу определяют в данном случае степень обозримости того или иного компьютерного доказательства.

Задачи и теоремы, которые мы отнесли к разряду необозримых, можно назвать задачами переборного типа, так как каждая из них требует огромного комбинаторного перебора. И при этом указанный перебор – единственный пока метод решения

[14] *Зенкин А.А*. Указ. соч. С. 130.

подобных задач. Об этом в самом начале дискуссии о проблемах доказательства теоремы о четырех красках говорил Пол Теллер (Paul Teller) в своей статье "Компьютерное доказательство": «МАТЕМАТИКИ решили старую четырех-цветную проблему о том, что каждая карта может быть окрашена с использованием только четырех цветов. Их доказательство, однако, имеет новый поворот: оно требует некоторых комбинаторных проверок, которые слишком долго делать вручную... Доказательство использует критическую лемму, используя проверку комбинаторного характера. Список комбинаций, которые должны быть проверены, безнадежно велик для человека, чтобы его произвести; так что эта часть работы должна быть выполнена компьютером»[15]. Решение задач переборного типа является в настоящий момент одной из фундаментальных проблем математики и вместе с ней - философии. Только использование компьютеров для организации подобных вычислений позволяет решить подобные задачи. Отмечая значимость проблемы, М.Гэри и Д.Джонсон пишут в своей работе: «Анализ трудностей, встретившихся на пути создания новых методов решения дискретных задач, привел к постановке центральной теоретико-методологической проблемы всей дискретной математики - можно ли исключить перебор при решении дискретных задач? До настоящего времени эта проблема остается открытой»[16]. И хотя эти слова были написаны более двадцати лет назад, они остаются актуальными и сейчас. Некоторые современные авторы говорят о текущей ситуации как об экстремальной и сравнивают ее с предшествующими кризисами в математике. Выход из создавшейся ситуации, по мнению автора приводимых далее строк, состоит в смене метода: «логика в дискретной математике должна уступить место интуиции как основному и самодостаточному методу исследования. ...Для развития дискретной математики более перспективным является развитие навыков интуиции как основного способа познания истины»[17]. Развитие интуитивных способностей, по словам В.З.Стрыгина, должно стать едва ли не главной задачей современной школы. Не будем отрицать категорически эти предположения, но заметим, что развить уровень интуитивного мышления до способностей Раманужана, на наш взгляд, так же «естественно», как

---

[15] *Teller P*. Computer proof // J. Of Philosophy, 1980. Vol. 77, N 12. P.797.

[16] *Гэри М., Джонсон Д*. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. С. 6.

[17] *Стрыгин В.З*. Критика формальнологического метода исследования в дискретной математике. М.: Изд. отд. ЦАГИ, 1998. С. 6-7.

«естественно» перепроверить доказательство теоремы о 4-х красках вручную...

В качестве примера задач рассматриваемого нами типа можно привести задачу определения чисел на простоту. Изначально такие программы составлялись исключительно из любопытства и интереса авторов этих программ к оригинальности составления алгоритма для поиска простых чисел. Позже результаты работы этих программ нашли применение в криптографии, где используются и сейчас. Первые примеры описаний работы программ можно найти в новой серии Кибернетического сборника середины восьмидесятых годов[18]. Масштаб производимых компьютером вычислений осознаешь, прочитав: «Это первый существующий тест проверки простоты, который без труда исследует числа с сотнями десятичных цифр... Программа на Паскале может работать с числами до 104 десятичных цифр, а программа на Фортране - с числами до 213 десятичных цифр»[19]. Даже один из найденных результатов такого порядка проверить безмашинным методом будет очень проблематично: 10^{213} нужно будет «поделить» на все числа (или все простые числа – в зависимости от избранного алгоритма) от 2 до 10^{107}!

Современные исследования на эту тему еще более впечатляют. Вот уже несколько лет действует проект GIMPS (Great Internet Mersenne Prime Search), поддерживаемый компанией Entropia. Цель проекта – с помощью организации распределенных компьютерных вычислений находить все новые и новые простые числа (причем, из класса чисел Мерсенна – те, которые можно представить в виде $n=2^p-1$). Приведем несколько достигнутых результатов с указанием использованных ресурсов.

Декабрь 2001 года – найдено число $n=2^{13\,466\,917}-1$. Над решением одной задачи одновременно трудились 130 тыс. добровольцев и 210 тыс. персональных компьютеров во всем мире. В общей сложности Gimps потребовалось затратить 13 тыс. лет машинного времени. Цифровая запись найденного простого числа содержит 405396 символов!

Декабрь 2003 года – найдено $n=2^{20\,996\,001}-1$ - сороковое по счету обнаруженное число Мерсенна. Затраченные ресурсы – примерно того же порядка. Количество цифр числа - 6 320 430.

Одно из последних достижений состоялось в июне 2004 года. Найденное число – $n=2^{24\,036\,583}-1$. Оно состоит более, чем из семи

[18] См статьи: Василенко О.Н. 1988. Вып.25; Коэн Х., Ленстра Х. Мл. 1987. Вып.24; Уильямс Х. 1986.Вып.23.

[19] *Коэн Х., Ленстра Х. мл.* Указ. сборник. С. 101, 143.

миллионов цифр. К сожалению, на текущий момент по этому результату дополнительной информации не найдено.

Именно доказательства математических утверждений и решения задач, которые мы отнесем к разряду необозримых, и представляют для нас особый интерес. Они приводят нас к новому знанию, но перепроверить правильность доказательства оказывается невозможным. (Здесь под “невозможным” мы понимаем невозможность буквально - шаг за шагом - повторить этот процесс человеком. Поэтому понятие необозримого доказательства можно уточнить и говорить о необозримости 1-го и 2-го рода. Необозримость 1-го рода означает, что нельзя провести те же действия, которые проделал компьютер, но можно провести проверку результата на истинность. Пример - приведенное опровержение гипотезы Эйлера. При необозримости 2-го рода нет возможности ни провести вычисления, ни проверить результат.)

Типы программ - пруверов. Далее рассмотрим вопрос об автономности программ-пруверов. Обратимся сначала к термину “прувер”. Не любая программа, производящая доказательство, может быть названа данным термином. Он относится к той части программ, которые организуют поиск вывода (в традиционном логическом понимании этого термина) в некоторой логической системе. К примеру, программа, с помощью которой была опровергнута гипотеза Л.Эйлера, не относится в нашей классификации к пруверам в узком смысле. Итак, прувер - компьютерная программа, в которой в качестве начальных условий содержится описание некоторой логической системы - это может быть зафиксированная система, либо выбираемая из некоторого набора систем, либо определяемая пользователем. "Описание содержит сигнатуру, правила вывода и аксиомы и составляется на специальном языке, позволяющем описывать как синтетические, так и аналитические правила".[20]

По степени автономности в работе и принятии решений такие программы мы разделим на следующие классы:

- прувер - автомат (ПАВ): задается начальная система аксиом и набор правил вывода. Программа совершает слепой или эвристический поиск, производя обозримые и (возможно) необозримые процедуры без вмешательства человека;
- прувер - ассистент (ПАС): начальный набор - тот же, что и в предыдущем случае, но возможно вмешательство человека

[20] *Новодворский А.Е., Смирнов А.В.* Интерактивная система поиска вывода в различных логических системах // Материалы XI Международной конференции по логике, методологии и философии науки. Москва - Обнинск, 1995. С. 164.

в вычислительный процесс для оценки возникающих в процессе вывода ситуаций и определения стратегии поиска;

- прувер - протокол (ПП): выбор начальной системы аксиом, правил вывода; фактически - хороший помощник для ведения протокола, сам вывод осуществляется в интерактивном режиме.

Приведем примеры соответствующих программ. Представитель первого класса - уже упоминавшийся нами «Логик – Теоретик». Подтверждением тому, что программа была самостоятельной в выборе решения, является то, «что в одном случае «Логик-теоретик» предложил даже более короткое и остроумное доказательство, чем данное Расселом и Уайтхедом»[21]. Программа работала самостоятельно, используя начальный набор аксиом и правил вывода. Используя метод перебора, она нашла пути до многих теорем.

Варианты перебора могут быть различными.

Во-первых, это метод полного перебора. Используя этот метод, программа систематично, шаг за шагом, перебирает все возможные варианты. Другой метод - перебор с использованием эвристических правил. Этот метод и был использован авторами «Логика...» - «создатели снабдили ее эвристикой обратного рассуждения. Она заставляет программу прежде всего проверить, какие исходные положения должны быть истинными, если считать теорему верной»[22]. Но, как правило, эти методы используются не обособленно, а согласованно. «В большинстве программ, предназначенных для решения задач, используются оба метода поиска: сначала для «обрезки» дерева применяются эвристики, а затем прибегают к полному перебору на оставшейся его части»[23].

Третий способ - метод случайного поиска. Если первые два метода обязательно приводят к цели, то здесь достижение цели может и не произойти, так как здесь возможно повторное прохождение путей. То есть к цели мы придем случайно.

В качестве программ класса ПАС можно назвать оболочки экспертных системы (ЭС), работающие в различных областях человеческой деятельности, своеобразные машины логического вывода. Один из примеров – экспертная система DENDRAL. Экспертные системы – одна из отраслей в системе искусственного интеллекта. Основными составляющими любой ЭС являются база

[21] Компьютер обретает разум. М.: Мир, 1990. С. 34-35.
[22] Там же. С. 36.
[23] Там же. С. 39.

данных и система логического вывода, которую и можно назвать в нашей терминологии пруверо м-ассистентом.

В математике пруверы-ассистенты тоже находят свое применение. Такую ситуацию предвидел В.М.Глушков, когда писал в своих работах: «Поскольку дедуктивные построения над языком математики будущего по необходимости должны быть гораздо сложнее, успешное развитие математики и ее приложений в других науках станет невозможным без автоматизации этих построений. ...Разработка специального языка «подсказок» приведет к эффективной совместной работе математика с ЭВМ по доказательству новых теорем»[24]. Одна из реализаций этого предвидения – в работе Л.Воза, где описана программа AURA, с помощью которой автор исследовал проблемы тернарной булевой алгебры, конечных полугрупп и т.д. Существенная характеристика, которую автор статьи дал указанной программе, состояла в том, что программа вела себя как коллега.

К программам последнего типа - ПП - можно отнести одну из отечественных разработок - программу Deductio, описанную в третьем выпуске сборника “Логика и компьютер”. «Программа Deductio поддерживает построение выводов в логических системах, определенных пользователем. Это своего рода специализированный текстовый процессор, реализующий символические вычисления в логике первого порядка»[25]. «Программа Deductio позволяет осуществлять в режиме диалога процедуру поиска вывода. При этом она контролирует соответствие всех изменений дерева поиска вывода системе поисковых правил, и тем самым гарантирует правильность построенного вывода»[26].

Итак, мы рассмотрели основные, на наш взгляд, способы классификации компьютерных доказательств. Несмотря на все это многообразие, будем считать, что наибольший интерес представляют «прежде всего те полученные с применением компьютеров теоремы, которые при сегодняшнем уровне знаний не могут быть доказаны без ЭВМ ввиду необходимости огромного количества расчетов, превышающего возможности человека»[27].

---

[24] *Глушков В. М*. Кибернетика: вопросы теории и практики. М.: Наука, 1986. С. 97, 99.

[25] *Новодворский А.Е., Смирнов А.В*. Указ. соч. С. 164.

[26] Логика и компьютер. Вып. 3. М.: Наука, 1996. С. 5.

[27] *Анисов А.М*. ЭВМ и понимание математических доказательств // «Вопросы философии». 1987, № 3. С.30.