УДК 518.74

З.М. Асельдеров
Национальный технический университет Украины
«Киевский политехнический институт»
К.П. Вершинин
Университет Paris XII – Val de Marne, г. Париж, Франция
А.И. Дегтярев
Королевский колледж, г. Лондон, Великобритания
А.В. Лялецкий, А.Ю. Паскевич
Киевский национальный университет им. Тараса Шевченко, Украина

Особенности обработки математических текстов в системе автоматизированной дедукции (САД)*

Исследовательская программа «Алгоритм очевидности» направлена на разработку системы автоматизации доказательств, предназначенной для доказательства теорем в контексте замкнутого математического текста. В настоящее время построена первая программная реализация такой системы под названием «Система автоматизированной дедукции» (САД). Основные свойства САД таковы: входной текст записывается при помощи формального языка, близкого к естественному; поиск вывода основан на секвенциальных исчислениях, формализующих «естественный стиль рассуждений», такой, как применение определений и вспомогательных утверждений. Эти исчисления также позволяют отделить дедукцию от решения уравнений, что дает подход к интеграции логических рассуждений с символьными вычислениями.

Введение

В работе [1] В.М. Глушков предложил программу исследований по автоматизации доказательств теорем, получившей название «Алгоритм очевидности», или АО. Основной целью АО является оказание помощи математику в обработке формализованных математических текстов, а именно: в построении и автоматизированной проверке длинных, но в некотором смысле «очевидных» доказательств. По существу, В.М. Глушков предложил одновременно вести исследования в следующих направлениях: разработка формализованных языков для представления математических текстов в наиболее удобном для пользователя виде; формализация и эволюционное развитие понятия шага машинного доказательства; создание информационной среды АО, влияющей на очевидность шага доказательства; построение средств поиска доказательств, предусматривающих участие человека.

* Авторы благодарят INTAS за поддержку данных исследований в рамах проекта INTAS 2000-447.

Данная работа отражает характерные особенности и текущее состояние дел по реализации программы АО в виде системы автоматизированной дедукции, прототипом для названия (САД) и создания которой послужила работа [2]. Заметим, что система САД, в конечном счете, ориентирована на решение задачи накопления математических знаний и их эффективного использования при обработке замкнутых математических текстов, записанных на формальном языке, максимально приближенном к языку естественных математических публикаций.

Языковая компонента САД

К языкам, используемым в системе САД для записи исходной математической информации, были предъявлены следующие требования. Такой язык должен обладать формальным синтаксисом и семантикой. Он должен позволять формулировку аксиом теории, теорем и доказательств, а также определений, чтобы обеспечить самодостаточность текста. Соответственно, тезаурус языка должен быть отделен от грамматики языка и быть пополняемым. С другой стороны, язык должен быть близок к естественному языку математических публикаций. Этого требуют соображения удобства работы пользователя при написании текста и в интерактивном режиме. Кроме того, это позволяет ставить перед системой задачи, не связанные прямо с поиском вывода.

Для заданного текста, в зависимости от поставленной задачи, язык должен предоставлять ту или иную исходную среду для работы системы. В этой среде проходит поиск доказательства, поиск вспомогательных утверждений и т.д.

Последним представителем семейства языков, разработанным в рамках работ по САД, является ForTheL (FORmal THEory Language) [3], имитирующий конструкции английского языка.

ForTheL-текст понимается как совокупность фраз, наделенная некоторой структурой, которая обеспечивает определенные «правила навигации» в тексте. Поэтому представляется разумным выделить две компоненты языка, в некотором смысле ортогональные. Первая есть язык ForTheL-фраз, вторая определяет построение текста в целом – группирование фраз, именование разделов, правила видимости переменных.

Синтаксис и семантика ForTheL-текста

Синтаксически ForTheL-текст представляет собой совокупность разделов, ограниченных лексемами-разделителями. Каждый раздел, в свою очередь, может содержать разделы более низкого уровня или фразы. Некоторые разделы – теоремы, доказательства, определения – играют особую роль в языке.

Каждая фраза является либо утверждением, либо предположением, если в начале ее стоит слово «let» или «assume».

Семантика ForTheL-текста определяется задачей, которая формулируется пользователем. Вот примеры таких задач: «проверить корректность текста», «проверить корректность утверждения в тексте», «построить контрпример к утверждению в тексте», «определить тематику текста», «найти, что говорится в тексте о ...» и так далее.

Система САД, получив на вход математический текст, написанный на языке ForTheL, переводит его в совокупность формул первого порядка с сохранением структуры разделов и сигнатуры исходного текста (этот промежуточный язык называется ForTheL1). На основании полученного текста строится среда доказательства для процедуры поиска логического вывода.

Язык ForTheL-фраз содержит язык первого порядка как собственное подмножество. Сверх того, ему присущи две ключевые особенности: словесная форма записи конструкций языка и так называемые понятия, являющиеся аналогами групп существительного.

Словесная форма записи позволяет строить ForTheL-фразы, как фразы естественного языка – с использованием подлежащих, сказуемых, дополнений и определений. ForTheL также предоставляет краткий символьный способ записи, принятый в математической логике.

Логические связки, наряду с несколькими предикатами и понятиями, предопределены как синтаксические единицы ForTheL. Все прочие синтаксические единицы должны быть введены при помощи определений.

Предикаты выражаются глаголами либо прилагательными, функции и понятия – существительными. Так, утверждение «|A| > n» может быть записано как «power of A is greater than n». Далее, говоря о словесных формах, мы будем употреблять термин «шаблон» вместо «функциональный символ» или «предикатный символ».

Понятие можно рассматривать как аналог терма языка первого порядка, однако его значением является не некоторый единичный объект, а (параметризованная) совокупность объектов. Пример понятия: «relation on X», где переменная X определена как некоторое множество (еще одно понятие языка ForTheL).

Каждому шаблону понятия арности n сопоставляется характеристический предикатный символ с арностью n+1. Соответствующее отношение выражает принадлежность объекта классу, описанному понятием.

Уточнение понятия, т.е. сужение задаваемого им класса, осуществляется с помощью специальных модификаторов – атрибутов и пост-условий. Один тип атрибутов накладывает ограничение на элемент класса, например «empty set» или «number dividing N». Другой тип атрибутов накладывает ограничение на значение некоторой унарной функции, заданной на классе, например, «set of power K» или «set of finite power». Легко заметить, что атрибуты являются производными от шаблонов предикатов. Постусловие есть обыкновенное утверждение, соединенное с шаблоном понятия посредством связки «such that».

Будучи предваренным кванторным словом «every» или «some», понятие образует терм, который может быть аргументом другого понятия, функции или предиката. Типичный пример простой ForTheL-фразы: «Every simple number greater than 2 is odd». Кванторы в тексте также могут быть ограничены объемом некоторого понятия: «For every number N there exists a successor of N».

Определения В ForTheL предопределены шаблоны понятий «class» и «set» и шаблоны предикатов «is equal to», «belongs to» и «is a». Все прочие шаблоны и символы определяются непосредственно в ForTheL-тексте. Заметим, что символы характеристических предикатов понятий позволяют элиминировать предикат «is a» при трансляции.

Дефиниции новых шаблонов составляют особый вид предложений ForTheL. Не вдаваясь в детали, достаточно сказать, что шаблоны предикатов определяются посредством указания критерия истинности, шаблоны функций – путем приравнивания функции некоторому терму либо через указание критерия равенства, шаблоны понятий – через уточнение некоторого понятия либо через указание характеристического отношения. ForTheL допускает и рекурсивные определения.

Пример forthel-текста

Здесь приводится корректный замкнутый ForTheL-текст, содержащий простую теорему о пустом множестве.

Definition 1. A is a subset of B iff every element of the set _ A belong to the set _ B.
Definition 2. A is empty iff set _ A has no elements.
Lemma 1. Any subset of any set is a set.
Lemma 2. There exists an empty set.
Theorem 1. Empty set is a subset of all sets.

Дедуктивная компонента САД

Как говорилось ранее, дедуктивная процедура САД работает в среде так называемого ForTheL1-текста, которое фактически может интерпретироваться как множество формул языка первого порядка. Эта процедура основана на исчислении секвенциального типа для классической логики первого порядка и отражает подходы, изученные [4] и [5].

Такой выбор был обусловлен тем, что секвенциальный подход является более близким к естественным способам рассуждений, чем резолюционные методы. В частности, нет необходимости представлять задачу множеством дизъюнктов со сколемовскими функциональными символами – мы можем эффективно работать непосредственно с исходными формулами и не обогащать сигнатуру новыми символами. Это облегчает понимание человеком хода доказательства при интерактивном режиме построения доказательства и позволяет установить дружественный диалог между пользователем и системой в процессе поиска доказательства. Важной чертой секвенциальных исчислений, разработанных в рамках исследований по САД, является то, что они дают возможность реализовать процедуры поиска вывода, сравнимые по эффективности с процедурами резолюционного типа.

Исчисление GD (Goal-Driven calculus), используемое в САД в настоящее время [5], имеет следующие характерные особенности:

1. Все необходимые «логические» преобразования проводятся только относительно выделенной «целевой» формулы. Каждый шаг сводит доказательство текущей цели к доказательству некоторой совокупности новых подцелей. Это свойство исчисления названо «целеуправляемостью».
2. Множество посылок на протяжении всего поиска доказательства остается неизменным. Таким образом, вывод в этом исчислении может быть представлен как дерево целей. На практике имеется возможность еще более упростить поиск вывода, ограничившись только литерными целями.

3. Исчисление использует технику отложенных «вычислений», при которой решение уравнений отделено от дедуктивного процесса. В дереве вывода накапливается система подстановочных уравнений, решение которой может быть отложено до произвольно выбранного момента.
4. Используя понятие допустимой подстановки, формулы с кванторами могут обрабатываться так же эффективно, как и в случае предварительной сколемизации, однако эта обработка проводится без обогащения исходной сигнатуры. Это позволяет использовать для систем уравнений любую подходящую технику нахождения их решений, в том числе и хорошо зарекомендовавшие себя методы внешних (по отношению к САД) систем компьютерной алгебры.
5. В исчисление могут быть включены специальные правила обработки определений и вспомогательных утверждений из входного текста [4].

Эффективность исчислений типа GD (по отношению к исчислениям генцено-кангеровского типов) достигается за счет того, что применение их правила вывода GD происходит только в «случае необходимости», что значительно сужает перебор при выборе очередного шага. С одной стороны, это заслуга стратегии целеуправляемости, которая «фокусирует» внимание на доказательстве единственной целевой формулы.

Другой важной чертой таких исчислений является использование в них оригинального понятия допустимой подстановки, которое не требует, чтобы терм, который должен подставляться вместо связанной переменной, был бы свободным для этой переменной, а принципиальная возможность реализации этого требования является следствием выполнения для подстановки определенных условий, вытекающих из определения допустимой подстановки (например, [6]).

Заметим, что вычисление подстановки, которая «замыкает» дерево вывода, проводится путем решения накапливаемой системы уравнений, причем поиск решения системы уравнений может начаться в любой момент времени (определяемый, быть может, пользователем). Полученная в результате подстановка затем проверяется на допустимость и, в случае необходимости, процесс поиска продолжается.

Проверка допустимости подстановок основывается на простой комбинаторной процедуре. Это позволяет избежать перебора, вызываемого в обычных генценовских исчислениях разными порядками снятия кванторов, и получить процедуру, которая по эффективности сопоставима с методами, использующими предварительную сколемизацию.

Основным объектом исчисления GD является так называемая e-секвенция. Она представляет собой расширение традиционного понятия секвенции путем добавления к секвенциям систем подстановочных уравнений, нахождение «допустимых» решений которых обеспечивает корректность предыдущих преобразований.

На основе порожденного транслятором ForTheL1-текста, с выделенным в нем разделом «теорема», формируется исходная e-секвенция, где формула первого порядка, выражающая утверждение этого раздела, становится целью, а ее «логические предшественники» в тексте (аксиомы, определения, леммы) – посылками. Понятие «логического предшественника» определяется формальным образом, исходя из структуры разделов ForTheL1-текста. Можно доказать следующую теорему.

В заданном ForTheL1-тексте теорема является логическим следствием своих логических предшественников тогда и только тогда, когда существует дерево вывода в исчислении GD, где корнем является соответствующая исходная е-секвенция, листьями – аксиомы исчисления, и существует допустимая подстановка, которая является решением совокупности всех подстановочных уравнений в дереве вывода.

Текущее состояние

Практической задачей системы САД является осуществление следующей цепочки преобразований (например, [5]):

1. Переход от обычного математического текста к ForTheL-тексту, созданному при участии человека и содержащему выделенное в нем утверждение, требующее доказательства;
2. Порождение транслятором замкнутого ForTheL1-текста по исходному ForTheL-тексту;
3. Построение среды доказательства, учитывающей методы логического вывода и способы обработки равенств (решения уравнений), и, в случае успеха,

Получение отредактированного доказательства, написанного на языке, близком к естественному математическому языку.

К данному моменту времени в рамках системы САД реализованы транслятор языка ForTheL в ForTheL1 и программа построения среды доказательства и автоматического поиска доказательства в рамках классической логики первого порядка на базе секвенциального формализма.

Модуль редактирования доказательства разрабатывается в настоящее время.

Для реализации всех программных модулей был использован язык программирования Си.

Транслятор ForTheL – ForTheL1 выполняет два основных задания: грамматический анализ входного ForTheL-текста одновременно с порождением «перевода» и поддержку внутреннего словаря определяемых шаблонов, в соответствии с которым и происходит трансляция.

Как уже было сказано, текущая версия языка имитирует английскую грамматику, однако исследуется возможность версий на основе русской и английской грамматик. В частности, это позволит осуществлять автоматический перевод ForTheL-текстов на различные языки при наличии словаря соответствий между определяемыми шаблонами.

Чтобы распознавать лексемы-слова в их различных формах (множественное число, причастие настоящего времени и т.д.) транслятор пытается определить для каждого существительного или глагола, вводимого определением, его неизменяемую основу и максимально возможное число букв в суффиксе.

Синтаксический анализатор генерируется программой Bison, которая традиционно используется для такого рода задач.

Процедура поиска логического вывода в составе САД (называемая далее «прувер») состоит из так называемой «главной последовательности» и нескольких вспомогательных модулей, отвечающих за обработку структур данных. Главная

последовательность включает три последовательно выполняемых этапа: обработка входных данных, создание дедуктивной среды, доказательство заданной теоремы.

На первом этапе прувер переводит формулы входного ForTheL1-текста в конъюнктивную нормальную форму и строит структуры данных, представляющие рассматриваемую проблему.

На втором этапе определяются пары контрарных литер, вычисляются соответствующие «базовые» подстановки и устанавливаются избыточные формулы и шаги в дальнейшем доказательстве. Заметим еще раз, что поскольку исчисление, на котором основан прувер, не меняет множество посылок в процессе вывода, то достаточно поддерживать дерево литерных целей вместо дерева секвенций. Это позволяет значительно экономить ресурсы.

На третьем этапе прувер исследует поисковое пространство, используя традиционную процедуру поиска в глубину с ограничением по глубине и бэктрекингом. Для повышения эффективности используются определенные стратегии, в частности, порожденные уравнения решаются только тогда, когда это необходимо.

Наиболее важным вспомогательным модулем является блок обработки систем подстановочных уравнений, который действует как посредник между прувером и «внешним» модулем решения уравнений. Получив от «внешнего» модуля решение данной системы, этот модуль проверяет допустимость порожденной подстановки и, в случае надобности, формулирует дополнительные уравнения.

В настоящий момент в качестве «решателя уравнений» действует модуль поиска наиболее общего унификатора. Однако, в зависимости от поставленной проблемы, прувер может использовать любой подходящий «решатель». Это становится возможным благодаря отказу от предварительной сколемизации и сохранению исходной сигнатуры.

Недавно была разработана и «встроена» прототипная версия процедуры элиминации равенства при помощи «уплощения» (flattening) термов, что позволяет говорить о том, что САД умеет осуществлять дедукцию вместе с непосредственной обработкой равенств в рамках классической логики первого порядка с равенством.

Заключение

Описанные выше особенности системы САД указывают на то, что САД спроектирована и реализована с учетом современных достижений в области построения компьютерных математических служб. В этой связи отметим следующее.

Базисом языка ForTheL являются фундаментальные логические и теоретико-множественные отношения, поэтому он пригоден для представления любого (не только математического) знания, если последнее формализуемо в терминах классической логики. ForTheL-текст может быть создан как человеком, так и компьютерной программой, а затем передан системе поиска доказательств или другой программе непосредственно или через сеть.

Что касается дедуктивной компоненты САД, то имеющиеся теоретические разработки и реализованная на их основе процедура поиска доказательства могут служить хорошим базисом для дальнейших усовершенствований дедуктивной техники в стиле «Алгоритма очевидности» в направлении повышения эффективности и построения средств интерактивной обработки математического текста.

Следует особо отметить заложенную в САД возможность сочетать дедукцию с аналитическими преобразованиями, выполняемыми внешними по отношению к САД системами, которая обеспечивается специальной техникой обработки математического текста и свойствами дедуктивного формализма, позволяющими сохранять сигнатуру исходной задачи и обрабатывать равенства независимо от поиска вывода.

Авторы надеются, что результаты, полученные в рамках исследований по САД, окажутся полезными при решении таких задач, как распределенный автоматический поиск доказательства, проверка корректности формализованного математического текста, извлечение сведений из математического текста, построение баз формального математического знания и удаленное обучение математическим дисциплинам.

Литература

1. Глушков В.М. Некоторые проблемы теории автоматов и искусственного интеллекта // Кибернетика. – 1970. – Т. 2. – С. 3-13.
2. Глушков В.М. Система автоматизации доказательств (САД) // Автоматизация обработки математических текстов. – Киев: ИК АН УССР, 1980. – С. 3-30.
3. Vershinin K., Paskevich A. ForTheL – the language of formal theories // IJ Information Theories and Applications. – 2000. – Vol. 3-7. – P. 121-127.
4. Degtyarev A., Lyaletski A., Morokhovets M. Evidence Algorithm and Sequent Logical Inference Search // Lecture Notes in Artificial Intelligence. – 1999. – Vol. 1705. – P. 44-61.
5. Lyaletski A., Verchinine K., Degtyarev A., Paskevich A. SAD, a System for Automated Deduction: a Current State // Proc. of International Workshop Automath'2002. – Edinburgh (Great Britain). – 2002.
6. Lyaletski A., Gentzen calculi and admissible substitutions // Actes preliminaries, du Simposium Franco-Sovetique «Informatika – 91». – Grenoble (France). – P. 99-111.

The goal of a research program «Evidence Algorithm» is a development of an open system of automated proving that is able to accumulate mathematical knowledge and to prove theorems in a context of a self-contained mathematical text. By now, the first version of such a system called a System for Automated Deduction, SAD, is implemented in software. The system SAD possesses the following main features: mathematical texts are formalized using a specific formal language that is close to a natural language of mathematical publications; a proof search is based on special sequent-type calculi formalizing «natural reasoning style», such as application of definitions and auxiliary propositions. These calculi also admit a separation of equality handling from deduction that gives an opportunity to integrate logical reasoning with symbolic calculation.

Статья поступила в редакцию 05.07.02.