

А.В. Анисимов, Т. Джебелян, А.В. Лялецкий. Н. Попов

Системы *Theorema* и автоматизация дедукции: сравнительный анализ

Рассмотрены основные парадигмы обработки математических знаний и обозначено место среди них систем *Theorema* и автоматизация дедукции. Описаны сравнительный анализ систем, общие черты их построения и различие в методах обработки данных.

Basic paradigms of the mathematical knowledge processing are considered and the place of the *Theorema* and SAD systems is outlined. A comparative analysis of the systems, common features of their construction and the distinction in data processing methods are described.

Розглянуто основні парадигми обробки математичних знань та позначено місце серед них систем *Theorema* та автоматизація дедукції. Описано порівняльний аналіз систем, загальні риси їх побудови та різницю в методах обробки даних.

Введение. В настоящее время повысилось внимание к созданию новых интеллектуальных сервисов как на базе интеграции уже имеющихся информационных систем¹, так и в результате создания новых компьютерных служб с использованием современных достижений в информатике и информационных технологиях. Это обусловлено тем, что решение задач автоматизации интеллектуальной деятельности человека требует умения компьютера проводить вычисления и/или аналитические преобразования, и/или логические рассуждения. Для достижения этих целей имеется достаточно большое количество различных служб, способных проводить числовые выкладки и/или символьные «вычисления», и/или дедуктивные построения в среде формального естественного языка и успешно применяемых для решения разнообразных задач автоматизации рассуждений.

Область автоматизации рассуждений представляет собой удачное сочетание теории и практики, хотя иногда оно оказывается искаженно толкуемым. С одной стороны, специалисты в прикладных вопросах центральной проблемой считают эффективное применение имеющихся теоретических результатов, но концентрация внимания только на реализации часто вынуждает их оставлять без внимания те фундаментальные проблемы, исследование которых необходимо для решения задач. С другой стороны, специалисты в фундаментальных вопросах обращают основное внимание на внутреннее теоретическое развитие своих исследований, акцентируя его на решении сложных

проблем. Вопросы же реализации трактуются ими как «чисто прагматические», «инженерные» задачи. Поэтому естественно стремиться к нахождению разумного баланса между этими двумя точками зрения, одновременно развивая в автоматизации рассуждений те теоретические и прикладные стороны математической логики, символьных преобразований и лингвистики, которые могут привести к эффективной реализации на компьютерах соответствующих подходов, методов и средств. Именно с этих позиций проводились разработка и реализация систем *Theorema* и автоматизация дедукции (САД).

Разработка и реализация системы *Theorema* [1] началась во второй половине 1990-х гг. в соответствии с одноименной программой [2]. Приблизительно в то же время были возобновлены работы по САД в англоязычной версии² [3] в рамках работ по реализации программы Алгоритм Очевидности [4]. Обе эти системы с самого начала их разработки были ориентированы на оказание разнообразной помощи человеку в его математической деятельности. Дадим более детальное их описание и проведем сравнение систем в рамках существующих парадигм автоматизации рассуждений и стилей обработки математических знаний.

Парадигмы обработки компьютерных знаний и системы *Theorema* и САД

На современном этапе можно выделить следующие парадигмы обработки компьютерных математических знаний.

¹ Такие системы в дальнейшем называются компьютерными математическими службами или просто службами.

² Русскоязычная версия САД [5] была разработана и реализована во второй половине 1970-х гг. в рамках работ по программе Алгоритм Очевидности.

Числовая парадигма отображает методы и средства приближенного или точного решения задач чистой или прикладной математики, что основывается на построении конечной последовательности действий над конечным множеством чисел. Обычно ее использование начинается с построения математической (континуальной) модели. После этого совершается переход к ее дискретному представлению, что позволяет разрабатывать различные алгоритмы числовых вычислений, которые впоследствии программируются на компьютерах в виде пакетов прикладных программ.

Символьная (аналитическая) парадигма базируется на возможности компьютера выполнять сложные символьные преобразования и числовые выкладки, строить графики функций, задавать модели необходимых процессов и др. Она ориентирована на построение и использование так называемых систем компьютерной алгебры (СКА) и возникла во второй половине 1960-х годов как альтернатива числовой парадигме в связи с появлением компьютеров такой скорости, информационной емкости и гибкости, что стало возможным программирование сложных интеллектуальных процессов. Появление СКА было вызвано также следующими обстоятельствами:

- наличием задач, которые не могут быть легко выполнены человеком при помощи ручки и листа бумаги, и для которых практически отсутствуют их адекватные представления в виде соответствующих числовых алгоритмов;
- уменьшением затрат времени и усилий на нахождение решения широкого круга научно-прикладных задач;
- компактностью записи и наглядностью аналитического решения поставленной задачи в сравнении с ее числовым решением.

Принципы построения СКА хорошо согласуются с эмпирическим опытом человека, приобретенным при решении различных математических и естественнонаучных задач. В то же время в современных СКА, как правило, полностью отсутствуют средства проведения логических рассуждений.

Дедуктивная парадигма использует тот факт, что существующие в компьютере знания име-

ют вид определенных синтаксических единиц (как правило, аксиом, определений, утверждений, теорем и пр.), а дополнительные знания получаются из имеющихся посредством применения к ним определенных схем умозаключений. Системы представления и обработки знаний, основанные на этой парадигме, получили название *систем автоматизации рассуждений*, значительная часть которых существует в виде систем доказательства теорем, поскольку именно логико-математический подход наиболее релевантен и эффективен при решении задач на построение цепочек дедуктивных рассуждений во многих прикладных областях (различаются декларативные и интерактивные системы автоматизации рассуждений).

Для того чтобы некоторая система автоматизации рассуждений могла оказывать дедуктивную помощь человеку, очевидно, что необходимо разработать подходящую языковую «оболочку», правильно сформулировать аксиомы и правила порождения новых утверждений из уже имеющихся, уметь генерировать предположения и гипотезы, которыми можно пользоваться, и пр.¹

Интеграционная парадигма. Можно выделить два основных типа системной интеграции: интеграцию на этапе проектирования, когда еще во время разработки системы предусматривается как наличие в ней компьютерных служб разного рода, так и возможность ее иерархического наращивания путем подключения к ней имеющихся служб (солверов, СКА и др.) и интеграцию на этапе эксплуатации, когда осуществляется предопределенное задачей комбинирование в одну систему готовых компьютерных служб (особое внимание к разработке систем такого типа вызвало появление информационных технологий, основанных на широком использовании сети Интернет).

¹ Конечно, в контексте данной работы в качестве альтернативы дедуктивной парадигме следует упомянуть *индуктивную парадигму*, которая охватывает процессы рассуждений, основанные на переходе от частных положений к некоторым гипотезам/общим утверждениям. (Такой способ рассуждений пока не нашел своего отражения в имеющихся версиях системы *Theorema* и САД).

Если теперь обратиться к системе *Theorema* и САД, то они могут рассматриваться как яркие представители интеграционной парадигмы первого типа (например, [6, 7]). Остановимся на этом подробнее.

Как упоминалось, система *Theorema* появилась в результате попытки реализации одноименного проекта, ориентированного в первоначальной постановке на создание декларативной логической надстройки [8] над известным пакетом компьютерной алгебры *Mathematica* [9]¹, что позволило позднее сориентировать ее на изучение математических теорий [10].

САД с самого начала задумывалась как декларативная дедуктивная система, предназначенная для оказания помощи математику в его научной деятельности. В ходе разработки САД ее архитектура проектировалась так, чтобы было возможно подключение различных математических служб, включая системы компьютерной алгебры и пруверы [12].

Опишем некоторые особенности и возможности системы *Theorema* и САД, сравнивая основные принципы их построения.

Система *Theorema* и САД в качестве своих входных языков используют формальные языки естественного типа, основное различие между которыми состоит в том, что входной язык САД представляет собой формальный естественный язык, который может быть транслируемым в обычный язык логики предикатов первого порядка, в то время как входной язык *Theorema* есть естественная разновидность языка логики высших порядков. Кроме того, *Theorema* включает в себя простой и удобный процедурный язык, обеспечивающий возможность проведения числовых вычислений и символьных преобразований в «контексте» логических рассуждений.

Выбор языков для системы *Theorema* и САД объясняет и различие в выборе их логических аппаратов: *Theorema* использует методы поиска вывода в логике высших порядков, а САД –

в классической логике первого порядка. Обе системы также оснащены и машинно-ориентированными эвристическими приемами доказательств, конечно, учитывающими логики, с которыми они работают. Дополнительно можно еще отметить, что собственный логический аппарат САД развивался так, чтобы можно было проводить поиск логического вывода в сигнатуре исходной теории с возможностью отделения обработки равенств от процесса дедукции для возможного использования внешних (по отношению к САД) солверов, в частности, систем компьютерной алгебры.

Для завершения описания принципов построения и сравнения системы *Theorema* и САД зафиксируем более четко их место среди перечисленных парадигм обработки математических знаний.

Прежде всего, обратим внимание на то, что в силу отмеченных языковых особенностей как *Theorema*, так и САД относятся к дедуктивной парадигме, базирующейся на декларативном подходе. Но при этом они используют разные логические аппараты.

Что же касается остальных парадигм, то наличие в системе *Theorema* средств процедурного программирования допускает реализацию на компьютере разных вычислительных алгоритмов, а возможность привлечения к ее работе системы *Mathematica* позволяет говорить, что *Theorema* использует не только вычислительную, но и символьную парадигму.

В системе же САД отсутствуют средства программирования в рамках вычислительной и символьных парадигм. Она спроектирована на принципе привлечения извне компьютерных служб, представляющих эти парадигмы, с учетом ее архитектуры и хорошо развитых языковых средств, дающих возможность интеграции различных подходов к обработке знаний.

Стили обработки данных в системе *Theorema* и САД

Целесообразно также сравнить систему *Theorema* и САД относительно стилей представления и обработки данных при решении задач автоматизации рассуждений, к наиболее интересным из которых принадлежат стиль прове-

¹ Позже *Theorema* начала развиваться и в других направлениях, например, использоваться для верификации программного обеспечения [11].

дения доказательств, степень подробности доказательств и стиль формализации.

Стиль проведения доказательства. Существенным свойством систем оказания помощи в автоматизации рассуждений есть вид их входных данных. Так называемые интерактивные системы (первой тип) чаще всего тактико-управляемы, а это значит, что заданное утверждение доказывается с помощью последовательности инструкций, вводимых в систему. Эти инструкции (т.е. тактики) могут быть примитивными, представляющими собой однократное применение правила вывода, или более сложными, подобными плану доказательства или привлечению к обработке рассматриваемой цели некоторого внешнего (по отношению к системе) пружера. К системам такого типа относятся *Isabelle* [14], *Coq* [15], *HOL* [16] и др. Работа с такими системами удобна и легка в случае, когда она предоставляет в распоряжение пользователя изящный набор мощных тактик, достаточных для решения задачи.

Второй тип систем ориентирован на декларативное представление решаемой задачи, отражающее стиль обычных математических публикаций, когда доказываемые утверждения и, при необходимости, их доказательства записываются на некотором формальном языке, который должен быть расширяем средствами (логического) структурирования текста. Верифицирующая система такого рода должна обладать способностью проверки правильности каждого шага доказательства, т.е. она должна управляться доказательством (при наличии такового).

Наиболее типичными образцами этого подхода для классической логики первого порядка есть САД и система *Mizar* [17], в то время как для логики высших порядков декларативный стиль наиболее отслежен в системе *Theorema*.

Заметим, что система *Isabelle*, расширенная средствами *Isar* [18] вместе с языком для записи структурированных доказательств, имитирующий стиль математических рассуждений, также может служить представителем систем, управляемых задаваемым доказательством.

Приведенное различие между двумя типами входных данных не очень существенно. Если

можно построить доказательства теорем с помощью тактико-управляемой системой, использующей промежуточный ввод целей и автоматическое закрытие подцелей, то такая система может рассматриваться как управляемая доказательством. Обратное, если шаги вывода, воспринимаемого некоторой системой, управляемой доказательством, снабжены детальными подсказками о приемах и методах верификации, то такая система может рассматриваться как тактико-управляемая.

Степень «грануляции». Дедуктивная мощь систем поддержки автоматизации рассуждений может меняться в зависимости от требований, выставляемых пользователем. Поэтому можно считать, что название таких систем колеблется между системами верификации доказательств и системами поиска доказательств. Первые принимают на вход только шаги доказательства, имеющие вид применений правил вывода, и, следовательно, должны быть детализированы с точностью до правил вывода. Система *Mizar* служит примером такой системы, хотя набор ее правил вывода достаточно велик.

Системы, относящиеся к системам поиска доказательств и/или верификации текстов, содержат методы поиска вывода и/или планировщики доказательств и пытаются восполнить «недостающие» места доказательства, используя «защитную» логику (логики). Системы САД и *Theorema*, а также *Nqthm* [19] и *ACL2* [20] – характерные представители систем такого рода.

Тактико-управляемые системы типа *Isabelle* и *Coq* обычно обладают исчерпывающими (для проведения доказательств) наборами тактик, так что их «дедуктивная мощь» может не иметь принципиального значения. И в этой связи они могут рассматриваться как системы построения доказательств. Однако некоторые эксперименты с *Isabelle* и *Coq* показывают, что следующая ситуация может оказаться критической для них: при всякой попытке «сразу же» доказать теорему «диалог» о построении доказательства становится сложным, сильно ветвящимся и тяжело отслеживаемым без предварительного выделения лемм, без применения

специальных тактик и использования существующих библиотек.

Стиль формализации. На стиль формализации обрабатываемого текста оказывают влияние предметная область, предварительные сведения, вид применяемых определений (являются ли они «компьютеризируемыми»), способ рассуждения (является ли он конструктивным, строго типизируемым или основанным на том или ином исчислении) и др. Также на стиль формализации влияют используемая базовая логика и теории, вовлекаемые в формализацию.

Существует два основных стиля формализации: первый ориентирован на применение логик высших порядков (с теорией типов в качестве математической базы), а второй – на логику первого порядка (как правило, с теорией множеств в качестве математической базы).

В силу изложенного, система *Theorema* относится к первому стилю формализации. Заметим, что теоретико-типовой подход применяется в большинстве хорошо известных систем, оказывающих математическую помощь таких, как *Isabelle*, *Coq*, *HOL* и др. При этом предпочтение его использованию, как правило, отдается в случаях обращения к индуктивно-определяемым областям и рекурсивным определениям. Он также хорошо подходит к формализации программистских и инженерных концепций.

К примерам второго стиля относится САД и упомянутая система *Mizar*, использующая логику первого порядка и теорию множеств Тарского–Гротендика. Такая ориентация *Mizar* соответствует традиционному способу построения математики; существует большой набор математических утверждений, проверенных в *Mizar*.

В отличие от *Mizar* система САД не адаптирована к какой-нибудь теории множеств (или другой фундаментальной теории), дающей общую основу для формализации. Вместо этого, для решения задачи пользователю предлагается задать необходимый набор предварительных сведений, выражающих базовые концепции на естественном формальном языке, транслируемом в определенную разновидность языка первого порядка; также требуется указать логиче-

ский прuver, наиболее релевантный для нахождения искомого решения.

Заключение. Итак, *Theorema* и САД базируются на идейно близких платформах, есть представителями интеграционной парадигмы с декларативным представлением знаний и обладают рядом как общих, так и различных принципов построения и стилей символьной и логической обработки данных. Это делает привлекательными попытки, которые могут быть совершены в направлении совершенствования САД путем применения оригинальных идей и опыта, полученного разработчиками системы *Theorema* в ходе ее имплементации и эксплуатации, и наоборот.

В долгосрочной перспективе исследования по дальнейшему развитию системы *Theorema* и САД, их развитию и взаимному обогащению могут привести к созданию мощной инфраструктуры по оказанию помощи как в проведении научных исследований, так и в образовании. Они могут оказаться полезными при решении задач верификации, а также при создании систем управления компьютерными знаниями.

1. Buchberger B., Jebelean T. *Theorema: The predicate logic prover* // Proc. of the First Intern. Theorema Workshop. – Hagenberg, Austria, 1997.
2. *A Survey of the Theorema project* / B. Buchberger, T. Jebelean, F. Kriftner et al. // Proc. of the Intern. Symp. on Symb. and Algebr. Comp. (ISSAC'97). – 1997. – P. 384–391.
3. *System for Automated Deduction (SAD): Linguistic and deductive peculiarities* / A. Degtyarev, A. Lyaletski, K. Verchinine et al. // Proc. of the 11th Intern. Symp. IIS 2002. – Poland, 2002. – P. 413–422.
4. Глушков В.М. Некоторые проблемы теории автоматов и искусственного интеллекта // Кибернетика. – 1970. – № 2. – С. 3–13.
5. *О системе обработки математических текстов* / Ю.В. Капитонова, К.П. Вершинин, А.И. Дегтярев и др. // Там же. – 1979. – № 2. – С. 48–55.
6. *Theorema: An integrated system for computation and deduction in natural style* / B. Buchberger, K. Aigner, C. Dupre et al. // RISC Report Series, Johannes Kepler Univ. of Linz, Austria. – 1998.
7. Lyaletski A., Verchinine K., Paskevich A. *Theorem proving and proof verification in the system SAD* // Lecture Notes in Computer Science (Proc. of the 3rd Intern. Conf. MKM 2004). – 2004. – 3119. – P. 236–250.

Окончание на стр. 77

8. *Buchberger B.* Theorema: A proving system based on Mathematica // *The Mathematica J.* – 2001. – 8(2). – P. 247–252.
9. *The Mathematica Book.* – <http://reference.wolfram.com/legacy/v5/TheMathematicaBook/index.html>
10. *Theorema: Towards Computer-Aided Mathematical Theory Exploration* / B. Buchberger, A. Craciun, T. Jebelean et al. // *J. of Applied Logic.* – 2006. – 4(4). – P. 470–504.
11. *Kovacs L., Popov N., Jebelean T.* Combining Logic and Algebraic Techniques for Program Verification in Theorema // *Proc. of Second Intern. Symp. on Leveraging Appl. of Formal Methods, Verification and Validation (ISOLA 2006).* – Paphos, Cyprus, 2006. – P. 59–68.
12. *Anisimov A., Lyaletski A.* The SAD system in three dimensions // *Proc. of the 7th Intern. Workshop on Symb. and Numeric Algorithms for Scientific Comp. (SYNASC 2006).* – Timisoara, Romania, 2006. – P. 85–88.
13. *Анисимов А.В., Джебелян Т., Лялецкий А.В., Попов Н.* Проекты Алгоритм Очевидности и *Theorema*: их особенности и реализация на современном этапе // Тез. доп. Міжнар. конф. *TAAPSD 2010.* – Київ, Україна, 2010. – С. 54–58.
14. *The Isabell.* – <http://www.cl.cam.ac.uk/Research/HVG/Isabelle/>
15. *The Coq system.* – <http://coq.inria.fr/>
16. *The system HOL.* – <http://www.cl.cam.ac.uk/Research/HVG/HOL/>
17. *The Mizar System.* – <http://mizar.uwb.edu.pl/>
18. *The Isar site.* – <http://isabelle.in.tum.de/Isar/>
19. *The Nqthm system.* – <http://www.cli.com/software/nqthm/>
20. *The ACL2 System.* – <http://www.cs.utexas.edu/users/moore/acl2/>

© А.В. Анисимов, Т. Джебелян, А.В. Лялецкий, Н. Попов,
2011

