Transactions of the Institute for Informatics and Automation Problems of NAS RA.
Collected reports of participants of JAF-XX.

ЭВИДЕНЦИАЛЬНАЯ ПАРАДИГМА И ОБРАБОТКА МАТЕМАТИЧЕСКИХ ТЕКСТОВ В СИСТЕМЕ САД

А. В. Лялецкий и П. П. Процик

В данной работе описывается подход к автоматизации рассуждений, получивший название эвиденциальной парадигмы и имеющий вид Системы Автоматизации Дедукции, САД, реализованной в настоящее время на платформах Linux и Windows. Одной из особенностей системы САД является то, что математическая обработка компьютерных знаний осуществляется в ней на базе целостного текста, записанного на формальном языке первого порядка, близком к языкам естественных математических публикаций. К другой же относится то, что для целей проведения дедуктивных преобразований САД использует специальный секвенциальный формализм, который дает возможность строить эффективные методы поиска вывода в сигнатуре исходной теории. Это позволяет использовать аналоги таких приемов поиска доказательства, как применение определений и вспомогательных утверждений, а также легко инкорпорировать в САД различные компьютерные службы, например, пруверы и системы компьютерной алгебры.

Abstract. In this paper, an approach to automated reasoning called the evidential paradigm and presented by the System for Automated Deduction, SAD, which is recently implemented on the platforms Linux and Windows is described. One of peculiarities of SAD is that it makes the mathematical processing of computer knowledge on the base of a self-contained text being written in a first-order formal language closely connected with languages of usual mathematical publications. Its other peculiarity is that a special sequent formalism is used for the purpose of deductive transformations, which gives a possibility for the construction of efficient technique for inference search in the signature of an initial theory. This permits to use analogues of such natural methods as applications of definitions and auxiliary propositions as well as to incorporate various computer services - for example, provers and computer algebra systems - in the SAD.

§1. ВВЕДЕНИЕ

Область автоматизации дедукции представляет собой удачливое сочетание теории и практики, хотя иногда оно оказывается искаженно толкуемым. С одной стороны, специалисты в области автоматизации дедукции считают центральным вопросом эффективное применение теоретических исследований по логике, но концентрация внимания на вопросах только реализации часто заставляет их оставлять в стороне те фундаментальные вопросы, исследование которых требуется для решения стоящей перед ними задачи по автоматизации рассуждений. Специалисты по логике, с другой

стороны, обращают основное внимание на внутреннее теоретическое развитие своих областей исследований, делая ударение на решении трудных проблем: вопросы же реализации трактуются ими как „чисто прагматические", „инженерные" задачи. Следовательно, возникает необходимость найти разумный баланс между этими двумя точками зрения, одновременно развивая те теоретические и прикладные стороны теории доказательств, математической логики и лингвистики, которые в текущий момент времени могут быть наиболее эффективно реализованы на компьютерах. Вот почему оказывается чрезвычайно важным строить и развивать системы интеллектуальной помощи человеку, которые способны как понимать тексты на формальном языке, максимально приближенном к языку обычных публикаций, так и делать умозаключения о компьютерных знаниях, представленных на таком языке. Краткое описание исследований в этом направлении, проводимых в рамках работ по так называемой эвиденциальной парадигме, как раз и представлено в данной работе.

В настоящее время быстрое развитие автоматизации рассуждений делает возможным предвидеть ощутимую помощь в автоматизации поиска доказательств, которая может оказаться чрезвычайно полезной как при решении проблем, связанных как с самой математикой, так и с различными прикладными областями. Например, распределенное и/или удаленное обучение различным математическим дисциплинам, проверка корректности математических текстов, и построение баз знаний для формализованных теорий уже начинают привлекать возрастающее внимание при проведении математических исследований. Существуют и дополнительные практические соображения, учитываемые при проектировании систем, которые поддерживают формальное развитие математики. В свою очередь, для целей верификации формальных текстов разного рода требуются различные математические библиотеки: библиотеки основных фактов из арифметики и анализа - для проверки электрических цепей и электронных схем, сведения по теории чисел и комбинаторике - для верификации криптографических протоколов, разнообразные математические библиотеки - для проведения анализа научных данных. Следовательно, корректное понимание ЭВМ формальных текстов и автоматизация дедукции являются необходимыми элементами поддержки, хранения, обмена и обработки

математической информации и других „компьютеризуемых" знаний.

Один из возможных подходов в этом направлении был предложен академиком В.М. Глушковым в конце 1960-х - начале 1970-х годов в рамках работ по так называемой программе Алгоритм Очевидности (Evidence Algorithm, EA) [1]. В.М. Глушков предложил проводить одновременно исследования по формальным языкам естественного типа для представления математических текстов в форме, наиболее удобной для пользователя; по построению „эвиденциальных" программ, т.е. по формализации и эволюционному развитию понятия машинного шага доказательства; по созданию „эвиденциальной" информационной среды, имеющей влияние на текущее понятие компьютерного шага доказательства; по интерактивным средствам поддержки пользователем процесса доказательства (см. сайт www.ea.unicyb.kiev.ua).

В данной работе описывается современное состояние работ по Алгоритму Очевидности, которое базируется на интеграции вычислительной, символьной и дедуктивной парадигм - современных хорошо известных подходов к обработке математических текстов на ЭВМ, что позволяет говорить об *эвиденциальной парадигме* [2], которая в настоящее имеет вид Системы Автоматизации Дедукции, САД (System for Automated Deduction, SAD) (http:// ea.unicyb.kiev.ua/sad.en.html), которая реализована на двух платформах: Linux и Windows.

§2. СОВРЕМЕННЫЕ ТЕНДЕНЦИИ ПОСТРОЕНИЯ СИСТЕМ АВТОМАТИЗАЦИИ РАССУЖДЕНИЙ

Характеризацию современного подхода к автоматизации рассуждений в стиле Алгоритма Очевидности удобнее всего проводить с учетом тенденций в следующих трех основных направлениях исследований по автоматизации рассуждений: стилю формализации, стилю доказательства, который требуется для них, и степени „грануляции" (подробности) доказательств.

Стиль формализации. На стиль формализации оказывают влияние предметная область, предварительные сведения, вид определений (являются ли они компьютеризируемыми), способ рассуждения (является ли он конструктивным, строго типизируемым или основанном на том или ином исчислении) и т.д. Очевидно, что на стиль формализации оказывают влияние, прежде всего, используемая базовая логика

и фундаментальные теории, вовлекаемые в формализацию.

В настоящее время существует два основных стиля формализации: первый ориентирован на применение логик высших порядков (с теорией типов в качестве основы), а второй - на логике первого порядка (с теорией множеств в качестве основы).

Типы применяются в большинстве хорошо известных ассистирующих систем, таких, как Isabelle/HOL [3], Coq [4], Omega [5], PVS [6], HOL [7], Automath [8], Theorema [9], Lambda-Clam [10] и других. Предпочтение применению теоретико-типового подхода отдается в случаях обращения к индуктивно-определяемым областям и рекурсивным определениям, и он хорошо подходит для проведения формализации программистских и инженерных концепций. Однако, этот подход не является идеальным для формализации традиционной математики [11], хотя в большей части упомянутых систем используются разнообразные наборы чисто математических теорий.

В качестве примера второго стиля можно упомянуть хорошо известную систему Mizar [12], которая использует логику первого порядка и теорию множеств Тарского-Гротендика. Такая ориентация хорошо соответствует традиционному способу построения математики, и существует больший набор математических утверждений, проверенных системой Mizar.

Система САД [13,14] не адаптирована к какой-нибудь теории множеств (или другой фундаментальной теории), дающей общую основу для формализации. Вместо этого, при решении той или иной задачи предлагается задать необходимый набор предварительных сведений, выражая базовые концепции на специальном формальном языке, транслируемым в определенную разновидность языка первого порядка.

Стиль доказательства. Другим важным свойством систем оказания помощи в автоматизации рассуждений является вид их входных данных. Так называемые интерактивные системы (первой тип) чаще всего являются тактико-управляемыми, что означает, 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 обработки рассматриваемой цели некоторого внешнего (по отношению к системе) прувера. К системам такого типа относятся Isabelle [3], Coq [4], Omega [5], PVS [6], HOL [7] и ряд других систем. Работа с системой такого рода оказывается легкой в случае, когда она обеспечивает изящный набор мощных тактик, которых достаточно для того, чтобы решить поставленную задачу.

Второй тип систем ориентирован на декларативное представление решаемой задачи, 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. Наиболее типичными представителями этого подхода являются системы Mizar [12] и SAD [13,14]. Система Isabelle [3], расширенная средствами Isar [15] (и включающая язык для записи структурированных доказательств, имитирующий язык обычных математических доказательств), также может быть отнесена к системам, управляемых доказательствами.

Различие между двумя типами входных данных не является очень существенным. Если можно построить доказательства теорем тактико-управляемой системой, использующей промежуточный ввод целей и автоматическое закрытие подцелей, то такая система может рассматриваться как управляемая доказательством. В обратную сторону, если шаги вывода, воспринимаемого некоторой системой, управляемой доказательством, снабжены детальными подсказками о способах верификации, то такая система может рассматриваться как тактико-управляемая, для которой подсказки играют роль тактик.

Степень „грануляции" доказательства. 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[1]. Первые принимают на вход только шаги доказательства, имеющие вид применений правил вывода, и, следовательно, должны быть детализированы с точностью до

[1]Это деление условное и определяется текущим понятием степени подробности („грануляции") доказательства.

правил вывода. Mizar [12] служит примером такой системы, хотя сам набор ее правил вывода достаточно велик.

Системы, которые называются системами поиска доказательств, содержат методы поиска вывода и/или планировщики доказательств и пытаются восполнить „недостающие" места доказательства. Системы SAD [13], Nqthm [16], ACL2 [17], и Theorema [9] являются представителями систем такого рода. Тактико-управляемые системы типа Isabelle [3] and Coq [4] 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 поиска доказательств. Однако некоторые эксперименты с Isabelle и Coq показывают, что следующее предложение может оказаться критическим для них: При всякой попытке „сразу же" доказать теорему, „диалог" о построении доказательства сразу становится сложным, сильно ветвящимся и тяжело отслеживаемым без предварительного выделения лемм, без применения специальных тактик и без использования существующих библиотек.

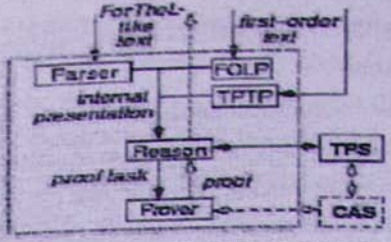
§3. СИСТЕМА САД: ТЕКУЩЕЕ СОСТОЯНИЕ

Прежде всего, заметим, что на настоящем этапе система САД предназначена для выполнения трех основных задач [13,14]:

- выполнения поиска вывода в классической логике первого порядка, основанного на оригинальном секвенциальном формализме [18,19], восходящим к работам [20,21];
- доказательства теорем в среде замкнутого[2] математического текста [22], записанного на формальном языке ForTheL [23,24], который максимально приближен к языкам естественных (математических) публикаций;
- верификации замкнутых ForTheL-текстов, содержащих как утверждения, так и их доказательства [25,26].

Относительно описанного „трехмерного пространства" САД имеет следующую архитектуру:

[2] Под замкнутым математическим текстом понимается текст, который, кроме утверждений, которые требуется доказать/верифицировать, содержит всю необходимую информацию для проведения доказательства/верификации (т.е. допущения, определения, вспомогательные утверждения).



Кратко опишем приведенную трехуровневую организацию системы САД.

На *первом уровне* модуль Parser делает синтаксический анализ входного текста, его структуры и логического содержания, закодированного в ForTheL-предложениях. После этого он транслирует его во входное представление (internal presentation). Результатом трансляции оказывается простое по своей организации дерево предложений, которое после „линеаризации" определяет последовательность целевых утверждений, которые должны быть доказаны с использованием своих логических предшественников. Модуль FOLP осуществляет синтаксический контроль входных данных в случае, когда системой САД решается задача установления выводимости исходной секвенции. TPTP обозначает библиотеку Thousands Problems for Theorem Proving [27], которая размещена в Internet и которая используется интерфейсом системы САД для получения „тестовых" заданий по установлению выводимости.

На *втором уровне* целевые утверждения последовательно обрабатываются модулем Reason, выполняющем сведение задачи, подлежащей обработке, к ряду подзадач, передаваемых прувер у. Модуль Reason работает в диалоге с прувером: он может расщепить основную цель на ряд более простых подцелей (proof task) для прувера или предложить альтернативную цель в случае, когда оказывается неудачной текущая попытка провести доказательство. В данное время набор программных средств этого модуля не является достаточно богатым и содержит совокупность приемов упрощения на пропозициональном уровне. Также отметим, что модуль Reason становится излишним, когда решается задача доказательства теоремы и, в частности, задача установления логической выводимости.

Поиск логического вывода делается модулем Prover (прувером) на третьем уровне. Внутренний прувер системы САД базируется на специальном секвенциальном исчислении для классической логики первого порядка. Это исчисление использует

оригинальное понятие допустимой подстановки, которое впервые было предложено в [20] и позволяет проводить эффективный поиск вывода в сигнатуре исходной теории (без сколемизации). Помимо этого, оно дает возможность накапливать системы уравнений (равенств), решение которых в настоящее время сводится к нахождению наиболее общего унификатора [18,22]. В последующих версиях системы САД системы уравнений планируется передавать специальным решателям, CAS (например, системам компьютерной алгебры), при появлении такой необходимости. Также заметим, что САД была реализована таким образом, что вместо своего внутреннего прувера она может подключать к себе пруверы (TPS), внешние по отношению к ней, такие, как Otter [28], SPASS [29], и Vampire [30].

На современном этапе главные особенности темы САД касаются ее лингвистических и дедуктивных средств, которые ниже описываются в общих чертах (за более подробным описанием языка ForTheL и логического аппарата САД можно обратиться к [18,19,23,24,26,31]).

§4. ЛИНГВИСТИЧЕСКИЕ ОСОБЕННОСТИ СИСТЕМЫ САД

В самом начале работы с системой САД пользователь должен записать входной текст на языке ForTheL. Этот текст должен содержать анализируемое утверждение, его предпосылки, необходимые факты и доказательства (при работе в режима верификации текста), а также некоторую информацию, считающуюся релевантной относительно рассматриваемого утверждения (утверждений).

ForTheL проектировался так, чтобы он был близок к языкам реальных математических доказательств. В частности, грамматика ForTheL следует общепринятой грамматике английского языка (конечно, это касается только очень небольшого его фрагмента, допускающего формализацию в ForTheL).

Преимущество такого подхода многократно обсуждалось ранее (например, см. архивы проекта QED [32]). Можно указать, по крайней мере, две причины для использования „естественного" стиля написания текстов. Первая заключается в том, что достигается дружественный интерфейс с пользователем. Вторая состоит в том, что разработчики системы САД придерживаются тезиса о том, что естественноподобные тексты содержат информацию в таком виде, который является более релевантным для оптимальной обработки математических (и не только их, а

любых формализуемых) текстов, чем в случае использования традиционных языков классической логики. Например, в естественных текстах определения отделены от аксиом, лемм и теорем; существительные, глаголы и прилагательные несут разную смысловую нагрузку, и т.п. Именно такие особенности естественных языков и принимались в расчет при разработке ForTheL.

Некоторое представление о стиле записи текстов на ForTheL можно получить из следующего примера, содержащего как формулировку обрабатываемой теоремы в среде замкнутого текста, так и запись ее доказательства.

```
Definition Def Subset. Let S be a set.
    A subset of S is a set with no elements in S.
Definition Def Empty. Let S be a set.
    S is empty iff S has no elements.
Axiom Ax Empty. There exists an empty set.
Proposition. Let S be a set.
    S is a subset of every set iff S is empty.
  Proof.
    First assume S is a subset of every set.
      Then S is empty.
      Indeed
        Let z be in S and E be an empty set.
        Then z is an element of E.
        We have a contradiction.
      end.
      end.
    Now assume S is empty.
      Let T be a set.
      Every element of S is in T (by Def Empty).
      Hence S is a subset of T (by Def Subset).
    end.
  qed.
```

§5. ДЕДУКТИВНЫЕ ОСОБЕННОСТИ СИСТЕМЫ САД

Одним из основных положений эвиденциальной парадигмы является понятие эволюционно развивающегося шага очевидности, которое определяется теми имеющимися в распоряжении дедуктивными средствами, которые компьютер может использовать для доказательства утверждений. На текущее время развит специальный секвенциальный формализм для логики первого порядка. Он позволяет не

только строить различные компьютерно-ориентированные секвенциальные исчисления для классической логики (см., например, [19,20]), но и дает подход к построению достаточно эффективных секвенциальных исчислений для интуиционистской логики [33,34]. Другой особенностью этого формализма [35] является то, что он позволяет единообразным способом выразить такие хорошо известные методы поиска доказательства/опровержения, как кангеровские исчисления, метод резолюции А.Дж. Робинсона и обратный метод С.Ю. Маслова.

Разработка секвенциального формализма проводилась относительно выполнения таких требований:

- синтаксическая форма входной задачи должна быть сохранена;

- поиск вывода должен вестись, при необходимости, в сигнатуре исходной теории (что требует развития эффективной специальной техники работы с кванторами);

- дедуктивный процесс должен быть отделен от обработки равенств (решения уравнений) с целью привлечения к поиску решения, при необходимости, специальных решателей и, в частности, систем компьютерной алгебры;

- эвристические методы (и, в первую очередь, специальные приемы применения определений и вспомогательных утверждений) должны быть легко инкорпорируемыми в дедуктивные средства;

- все шаги доказательства должны обеспечивать, как минимум, корректность дедуктивных преобразования и, как максимум, их полноту;

- должны быть предусмотрены интерактивные режимы поиска доказательства.

В реализованной версии системы САД указанный выше секвенциальный формализм нашел свое выражение в виде имплементации секвенциального исчисления из [14], что, вместе с модулем Reason, позволяет провести как верификацию текста, приведенного выше, так и доказательство утверждения из этого текста, не прибегая к написанному для него доказательству.

§6. ИНФОРМАЦИОННАЯ СРЕДА

Как отмечалось выше, САД должна работать в информационной среде (ИС) декларативных математических знаний. В настоящее время различается несколько видов ИС, которые могут быть использованы для представления и обработки математических знаний в системе САД.

Внешняя ИС - распределенная, виртуальная, существующая в виде компьютерных математических служб. Доступ к ней осуществляется, например, через Internet. Она служит для информационного поиска ответов на вопросы во внешнем мире, используя уже существующие средства извлечения математических знаний, а также для осуществления Internet-доступа к существующим алгоритмическим службам, которые могут оказаться полезными для (алгоритмического) решения поставленной задачи. Примером такого информационного доступа к внешней среде в существующей версии системы САД может служить библиотека TPTP.

Резидентная ИС должна содержать всю совокупность накапливаемых математических знаний о предметных областях, в которых работал и/или работает САД. Она представляет собой совокупность иерархически упорядоченных „слоев". Слои должны формироваться по ходу эволюционного развития САД и использоваться как для выдачи ответов (с разными уровнями подробности) на математические вопросы из „внешнего" мира, так и для создания различных сред доказательства САД, которые отражают текущие ее дедуктивные возможности относительно решаемой задачи.

Внутренняя ИС фактически должна представлять собой семантическую сеть с определенными на ней правилами вывода. Основная функция внутренней ИС - формирование и поддержка динамической среды доказательства для процедур поиска логического вывода, в частности, аппарата раскрытия определений и поиска вспомогательных утверждений и их применения. Информационной основой для построения внутренней ИС служит резидентная ИС. Для реализации внутренней ИС целесообразно использовать ситуации и операции над ними, которые были реализованы и апробированы в ранних версиях САД.

§7. ИНТЕРФЕЙСНЫЕ СРЕДСТВА

Разработка базового формализма и основе перечисленных выше принципов построения дедуктивных средств системы САД позволяет задавать методы поиска доказательства в рамках сигнатуры исходной теории. Это, в свою очередь, дает возможность генерировать достаточные утверждения и получать следствия в достаточно привычном для человека виде, что может послужить отправной точкой для создания гибких интерфейсных средств системы САД.

§8. ЗАКЛЮЧЕНИЕ

Система САД, разработанная на базе эвиденциальной парадигмы, может рассматриваться как дедуктивная „оболочка" для автоматизации рассуждений с естественным интерфейсом, которая „упрятывает" от пользователя „утомительные" технические детали и дает возможность решать многочисленные проблемы, применяя как тактики, учитывающие специфику проблемной области, так и эффективный поиск вывода в различных логиках.

Эта „оболочка" может служить основой для построения различных интеллектуальных систем для решения математических проблем и индустриальных задач. В частности, следующие области могут быть зафиксированы в качестве возможных приложений системы САД: компьютерная помощь в обучении математике и логике, удаленное компьютерное обучение математическим дисциплинам, автоматизированное доказательство теорем и верификация формализованных текстов, извлечение знаний из математических работ, конструирование баз знаний для формализованных теорий, верификация программного обеспечения и аппаратных средств, проверка протоколов на корректность.

В длительной перспективе, работы по реализации эвиденциальной парадигмы направлены на создание мощной инфраструктуры компьютерных знаний для образования и для проведения математических исследований.

ЛИТЕРАТУРА

1. Глушков В.М. Некоторые проблемы теории автоматов и искусственного интеллекта. Кибернетика, 1970, № 2, 3-13.

2. Lyaletski A. and Morokhovets M. Evidential paradigm: a current state. Program of International Conference "Mathematical Challenges of the 21st Century", University of California, Los Angeles, USA, 2000, 48.

3. Nipkow T., Paulson L.C., and Wenzel M. Isabelle/HOL: A proof assistant for higher-order logic. Vol. 2283 of Lecture Notes in Computer Science, Springer-Verlag, 2002. (The Isabelle system: http://www.cl.cam.ac.uk/ Research/HVG/Isabelle/).

4. Barras B., Boutin S., Cornes C., Courant J., Filliatre J., Gimonez E., Herbelin H., Huet G., Munoz C., Murthy C., Parent C., Paulin C., Saïbi A., and Werner B. The

Coq proof assistant reference manual - version v6.1. Tech. Rep. 0203, INRIA, 1997. (The Coq system: http://coq.inria.fr/).

5. Benzmuller C., Cheikhrouhou L., Fehrer D., Fiedler A., Huang X., Kerber M., Kohlhase M., Konrad K., Meier A., Melis E., Schaarschmidt W., Siekmann J.H., and Sorge V. Omega: Towards a mathematical assistant. W. McCune (Ed.), Automated Deduction - CADE-14: 14th International Conference on Automated Deduction. Vol. 1249 of Lecture Notes in Computer Science, Springer-Verlag, 1997, 252-255. (The Omega system: http: //www.ags.uni-sb.de/ autexier/omega/).

6. Owre S., Rushby J.M., and Shankar N. PVS: a prototype verification system, in: Kapur D. (Ed.), Automated Deduction - CADE-11: 11th International Conference on Automated Deduction. Vol. 607 of Lecture Notes in Computer Science, Springer-Verlag, 1992, 748-752. (The PVS system: http://pvs.csl.sri.com/).

7. Gordon J.C. and Melham T.F. Introduction to HOL: a theorem proving environment for higher order logic, Cambridge University Press, 1993. (The system HOL: http://www.cl.cam.ac.uk/Research/HVG/HOL/).

8. Nederpelt R.P. and Geuvers J.H. de Vrijer R.C. (Eds.), Selected Papers on Automath. Vol. 133 of Studies in Logic and the Foundations of Mathematics, North-Holland, 1994.

9. Buchberger B., Jebelean T., Kriftner F., Marin M., Tomuta E., and Vasaru D. A survey of the Theorema project. W. Kuchlin (Ed.), ISSAC'97 - Proc. International Symposium on Symbolic and Algebraic Computation, ACM Press, Maui, Hawaii, USA, 1997, 384-391. (The Theorema system: http://www.risc.uni-linz.ac.at/research/theorema/description/).

10. Richardson J., Smaill A., and Green I. System description: Proof planning in higher-order logic with Lambda-Clam. Automated Deduction - CADE-15: 15th International Conference on Automated Deduction, Vol. 1421 of Lecture Notes in Computer Science, Springer-Verlag, 1998, 129-133. (The Lambda-Clam System: http://dream.dai.ed.ac.uk/software/lambda-clam/)

11. Lamport L. Types considered harmful, DEC SRC internal note, Dec. 1992.

12. Trybulec A. and Blair H. Computer assisted reasoning with Mizar. Proc. of the 9th Intern. Joint Conference on Artificial Intelligence, 1985, 26-28.
(The Mizar System: http://mizar.uwb.edu.pl/).

13. Lyaletski A., Verchinine K., Degtyarev A., and Paskevich A. System for Automated Deduction (SAD): Linguistic and deductive peculiarities. Klopotek M.A., Wierzchon S.T., Michalewicz M. (Eds.), Intelligent Information Systems 2002, (IIS'2002 Symposium, Sopot, Poland, June 3?6, 2002), Advances in Soft Computing, Physica-Verlag, 2002, 413-422.
(The SAD system: http: //ea.unicyb.kiev.ua/sad.en.html)

14. Lyaletski A., Paskevich A., and Verchinine K. Theorem proving and proof verification in the system SAD. Asperti A., Bancerek G., Trybulec A. (Eds.), Mathematical Knowledge Management: Third International Conference, MKM 2004, Vol. 3119 of Lecture Notes in Computer Science, Springer-Verlag, 2004, 236-250.

15. Wenzel M.: Isar - a generic interpretative approach to readable formal proof documents, in: Theorem Proving in Higher Order Logics: 12th International Conference. TPHOLs'99, Vol. 1690 of Lecture Notes in Computer Science, Springer-Verlag, 1999, 167-184. (The Isar site: http:// isabelle.in.tum.de/Isar/).

16. Boyer R.S. and Yuan Yu. Automated correctness proofs of machine code programs for a commercial microprocessor. D. Kapur (editor), Automated Deduction — CADE-11, Vol. 607 of Lecture Notes in Computer Science, Springer-Verlag, 1992, 416-430. (The Ngth system. http://www.cli.com/ software/nqthm/)

17. Kaufmann M., Manolios P., and Moore J.S. Computer-aided reasoning: an approach. Kluwer Academic Publishers, June, 2000.
(The ACL2 System: http:// www.cs.utexas.edu/users/moore/acl2/).

18. Degtyarev A., Lyaletski A., and Morokhovets M. Evidence Algorithm and sequent logical inference search. Vol. 1 705 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 1999, 44-61.

19. Лялецкий А.В. Эвиденциальная парадигма: логический аспект. Кибернетика и системный анализ. 2003, № 5, 37-47.

20. Лялецкий А.В. Вариант теоремы Эрбрана для формул в пренексной форме. Кибернетика, 1981, № 1, 112-116.

21. Lyaletski A. Gentzen calculi and admissible substitutions // Actes preliminares du Simposium Franco-Sovietique „Informatika-91", Grenoble, France, 1991, 99-111.

22. Degtyarev A., Lyaletski A., and Morokhovets M. On the EA-Style Integrated Processing of Self-Contained Mathematical Texts. Symbolic Computation and Automated Reasoning (the book devoted to the CALCULEMUS-2000 Symposium: edited by M. Kerber and M. Kohlhase), A.K. Peters, Ltd, USA, 2001, 126-141.

23. Vershinin K. and Paskevich A. ForTheL - the language of formal theories, International Journal of Information Theories and Applications, 2000, 7 (3), 120-126.

24. Verchinine K. and Paskevich A.: ForTheL reference manual, Unpublished draft, hosted at http://ea.unicyb.kiev.ua/download/forthel.ps.gz, 2004.

25. Verchinine K., Degtyarev A., Lyaletski A., and Paskevich A. SAD, a System for Automated Deduction: a Current State. Proceedings of the Workshop on 35 Years of Automating Mathematics. - Heriot-Watt University, Edinburgh, Scotland. 10-13 April, 2002.

26. Lyaletski A., Verchinine K., and Paskevich A. On verification tools implemented in the System for Automated Deduction. Proceedings of the 2nd Workshop on Implementation Technology for Computational Logic Systems (ITCLS), Pisa, Italy, 2003, 25-36.

27. Sutcliffe G., Suttner C.B., and Yemenis T. The TPTP problem library. Automated Deduction // CADE-12, Proc. 12th International Conference on Automated Deduction. Volume 814 of Lecture Notes in Computer Science., Springer (1994) 252-266 (Also see http://tptp.org).

28. McCune W. Otter 3.0 reference manual and guide. Tech. Report ANL-94/6, Argonne National Laboratory, Argonne, USA, 1994.
(The Otter automated deduction system: http://www.mcs.anl.gov/ AR/otter/).

29. Weidenbach C., Brahm U., Hillenbrand T., Keen E., Theobald C., and Topic D. SPASS Version 2.0. Voronkov A. (Ed.), Automated Deduction - CADE-18: 18th International Conference on Automated Deduction, Vol. 2392 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 275-279 (The SPASS Prover: http://spass.mpi-sb.mpg.de/).

30. Riazanov A. and Voronkov A. The design and implementation of Vampire, AI Communications, 2002, 15 (2?3), 91-110. (The Vampire prover: http://www.cs.man.ac.uk/ riazanoa/ Vampire/).

31. Aselderov Z., Verchinine K., Degtyarev A., Lyaletski A., Paskevich A., and Pavlov A. Linguistic tools and deductive technique of the System for Automated Deduction. Proceedings of the 3rd Intern. Workshop on the Implementation of Logics, Tbilisi, Georgia, 2002, 21-24.

32. The QED Project: http://www.mcs.anl.gov/qed/.

33. Efficient quantifiers manipulation in Gentzen's calculi LK and LJ. Abstracts of the International meeting "2nd St.-Petersburg Days of Logic and Computability" devoted to the centennial of A.A.Markov, 24-26 августа 2003 г.

34. Admissible Substitutions in Classical and Intuitionistic Sequent Logics. Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM?04 Conference held in Zakopane, Poland, May 17-20, 2004 Series: Advances in Soft Computing, Physica/Springer Verlag, Heidelberg New York, 2004, XIV, 49-58.

35. Лялецкий А.В. Секвенциальный формализм и дедуктивные системы для классической логики первого порядка. Proc. of Intern. Conf. "Logic and Applications", Новосибирск, 2000.

1 сентября 2005

Киевский национальный университет имени Тараса Шевченко, Киев, Украина

E-mail: forlav@bigmir.net, ppp_exter@univ.kiev.ua