

DOI: 10.15514/ISPRAS-2025-37(6)-48



## Выбор интерфейса системы Isabelle для взаимодействия с большими языковыми моделями

<sup>1</sup> М.У. Мандрыкин, ORCID: 0000-0002-9306-7719 <mandrykin@ispras.ru>

<sup>1,2</sup> В.С. Мутилин, ORCID: 0000-0003-3097-8512 <mutilin@ispras.ru>

<sup>1,3,4</sup> А.К. Петренко, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>

<sup>1,4</sup> К.С. Сорокин, ORCID: 0000-0002-6861-3802 <ksorokin@ispras.ru>

<sup>1,5</sup> И. Ю. Тюкин, ORCID: 0000-0002-7359-7966, <i.tyukin@skoltech.ru>

<sup>1</sup> Институт системного программирования РАН,  
109004, Россия, Москва, ул. А. Солженицына, д. 25.

<sup>2</sup> Московский физико-технический институт,  
141701, Московская область, г. Долгопрудный, Институтский переулок, д. 9.

<sup>3</sup> Московский государственный университет имени М.В. Ломоносова,  
119991, Россия, Москва, Ленинские горы, д. 1.

<sup>4</sup> НИУ Высшая школа экономики, 101978, Россия, Москва, ул. Мясницкая, д. 20.

<sup>5</sup> Сколковский Институт Науки и Технологий,  
121205, Россия, Москва, Большой Бульвар, д. 30 с. 1.

**Аннотация.** В статье дается обзор существующих работ по использованию языковых моделей (Large Language Models, LLM) для автоматизации построения формальных доказательств в системе Isabelle. Рассматриваются различные уровни взаимодействия пользователей и LLM с системой доказательства. Для каждого уровня рассматриваются преимущества и недостатки практического использования в ходе построения формальных доказательств пользователем, а также возможности автоматизации процесса доказательства с помощью LLM. На основе обзора предлагаются способы улучшения средств использования LLM в системе Isabelle.

**Ключевые слова:** автоматизация верификации; дедуктивная верификация; большие языковые модели LLM; ИИ-ассистент.

**Для цитирования:** Мандрыкин М.У., Мутилин В.С., Петренко А.К., Сорокин К.С., Тюкин И.Ю. Выбор интерфейса системы Isabelle для взаимодействия с большими языковыми моделями. Труды ИСП РАН, том 37, вып. 6, часть 4, 2025 г., стр. 27-44. DOI: 10.15514/ISPRAS-2025-37(6)-48.

## Choosing an Isabelle System Interface for Interacting with Large Language Models

<sup>1</sup> M.U. Mandrykin, ORCID: 0000-0002-9306-7719 <mandrykin@ispras.ru>

<sup>1,2</sup> V.S. Mutilin ORCID: 0000-0003-3097-8512 <mutilin@ispras.ru>

<sup>1,3,4</sup> A.K. Petrenko, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>

<sup>1,4</sup> K.S. Sorokin, ORCID: 0000-0002-6861-3802 <ksorokin@ispras.ru>

<sup>1,5</sup> I.Y. Tyukin, ORCID: 0000-0002-7359-7966 <i.tyukin@skoltech.ru>

<sup>1</sup> Institute for System Programming of the Russian Academy of Sciences,  
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

<sup>2</sup> Moscow Institute of Physics and Technology,

9, Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.

<sup>3</sup> Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia.

<sup>4</sup> National Research University, Higher School of Economics,  
20, Myasnitskaya ulitsa, Moscow, 101978, Russia.

<sup>5</sup> Skolkovo Institute of Science and Technology,

Bol'shoi Bol'var 30 c 1, Moscow, 121205, Russia.

**Abstract.** The paper presents a review of existing research on the use of large language models (LLMs) for automating the construction of formal proofs in the Isabelle proof assistant. It examines several levels of interaction between users or LLMs and the proof system. For each level, the paper discusses the advantages and limitations of practical use during routine proof development, as well as the current and potential opportunities for automating the proof process with LLMs. Based on this review, the paper proposes directions for improving the integration of LLM-based proof automation tools into the Isabelle system.

**Keywords:** verification automation; deductive verification; LLM; proof assistant.

**For citation:** Mandrykin M.U., Mutilin V.S., Petrenko A.K., Sorokin K.S., Tyukin I.Y. Choosing the interface of the Isabelle system for interacting with large language models. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 6, part 4, 2025., pp. 27-44 (in Russian). DOI: 10.15514/ISPRAS-2025-37(6)-48.

### 1. Введение

Система интерактивного доказательства теорем Isabelle [1-3] является одной из самых широко используемых [4] и простых в освоении [5-6] систем подобного рода наряду с аналогичными инструментами, такими как Rocq [7], Lean 4 [8], HOL 4 [9], HOL Light [10] и Metamath [11]. В области автоматизации процессов формализации и доказательства теорем в системе Isabelle с помощью глубокого машинного обучения, в частности на основе больших языковых моделей, уже существует достаточно большое количество работ и инструментов [12-17]. Пока нет достаточно согласованного понимания как текущих, так и потенциальных практических возможностей таких инструментов в смысле повышения эффективности или автоматизации работы исследователей и инженеров, использующих систему Isabelle в областях компьютерной науки, математики и формальных методов. При этом наиболее очевидным способом оценки этих возможностей кажется эмпирический подход, основанный на накоплении соответствующего достаточного практического опыта применения больших языковых моделей (Large Language Models, LLM) и нейронного доказательства теорем (Neural Theorem Proving, NTP) в соответствующих областях.

Наш собственный опыт использования системы Isabelle и наблюдения за состоянием сообщества ее пользователей подсказывает, что для накопления такого опыта могут возникать как технические, так и концептуальные препятствия. Такие препятствия связаны с тем, что многие исследования в области NTP изначально не направлены на интеграцию разрабатываемых методов в повседневную практику использования инструмента, и поэтому для достижения лучших показателей работы LLM на тестовых выборках в них часто

используются интерфейсы и уровни взаимодействия с системой интерактивного доказательства, существенно отличные от тех, с которыми обычно работают пользователи системы. В частности, в исследовательских экспериментах могут использоваться как более низкоуровневые (и часто неудобные для пользователей) интерфейсы системы, чем обычно используются на практике (такие как длинные тактические доказательства [18-20]), так и наоборот, более высокоуровневые интерфейсы (такие как новые, преимущественно декларативные языки структурированных доказательств [16]), с помощью которых на практике часто получают корректные, но неэффективные и трудно сопровождаемые доказательства.

Основные разработчики системы Isabelle на сегодняшний день пока не отдают явного предпочтения ни одному из существующих подходов и в явном виде не планируют скорую интеграцию какого-либо из них в базовую сборку системы. Такая ситуация создает большую неопределенность для пользователей инструмента, которая усиливается быстро случившейся интеграцией нескольких известных и широко используемых LLM (ChatGPT и Claude Sonnet) в базовый пользовательский интерфейс системы интерактивного доказательства Lean (несмотря на наличие достаточно большого разнообразия других, более специфичных для этой системы, разработок [21-22]).

В данной статье мы рассмотрели ключевые результаты, посвященные использованию LLM для автоматизации построения формальных доказательств в системе Isabelle, а также в других системах интерактивного доказательства – в тех случаях, когда соответствующие подходы могут быть гипотетически перенесены в Isabelle. Анализ проводился преимущественно с точки зрения применяемых интерфейсов взаимодействия между LLM, системой доказательства и пользователем, а также с позиции их интегрируемости в повседневные рабочие процессы.

На основе проведенного обзора мы предложили два возможных направления для улучшения интеграции средств работы с LLM в Isabelle. Первое направление предполагает создание нового полноценного интерфейса синхронного взаимодействия с LLM, построенного на базе Isabelle/Scala. Второе направление ориентировано на разработку легкого чат-интерфейса, обеспечивающего удобный перенос автоматически размеченных фрагментов формального текста между сообщениями и областями ввода-вывода в редакторе Isabelle/jEdit, который остается наиболее распространенным инструментом среди пользователей Isabelle.

Выбранная классификация существующих подходов, основанная на типах интерфейсов взаимодействия с системой интерактивного доказательства, позволяет выявить концептуальные различия между потенциальными способами интеграции LLM в инфраструктуру Isabelle. Такой подход делает возможным более четкое понимание преимуществ и ограничений различных интерфейсов, а также определение направлений, наиболее перспективных для практической автоматизации.

#### **Вклад работы состоит в следующем:**

- проведен обзор уровней взаимодействия системы Isabelle с пользователем и определены соответствующие возможности автоматизации с помощью LLM; для каждого уровня описаны преимущества и недостатки существующих решений;
- предложены перспективные направления развития инструментов, ориентированные как на повышение качества взаимодействия пользователя с системой, так и на расширение возможностей автоматизации за счет более глубокой интеграции LLM.
- Этот анализ и сформулированные рекомендации могут служить основой для дальнейших исследований и разработки средств, обеспечивающих более эффективное применение LLM в экосистеме Isabelle.

Дальнейшее изложение имеет следующую структуру: во втором разделе мы кратко опишем исторически сложившиеся интерфейсы взаимодействия с системой Isabelle, образующие

своего рода уровни представления формальных доказательств, а также новые уровни, добавляемые в систему при некоторых способах интеграции в нее интерфейсов взаимодействия с LLM. Во третьем разделе рассмотрены основные преимущества и недостатки использования каждого уровня взаимодействия с системой Isabelle для интеграции в нее средств взаимодействия с LLM. В четвертом разделе мы представили свои предварительные выводы и оценку перспективности использования различных способов взаимодействия LLM с системой Isabelle/HOL с точки зрения простоты интеграции получаемых инструментов NTP в повседневный процесс практического использования системы.

## 2. Интерфейсы и уровни взаимодействия с системой Isabelle

Система Isabelle разрабатывается уже в течение почти 40 лет (с 1986 года) [23] и в целом ее развитие постоянно сопровождалось появлением новых интерфейсов взаимодействия с пользователем, позволяющих сделать процесс формализации математических (логических символьных) моделей и доказательств более удобным и эффективным. При этом логическое ядро системы за это время претерпело достаточно мало существенных изменений, а более старые интерфейсы взаимодействия лишь дополнились новыми, но не заменились ими полностью, оставив ранние интерфейсы взаимодействия с системой доступными практически в неизменном виде. Таким образом, достаточно уместно говорить скорее об уровнях или слоях взаимодействия с системой Isabelle, которые последовательно дополняют и абстрагируют соответствующие нижележащие интерфейсы, оставляя при этом их доступными для использования.

### 2.1 Логическое ядро

Базовый уровень взаимодействия с системой Isabelle – это уровень взаимодействия с логическим ядром системы, которое образует ее доверенную кодовую базу и на котором основана логическая корректность разрабатываемых с ее помощью формальных доказательств. Логическое ядро системы предоставляет интерфейсы для работы с теоремами, имеющими вид  $\Theta, \Gamma \vdash P x \Rightarrow C x$ , где  $\Theta$  – это теория, содержащая символы и аксиомы используемой формальной логики (обычно, логики высшего порядка – HOL), а также все ранее введенные определения и ранее доказанные теоремы,  $\Gamma$  – набор гипотез, используемых теоремой,  $P x \Rightarrow C x$  – формулировка теоремы в виде формулы Хэрропа (обобщения хорновского дизъюнкта для логик высшего порядка). Не вдаваясь в логические подробности, отметим, что теоремы в таком виде представляют собой произвольно рекурсивно вложенные импликации (вида  $A \Rightarrow B$ ), антецеденты ( $A$ ) и консеквенты ( $B$ ) которых могут начинаться с произвольного количества кванторов всеобщности (которые в системе Isabelle обозначаются символом  $\Lambda$ ). Листовые (атомарные) предикаты таких формул представлены формулами просто типизированного  $\lambda$ -исчисления, то есть, по сути, состоят из комбинации операций абстракции ( $\lambda x. f$ ) и аппликации ( $f x$ ), а также нескольких (3-х) видов переменных и констант. Для формализации логики высшего порядка над таким представлением утверждений остается лишь определить  $\alpha\beta\eta$ -нормализацию (с помощью частичной аксиоматизации равенства –  $\equiv$ ) и реализовать правило резолюции высшего порядка. На практике это означает, что весь интерфейс логического ядра такой системы может быть сведен к построению формул в просто типизированном  $\lambda$ -исчислении с использованием фиксированных символов  $\Rightarrow, \Lambda$  и  $\equiv$ , применению аксиом равенства (с точностью до  $\alpha\beta\eta$ -эквивалентности) и всего одного правила вывода – резолюции высшего порядка. На самом деле интерфейс ядра Isabelle несколько больше, но остальные функции в нем представлены фактически лишь с целью оптимизации наиболее часто используемых манипуляций представлениями теорем. Ядро Isabelle полностью реализовано на языке StandardML и написано с использованием только

чистого функционального программирования. Таким образом, интерфейсом логического ядра системы является относительно небольшая сигнатура модуля `Thm` и нескольких используемых им модулей (в основном `Term`, `Sign`, `Theory` и `Unify`) на языке `StandardML`. Доказательства на этом уровне взаимодействия представляют собой программы (выражения) на языке `StandardML`, выполнение которых приводит к построению соответствующих требуемых теорем через интерфейсы логического ядра Isabelle. Так как теорема представлена в сигнатуре модуля `Thm` абстрактным типом, система типов языка `StandardML` не позволяет получать соответствующие значения этого типа (теоремы) никаким другим способом, кроме как с помощью вызовов интерфейсов ядра системы Isabelle. Интерфейсы же ядра, по сути, соответствуют правилам вывода логики высшего порядка.

## 2.2 Тактические доказательства

Следующий уровень взаимодействия с системой – уровень тактических доказательств в системе Isabelle реализован как доменно-специфичное подмножество (DSL) языка `StandardML`, предоставляющее средства для использования широко распространенного приема построения низкоуровневых логических доказательств – обратного рассуждения (англ. *backward reasoning*). Этот прием заключается в том, что вначале вместо требуемого утверждения  $G$  формулируется тривиальная тавтология  $G \Rightarrow \# G$ , где  $\# Q \equiv Q$  для любого предиката  $Q$  по определению (символ  $\#$  используется для отделения утверждений с несколькими импликациями). Затем доказательство утверждения  $G$  сводится к последовательному применению специальных производных правил вывода – тактик, которые всегда (с точностью до инстанцирования схематических переменных) видоизменяют только посыл самой внешней импликации теоремы (слева от  $\Rightarrow$ ), оставляя неизменным ее “защищенное” символом  $\#$  следствие  $G$ , соответствующее исходно требуемой формулировке. В конечном итоге, когда посыл внешней импликации становится тривиальным, соответствующая теорема вида  $T \Rightarrow \# G$  с помощью какой-либо простой тактики сводится к виду  $\# G$ , после чего определение символа  $\#$  раскрывается через интерфейс ядра Isabelle, порождая теорему с изначально требуемой формулировкой (с точностью до возможных подстановок вхождений схематических переменных). При разработке тактических доказательств используются специальные термины, соответствующие конфигурации посылы самой внешней импликации последовательно преобразуемой теоремы. Для посылы вида  $G_1 \Rightarrow \dots \Rightarrow G_n$  Хэрроповские формулы  $G_1, \dots, G_n$  называются текущими *подцелями* доказательства (идентифицируемыми по порядковым номерам), а в каждой такой подцели  $G_i$  вида  $\lambda x. A_1 x \Rightarrow \dots \Rightarrow A_n x \Rightarrow C x$  связанные переменные  $x$  называются *параметрами* подцели, предикаты  $A_1 x, \dots, A_n x$  называются *предположениями* этой подцели, а  $C x$  – ее *выводом*. Таким образом, каждая новая подцель фактически является новым целевым утверждением для доказательства, которое затем с помощью последовательного применения тактик постепенно сводится к набору новых целевых утверждений (обычно более простых), и так до тех пор, пока в конечном итоге все полученные целевые утверждения не окажутся тривиально разрешимыми с помощью применения подходящих тактик доказательства. При использовании такого подхода доказательства теорем также представляются выражениями на языке `StandardML`, но с использованием специализированного DSL тактических доказательств, предоставляющего готовые (но легко расширяемые) наборы популярных тактик и их комбинаторов. В целом такой интерфейс взаимодействия с системой интерактивного доказательства является в некотором смысле консенсусом среди сразу нескольких наиболее распространенных аналогичных систем, в частности `HOL4`, `HOL Light`, `Rocq` и `Lean` концептуально предоставляют именно интерфейс тактических доказательств в качестве основного интерфейса взаимодействия с пользователем системы. `HOL4` и `HOL Light` фактически предоставляют интерфейс тактических доказательств, полностью аналогичный соответствующему уровню взаимодействия с системой Isabelle, но предлагающий большее разнообразие тактик и их комбинаторов. В `Rocq` и `Lean`, в отличие от Isabelle, помимо

большого разнообразия тактик и комбинаторов, также есть возможность обращения к параметрам и предположениям подцелей доказательства (в Rocq и Lean они называются гипотезами) по их именам, чего не могут непосредственно делать тактики системы Isabelle, так как соответствующие сущности представлены глубоко внутри формулировки преобразуемой целевой теоремы, а представление термов в системе Isabelle не подразумевает наличия внутренней разметки. Подобные возможности технически реализуемы другими средствами (например, с помощью тегов теорем), но практическая потребность в них существенно ниже по причине активного использования более высокоуровневых интерфейсов. Возможно, отличие в используемых интерфейсах – одна из основных причин, по которым ранние исследования в области NTP (например, [18-20]) практически не ориентировались на систему Isabelle, так как они были направлены на генерацию именно тактических доказательств.

В этом смысле Isabelle наиболее существенно отличается от большинства широко распространенных аналогичных систем интерактивного доказательства тем, что в качестве основного (предпочитаемого) интерфейса взаимодействия с пользователем в ней был выбран другой интерфейс, лежащий сразу на несколько уровней абстракции выше обычных тактических доказательств. В конечном счете – это уровень интерактивных формальных документов, разрабатываемых в среде Isabelle/jEdit (или Isabelle/VSCode). Но в системе Isabelle между уровнем тактических доказательств и уровнем интерактивного формального документа лежат еще 3 достаточно отдельных и самостоятельных уровня абстракции.

## 2.3 Структурированные доказательства

Первый из них – уровень структурированных доказательств (семантически соответствующий командам языка Isag [24]) добавляет к тактическим комбинаторам возможность гибкого управления параметрами, предположениями и выводами подцелей доказательства с помощью механизма контекстов и состояний доказательства, а также добавляет концептуально новую возможность сложной “горизонтальной” декомпозиции доказательств с помощью механизма локальных фактов (именованных списков теорем), полностью ортогонального к обычной, “вертикальной” (иерархической) декомпозиции доказательств с помощью последовательного применения тактик. Основой построения структурированных доказательств в системе Isabelle является блок структурированного доказательства, который находится в отношении уточнения с одной или несколькими текущими подцелями внешнего доказательства. Уточнение подразумевает, что параметры, предположения и выводы блока доказательства являются подмножествами параметров, предположений и выводов соответствующих подцелей доказательства с точностью до унификации. Механизм уточнения (англ. goal refinement) позволяет произвольно именовать и манипулировать параметрами, предположениями и выводами подцелей доказательства с помощью структурированных блоков. Но, помимо этого, внутри структурированных блоков можно также формулировать и доказывать произвольные именованные локальные факты (списки теорем) с помощью тех же механизмов, которые используются для теорем верхнего уровня. Это дает возможность произвольной горизонтальной декомпозиции доказательств внутри блока с использованием произвольных графов зависимостей между такими локальными фактами, которые также доступны и во вложенных блоках доказательства (с учетом порядка формулирования и доказательства этих фактов). Состояния доказательства внутренних фактов блока при этом никак не связаны с состоянием соответствующего внешнего доказательства. Прагматической традицией формализации доказательств в системе Isabelle является существенное ограничение языка тактик, используемых для организации внешних доказательств (такие тактики часто называют начальными методами доказательства – англ. initial proof methods, например, standard, induction, cases, rule, intro, elim, subst), а также перенос применений сложных, высоко автоматизированных тактик (методов) в листовые доказательства, не использующие вложенные блоки (соответствующие тактики называют

окончными или терминальными методами – англ. *terminal proof methods*, например, *auto*, *blast*, *force*, *metis*, *meson*, *smt*). Таким образом, на уровне структурированных доказательств способом их представления является древовидная композиция блоков доказательства, связанных вертикальными (применение тактик/методов) и горизонтальными (использование локальных фактов) связями. Для этого у блоков доказательств предусмотрены также специальные параметры для возможного задания соответствующих начальных и терминальных методов. Тривиальный блок доказательства, состоящий из последовательного применения соответствующего начального и терминального метода, является специальным случаем, для которого (на нескольких последующих уровнях взаимодействия) предусмотрено специальное сокращенное обозначение (функция *terminal proof*/команда *by*). Важно отметить, что непосредственно на уровне структурированных доказательств их представлением по-прежнему являются выражения языка StandardML, в которых можно произвольно комбинировать соответствующие интерфейсы работы с персистентными состояниями доказательства – стеками блоков структурированных доказательств. Установление соответствия последовательности состояний такого стека блоков доказательств с некоторой квази-синтаксической структурой последовательного текста доказательства на подходящем доменно-специфичном языке является задачей следующего уровня.

## 2.4 Уровень транзакций

Следующий уровень взаимодействия – уровень транзакций к общему состоянию. Задачей этого уровня фактически является переворачивание всего предыдущего процесса взаимодействия с системой интерактивного доказательства. В то время как на всех предыдущих уровнях взаимодействия доказательства представляются выражениями на языке StandardML, изначально имеющими рекурсивную форму, соответствующую грамматике этого языка программирования, уровень транзакций представляет процесс доказательства в виде последовательности связанных общим изменяемым состоянием транзакций, которые соответствуют изменению некоторого общего состояния, называемого состоянием узла документа (англ. *node state*) или состоянием теории. Модули StandardML становятся частью состояния теории, а взаимодействие с системой доказательства на следующих уровнях происходит с помощью асинхронного выполнения транзакций к состояниям теорий, которые могут включать инкрементальную компиляцию и компоновку модулей или отдельных выражений на языке StandardML. Таким образом, уровень транзакций делает возможным задействование механизмов асинхронного выполнения и планирования транзакций, а также обработки ошибок с автоматическим восстановлением последнего консистентного состояния и персистентного сохранения таких состояний узлов с помощью (де-)сериализации. Это существенно влияет на восприятие всей системы Isabelle, так как, по сути, к ней напрямую не применимы такие понятия, как “компиляция” или “сборка” теорий, по аналогии с теориями Rosc или Lean. Теории Isabelle в этом смысле по сути являются лишь мгновенными снимками некоторого консистентного состояния, которое может быть потенциально относительно легко заменено на любое другое состояние этой же теории, например, на предыдущее. В частности, планирование транзакций в системе Isabelle выполняется из оптимистичных предположений, из чего следует, что в системе нет однозначного момента окончания “сборки” какой-либо теории, так как транзакции зависимых от нее дальнейших теорий могут начать выполняться до окончания выполнения символьных вычислений из вложенных блоков этой теории и, как следствие, сама теория может впоследствии оказаться незавершенной из-за ошибки в доказательстве в одном из ее внутренних блоков. Эти особенности также сказываются при разработке средств автоматизации доказательств с помощью агентов, так как система Isabelle изначально рассчитана на работу с агентами-пользователями, поддерживающими асинхронное взаимодействие с возможностью позднего обнаружения ошибок, что не предусмотрено в современных протоколах взаимодействия с

LLM, такими как протокол MCP (да и в самом процессе работы современных моделей искусственного интеллекта в целом).

В частности, при обучении и тестировании некоторых экспериментальных языковых моделей формальных доказательств в системе Isabelle, например, [16, 25], были разработаны и задействованы новые уровни взаимодействия с системой Isabelle в синхронном режиме, основанные на уровне транзакций, но добавляющие поверх него соответствующие механизмы синхронного ожидания результатов выполнения транзакций (например, с помощью активного ожидания и экспоненциального отката, англ. exponential back-off).

## 2.5 Уровень интерактивного формального документа Isabelle/Scala и графический интерфейс Isabelle/jEdit

Основной же механизм взаимодействия с системой Isabelle, предусмотренный ее разработчиками и практикуемый большинством ее пользователей, предполагает постепенное установление взаимно однозначного соответствия между транзакциями к состояниям узлов (теорий) и областями (текстовыми диапазонами) формального документа, представляющего процесс формального доказательства в системе Isabelle. Поскольку интерактивный формальный документ доказательства в системе Isabelle представляется в форме размеченного текста с интерактивными элементами, поверх уровня транзакций, все еще работающего со структурами данных языка StandardML, разработан уровень Isabelle/Scala, взаимодействующий со средой исполнения языка StandardML (в Isabelle используется специфичная платформенно-зависимая среда исполнения Poly/ML) через сетевой протокол PIDE (от англ. Proof IDE), основанный на использовании варианта расширяемого языка разметки XML. Соответствующий уровень помимо предоставления нового интерфейса взаимодействия с состояниями узлов с помощью размеченного текста также интегрирует все узлы одной разрабатываемой сессии формального доказательства в один документ, а также устанавливает порядок и границы областей этого документа, ставя в соответствие каждой такой области одну команду верхнего уровня. Таким образом, на уровне Isabelle/Scala, документ представляет собой ориентированный ациклический граф узлов-теорий, содержащий линейные последовательности асинхронно исполняемых команд (выполнение следующей команды может начинаться ранее окончания выполнения предыдущей, несмотря на строгий линейный порядок их расположения). Грамматика аргументов каждой конкретной команды при этом в общем случае независима друг от друга. Таким образом, текстовые представления теорий Isabelle лишены единой общей грамматики и обычно не рассматриваются как отдельный формальный язык. Впрочем, эта особенность практически не оказывает влияния на поддержку такого формата современными LLM, так как они достаточно легко распознают сложные контекстуальные зависимости (такие как расширение набора доступных команд, локальные изменения используемой грамматики термов и др.).

Уровень Isabelle/Scala также предоставляет основные программные интерфейсы, рекомендованные разработчиками Isabelle для использования в средствах взаимодействия пользователей с этой системой. Наиболее развитым и широко используемым средством такого взаимодействия является специализированный редактор Isabelle/jEdit, который предоставляет конкретный графический интерфейс пользователя для интерактивного редактирования содержимого формальных документов Isabelle. Внутри этого редактора также встраиваются плагины, реализующие серверную часть протокола MCP для использования системы Isabelle ИИ-агентами. На данный момент нам известна только одна публично доступная реализация такого сервера (инструмента), плагин I/Q, разрабатываемый лабораторией AWS Labs [26]. Однако в целом интерфейс, который может предоставлять ИИ-агентам сервер, реализованный в виде плагина редактора Isabelle/jEdit с использованием интерфейсов Isabelle/Scala, может быть достаточно произвольным, в том числе он может поддерживать обращение к состоянию документа как на уровне файлов и текстовых локаций, так и на уровне последовательностей команд и ориентированных графов зависимостей между

теориями, возможны также и более абстрактные представления, например, учитывающие структуру блоков доказательств и текстовых представлений термов в формулировках утверждений и целей доказательства (и таким образом формализующие, к примеру, такие понятия, как “текущий блок доказательства” или “последовательность подцелей доказательства, ведущая к текущему блоку”).

### 3. Сравнение возможностей уровней взаимодействия с системой Isabelle

Основные характеристики различных уровней взаимодействия с системой Isabelle, рассмотренные в данном разделе, кратко представлены в табл. 1. После приведенного ознакомительного описания существующих и разрабатываемых в исследовательских проектах уровней взаимодействия с системой Isabelle перечислим кратко основные наиболее очевидные преимущества и недостатки существующих и возможных реализаций взаимодействия с LLM на различных уровнях.

Табл. 1. Уровни взаимодействия с системой Isabelle.

Table 1. Levels of interaction with the Isabelle system.

Уровень взаимодействия с системой интерактивного доказательства	Объект воздействия	Представление состояния	Среда выполнения / язык программирования	Интерфейс взаимодействия
Логическое ядро	теорема	значения абстрактных типов языка ML и термы (направленные ациклические графы над символами)	Isabelle/ML (Poly/ML)	сигнатуры модулей ML
Тактические доказательства на ML	цель доказательства	абстрактные представления над термами	Isabelle/ML	сигнатуры модулей ML
Структурированные доказательства на ML	состояние доказательства	абстрактные представления над термами	Isabelle/ML	сигнатуры модулей ML
Транзакции к состоянию теории на ML	состояние теории/узла	абстрактные представления над термами	Isabelle/ML	сигнатуры модулей ML
Isabelle/Scala	состояние версии формального документа	объекты Scala/Java (JVM) над структурированным размеченным текстом XML	Isabelle/Scala (JVM)	протокол PIDE
Isabelle/jEdit	интерактивный формальный документ	структурированный размеченный текст	Isabelle/Scala (JVM)	GUI (Java/Swing)
Isa-REPL (синхронный режим)	глобальное состояние	абстрактное состояние сервера на стороне Isabelle/ML	Python и Isabelle/ML	Python API
Инструмент Isabelle (Isabelle tool)	— (без явного состояния)	JSON	Python	протокол MCP (JSON-RPC)

### 3.1 Уровни тактических и структурированных доказательств

Основные преимущества взаимодействия с системой Isabelle на уровнях тактических и структурированных доказательств:

- Возможность реализации и поддержки альтернативных, возможно, более специфичных для доказательств с помощью LLM, языков описания и структурирования доказательств, а также соответствующего набора тактик и методов доказательства. Возможность интеграции новых структурирующих конструкций с существующим языком доказательств Isar. Примерами подобных расширений системы Isabelle являются, в частности, достаточно широко используемый метод выбора подтермов на основе шаблонов [27] (предоставляющий возможности, аналогичные стандартным тактикам переписывания из систем Rocq и Lean), набор методов [28] E-унификации [29-30] и резолюции (расширяющих стандартные методы резолютивного вывода для автоматической поддержки классов эквивалентности различных синтаксических представлений равных термов), а также набор методов и язык высокоуровневого структурирования доказательств Auto2 [31] (реализующий поддержку мощного метода автоматического поиска доказательств, также основанного на E-унификации и резолюции) и команда *wlog* (англ. without loss of generality, не теряя общности рассуждений) [32] для оптимального структурирования на языке Isar доказательств, имеющих симметричную структуру (сразу рассматривающих с помощью одного блока доказательств множество или класс аналогичных случаев, совпадающих с точностью до какого-то соотношения). Подобные расширения набора методов и языка доказательств могут использоваться для поддержки совместимости с определенным специфичным стилем структурирования рассуждений, если он по каким-то причинам оказывается особенно эффективным при взаимодействии с LLM. Например, в работе [16] используется альтернативный язык записи структурированных доказательств IsaMini, который минимизирует использование в текстах доказательств явных упоминаний методов (тактик) Isabelle и их аргументов, уменьшая вероятность возникновения в выводах LLM возможных ошибок, связанных с детальным пониманием процесса работы этих методов (тактик). Конкретные вызовы тактик затем восстанавливаются с помощью более специализированных статистических (не языковых) моделей и мощных решающих процедур через интерфейсы инструментов Tty и Sledgehammer [33]. Использованный в работе [16] модифицированный язык доказательств IsaMini был фактически реализован в виде нового уровня представления доказательств поверх уровня тактик и отчасти непосредственно поверх логического ядра (а также опирается на использование метода `auto_sledgehammer` тех же авторов).
- Возможность симуляции поведения других аналогичных систем доказательства, в особенности основанных на логике высшего порядка (HOL4 и HOL Light), с помощью портирования в систему Isabelle используемых в них наборов тактик и их комбинаторов. Большие библиотеки математических доказательств ранее уже были успешно портированы из системы HOL4 в систему Isabelle, в том числе с использованием автоматических трансляторов [34-35]. Другим примером успешного портирования инструментария HOL 4 в систему Isabelle является суперпозиционный решатель Metis [36-37] и соответствующий метод доказательства. Подобная поддержка тактик и комбинаторов HOL 4 или HOL Light с помощью их портирования в систему Isabelle может позволить напрямую использовать модели, обученные генерации доказательств для этих систем, если это окажется необходимым.

Среди основных недостатков непосредственного использования уровней тактических и структурированных доказательств при взаимодействии с LLM можно отметить следующие:

- Невозможность оперирования с сущностями более высоких уровней взаимодействия, таких как узлы (теории), сессии и общая структура формального документа (в особенности, структура сессий и зависимости между ними). Тактики, методы и функции уровня доказательства работают в заданном более высокими уровнями взаимодействия формальном контексте, поэтому, в частности, они не могут самостоятельно запрашивать или импортировать формальные сущности из любых произвольно заданных теорий Isabelle. Например, для семантического поиска в больших библиотеках существующих математических формализаций вроде архива формальных доказательств AFP [38], необходимо взаимодействие с системой Isabelle по крайней мере на уровне транзакций (для инициализации начального состояния контекста требуемой теории из сериализованного представления).
- Отсутствие доступа к артефактам взаимодействия с пользователем на более высоких уровнях абстракции, в частности, к артефактам асинхронного взаимодействия. В системе Isabelle помимо основного пути взаимодействия с пользователем через текущий контекст доказательства, реализации команд, синтаксических трансляций и методов доказательства могут асинхронно взаимодействовать с пользователем и даже средой выполнения системы Isabelle (Isabelle/Scala, JVM и ОС) с помощью динамической разметки формального текста, отправки размеченных информационных сообщений и предупреждений, а также ряда других средств (диалоговых окон, реактивных областей ввода и даже запуска разрешенных внешних процессов на машине пользователя, например, для эффективного поиска доказательств с помощью внешних решающих процедур). Все эти возможности не предоставляются через обычные интерфейсы уровней тактик и структурированных доказательств и требуют использования более высоких уровней взаимодействия, как правило, асинхронного взаимодействия с Isabelle/Scala по протоколу PIDE (с помощью отправки сообщений и использования отложенных вычислений – англ. *promises* или *futures*).
- Взаимодействие с LLM непосредственно на уровне тактик и структурированных доказательств не соответствует наиболее распространенному процессу взаимодействия пользователей системой Isabelle, в частности, на этих уровнях не представлен конкретный синтаксис команд формального документа, и, в частности, языка Isar, использованию которого обучаются LLM, использующие в своих выборках большинство открытых и широко доступных материалов, описывающих использование этой системы (книги, руководства, библиотеки доказательств и др.). Для обучения моделей в работе [16] был реализован специальный транслятор доказательств на основе правил трансформации, который применялся к материалам из архива AFP.

## 3.2 Уровень транзакций

Основные преимущества реализации взаимодействия с LLM на уровне транзакций:

- Отсутствие необходимости поддержки консистентного текстового представления фрагментов формального документа, учитывающего конкретные текстовые диапазоны команд, вызовов методов и выражений языка термов. В отличие от уровня Isabelle/Scala, работающего с динамически и асинхронно размечаемым текстом, в котором отдельные диапазоны символов могут выступать в качестве локаций для размещения элементов разметки, перекрестных ссылок, вложенных информационных сообщений, уведомлений, предупреждений и др. интерактивных

элементов, транзакции к состоянию теории могут вообще не иметь никакого конкретного текстового представления. Это может быть удобнее для реализации синхронного взаимодействия с LLM с помощью простого незамеченного текста. Подобный интерфейс удобнее реализовывать на уровне транзакций благодаря отсутствию необходимости использования конкретного синтаксиса существующего набора команд, в том числе команд языка Isar. В частности, интерфейсы системы Isabelle, использованные в работах [13-16], scala-isabelle [39], Portal to ISAbelle [40] и Isa-REPL [41], реализованы в основном на уровне транзакций.

- Возможность прямого доступа к внутреннему представлению термов и состояний контекста доказательства, который позволяет извлекать из них требуемые текстовые представления, оптимизированные для формирования текстовых сообщений при взаимодействии с LLM. На уровнях Isabelle/Scala и Isabelle/jEdit подобные специфичные запросы к содержимому логического контекста (поиск констант и теорем, отображение содержимого контекста доказательства или локалей) обычно осуществляется с помощью запроса к уровню транзакций через протокол PIDE (либо с помощью специальной отладочной команды формального документа, также посылающей свой асинхронный вывод через этот протокол).
- Возможность гибкого планирования и выполнения транзакций к состоянию теории, которую можно использовать для управления различными версиями документа, получаемыми, например, из альтернативных выборок вывода одной LLM (на одном запросе). В частности, на уровне транзакций относительно легко организовать разветвление состояния теории на несколько версий и даже последующее слияние нескольких версий (состояний) одной теории, если возникнет такая необходимость (механизм слияния теорий Isabelle, вообще говоря, может быть использован для этих целей, но использование его на более высоких уровнях взаимодействия ограничено инициализацией новых теорий).

Основные недостатки реализации взаимодействия с LLM на уровне транзакций:

- Отсутствие полного доступа к артефактам асинхронного взаимодействия с пользователем и окружением (средой выполнения системы Isabelle), таким как полная разметка структуры конкретного синтаксиса команд и термов (а также, например, результаты вывода типов констант и сортов переменных), информационные сообщения, подсказки и другие средства. Ситуация здесь примерно такая же, как и при использовании уровней тактик и структурированных доказательств.
- Отсутствие возможности непосредственного редактирования текстового представления формального документа с указанием конкретных файлов и текстовых диапазонов, например, для предоставления интерфейса взаимодействия с агентом (по протоколу MCP или другим способом).

### 3.3 Уровень Isabelle/Scala

Основные преимущества использования уровня Isabelle/Scala:

- Полный доступ ко всем артефактам и интерфейсам взаимодействия с пользователем системы и средой выполнения, включая асинхронные каналы (динамическая разметка текста теорий, сообщения, реактивные области ввода и др.) и вызовы внешних инструментов.
- Возможность свободного манипулирования всеми уровнями содержимого формальных документов, включая произвольную инициализацию теорий, сохранение и загрузку состояний сессий (групп теорий), а также управление несколькими параллельными версиями документов. На уровне Isabelle/Scala

возможна работа со всеми доступными установленной версии системы Isabelle сессиями и теориями, в частности, возможна реализация поиска по большим библиотекам формальных доказательств (таким как AFP).

- Отсутствие необходимости обязательной непрерывной синхронизации состояний и текста теорий, а также асинхронных каналов взаимодействия с уровнями Isabelle/ML (в частности, динамической разметки и сообщений) с соответствующими им визуальными графическими представлениями, которая возникает на уровне Isabelle/jEdit. Также отсутствует необходимость обязательной обработки событий графического интерфейса пользователя (в том числе в асинхронном режиме), которая возникает, например при реализации взаимодействия с LLM, например через элементы управления в редакторе Isabelle/jEdit (где пользователь и агент могут работать с одними и теми же элементами интерфейса).

Основные недостатки использования уровня Isabelle/Scala:

- Отсутствие прямого доступа к внутреннему представлению термов и состояний доказательства, в отличие, например, от уровня транзакций.
- Необходимость полной инициализации всего процесса взаимодействия с формальным документом, включая корректную инициализацию уровней Isabelle/ML, в чем нет необходимости при использовании уровня Isabelle/jEdit, в котором можно использовать текущую активную сессию работы пользователя системы Isabelle.
- Отсутствие непосредственной синхронизации с графическим представлением формального документа, например, в редакторе Isabelle/jEdit, которая может быть полезна, например, для быстрой оценки пользователем процесса или результатов работы LLM в режиме агента.

### 3.4 Уровень Isabelle/jEdit

Основные преимущества использования уровня Isabelle/jEdit:

- Возможность простой реализации имитации действий пользователя при работе LLM в режиме агента.
- Отсутствие необходимости полной инициализации системы Isabelle за счет использования текущей активной сессии.

Основные недостатки использования уровня Isabelle/jEdit:

- Ограниченность взаимодействия состоянием текущей активной сессии, так как для работы с доступными системе Isabelle, но не загруженными в текущей сессии теориями, необходимо использование интерфейсов уровня Isabelle/Scala.
- Необходимость корректной синхронизации с состоянием графического интерфейса пользователя, что может создавать трудности при необходимости работы с несколькими версиями (например, выборкой из) вывода LLM.

В целом большинство существующих работ в области автоматизации построения формальных доказательств в системе Isabelle либо основаны на использовании данных, извлеченных ранее из интерактивных документов Isabelle через различные специализированные инструменты, реализованные на базе интерфейсов уровня транзакций и Isabelle/Scala ([13-15]), либо непосредственно основаны на использовании интерфейсов уровней тактик, структурированных доказательств и транзакций ([16]), только уровня транзакций ([25]), или уровней Isabelle/Scala (через основанный на этом уровне инструмент *isabelle server* в [17]) или Isabelle/jEdit ([26]). Нам не удалось найти работы, в которых бы предпринимались попытки применения подходов для пошагового построения тактических доказательств (подобных [18-20]) к системе Isabelle.

#### 4. Оценка перспектив использования уровней взаимодействия с системой Isabelle

После рассмотрения преимуществ и недостатков различных интерфейсов Isabelle для интеграции больших языковых моделей перейдем к предварительной оценке перспектив их использования в повседневной практике формального доказательства.

- **Интерфейс уровня тактик и комбинаторов тактик.** Интерфейс этого уровня фактически не применяется в Isabelle уже более 25 лет – с момента широкого внедрения структурированных доказательств на языке Isar [24]. Интерфейс тактик был вынесен за пределы основного пользовательского интерфейса и сохранился лишь в ограниченном виде в минималистичном языке методов доказательства (команды *apply*, *by*, *proof*). Хотя комбинации тактик остаются востребованными при разработке новых методов доказательства в Isabelle/ML, их практическое применение в повседневной работе минимально. Это, вероятно, объясняет отсутствие исследований по использованию пошагового тактического доказательства с LLM в системе Isabelle.
- **Интерфейс уровня структурированных доказательств.** Подобные языки также не получили широкого распространения среди пользователей Isabelle. Практически вся библиотека AFP (около 5 млн строк) использует язык Isar, иногда с небольшими расширениями вроде команды *wlog* [32]. Кроме того, в последнее время появились работы, направленные на восстановление подробных структурированных доказательств из выводов SMT и суперпозиционных решателей [42-43], что косвенно указывает на высокий интерес пользователей к человекочитаемым представлениям автоматически генерируемых доказательств. В такой ситуации практическая ценность новых сильно абстрагированных от деталей рассуждений языков доказательства, таких как IsaMini [16], остается неочевидной.
- **Интерфейс уровня транзакций.** Они удобны для извлечения из теорий Isabelle различных представлений, востребованных при формировании обучающих выборок для LLM. Однако интеграция взаимодействия с натренированными на таких данных моделями в распространенные пользовательские среды системы Isabelle (Isabelle/jEdit, Isabelle/VSCode) может оказаться нетривиальной задачей. Главная проблема – узкая специализация моделей, обученных выполнять один тип действия: например, дополнять последовательность команд Isar или генерировать один терминальный блок доказательства. На практике же пользователям требуется помощь в гораздо более разнообразных задачах: формулировке определений, создании нотаций, работе с локалями [44], диагностике причин неуспешного применения методов доказательства, оптимизации структуры доказательств и др. Существенна и роль LLM в образовательных сценариях, которые предполагают свободный диалог – режим, фактически отсутствующий в большинстве современных датасетов по NTP [12-16, 18-20, 45].
- **Интерфейс уровня Isabelle/Scala.** Известен как минимум один пример практического применения LLM на уровне Isabelle/Scala для формализации результата из теории решеток в Isabelle/HOL [46]. Судя по доступным комментариям автора [47], модели использовались в свободном диалоговом режиме, включая инструмент StepProof [17], как чат для начального освоения Isar. Однако StepProof ограничен теорией *HOL.Main* и не интегрирован с Isabelle/jEdit, что затрудняет его использование в основной работе. Тем не менее сам факт успешного применения демонстрирует, что взаимодействие с Isabelle на уровне Isabelle/Scala (в случае StepProof – через *isabelle server* с ограниченным интерфейсом) может быть достаточно легко включено в практику.

- **Интерфейс уровня Isabelle/jEdit.** Эти интерфейсы наиболее доступны для реального использования. Однако наш опыт работы с агентом I/Q [26] (в частности, с моделями Claude Sonnet 4.5, Gemini 2.5-flash, Gemini 2.5-pro) выявил ряд существенных ограничений. Интерфейсы MCP оказываются трудными для некоторых моделей, что приводит к большому числу ошибок. Работа агента в документе параллельно с пользователем происходит слишком быстро и затрудняет своевременное вмешательство, при том что ошибки агента (например, чрезмерно глубокое разбиение доказательства на случаи, случайное удаление корректных блоков) возникают часто. Возможности использования агентов через внешние интерфейсы LLM-провайдеров (Claude Desktop, Google ADK) ограничены: нет удобного управления состоянием диалога, удаления неудачных шагов, сокращения контекста, сохранения и загрузки шаблонов запросов; интерфейсы плохо приспособлены для формального текста и символов Isabelle/Unicode. Кроме этого, такие решения обычно не позволяют запрашивать несколько моделей параллельно и свободно комбинировать полученные варианты доказательства.

Все это указывает на целесообразность дополнения интерфейса I/Q или другого агентного интерфейса средством взаимодействия с LLM в виде чата непосредственно в редакторе Isabelle/jEdit. Такой чат-интерфейс мог бы быть расширен рядом полезных функций: редактированием и удалением сообщений, удобным переносом автоматически размеченных фрагментов формального текста между сообщениями и областями ввода/вывода, а также другими инструментами, упрощающими взаимодействие с формальным документом.

Другим перспективным направлением является улучшение интерфейсов, доступных агентам, прежде всего через протокол MCP. Основные возможные улучшения включают:

- высокоуровневые средства выделения фрагментов текста (по именам теорем, номерам подцелей, шаблонам формулировок фактов и т.п. вместо работы с явными текстовыми диапазонами и регулярными выражениями);
- оптимизированные форматы представления отчетов о состоянии доказательства (например, трассировка шагов изменения целей, компактное отображение текущего доказательства без содержимого ветвящихся блоков, выборочное аннотирование типов и сортов);
- выделение самостоятельной версии документа, с которой может работать агент независимо от пользователя;
- автоматическое исправление терминальных доказательств с использованием инструментов `try` и `Sledgehammer`, аналогичное применяемому в работах [14, 45].

Если же реализация полноценной поддержки отдельного документа окажется слишком трудозатратной, возможен и более легкий вариант: предоставить агентам режим, в котором они могут запрашивать информацию о текущем состоянии формального документа, а также получать результаты гипотетических модификаций без внесения реальных изменений в файлы. Это может быть реализовано с опорой на существующий в Isabelle/PIDE механизм версий документа и журнал действий пользователя.

## 5. Заключение

В ходе работы мы обобщили и проанализировали современные исследования, посвященные применению больших языковых моделей (LLM) для автоматизации построения формальных доказательств в Isabelle и родственных системах интерактивного доказательства. Особое внимание было уделено уровням взаимодействия пользователей и LLM с ядром системы доказательства, а также тем интерфейсам, которые определяют удобство, гибкость и степень автоматизируемости рабочих процессов. Сопоставление существующих подходов позволило

вывить их сильные и слабые стороны с точки зрения практического использования и интеграции в повседневную работу пользователя.

На основании проведенного обзора мы предложили направления для совершенствования средств применения LLM в Isabelle, ориентированные как на повышение качества взаимодействия с пользователем, так и на расширение возможностей автоматизации. Эти рекомендации могут служить основой для дальнейших исследований и разработки инструментов, способствующих более глубокому и эффективному включению LLM в инфраструктуру современных систем интерактивного доказательства.

## Список литературы / References

- [1]. Paulson L. C. Isabelle – A Generic Theorem Prover (with a contribution by T. Nipkow). *Lecture Notes in Computer Science*, vol. 828, 1994.
- [2]. Isabelle Development Team. Isabelle. 1994–2025. Available at: <https://isabelle.in.tum.de/>, accessed 10.12.2025.
- [3]. Nipkow T., Wenzel M., Paulson L. C. Isabelle/HOL. A Proof Assistant for Higher-Order Logic. *Lecture Notes in Computer Science*, vol. 2283, 2002.
- [4]. Lincroft G., Minsung Cho, Hough K., Bazzaz M., Bell J. Thirty-Three Years of Mathematicians and Software Engineers: A Case Study of Domain Expertise and Participation in Proof Assistant Ecosystems. In: *Proceedings of the 21st International Conference on Mining Software Repositories: MSR '24, 2024*, pp. 1–13.
- [5]. Bayer J., David M., Pal A., Stock B. Beginners' Quest to Formalize Mathematics: A Feasibility Study in Isabelle. In: *Proceedings of Intelligent Computer Mathematics, CICM 2019, 2019*, pp. 16–27.
- [6]. Koutsoukou-Argyraiki A. Formalising Mathematics – in Praxis; A Mathematician's First Experiences with Isabelle/HOL and the Why and How of Getting Started. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, vol. 123, 2021, pp. 3–26.
- [7]. Rocq Development Team. The Rocq Prover. 1989–2025. Available at: <https://rocq-prover.org/>, accessed 10.12.2025.
- [8]. Lean Development Team. Lean. 2015–2025. Available at: <https://lean-lang.org/>, accessed 10.12.2025.
- [9]. HOL Development Team. HOL Interactive Theorem Prover. 1988–2025. Available at: <https://hol-theorem-prover.org/>, accessed 10.12.2025.
- [10]. Harrison J. The HOL Light theorem prover, 2025. Available at: <https://hol-light.github.io/>, accessed 10.12.2025.
- [11]. Megill N., Wheeler D. A. *Metamath: a computer language for mathematical proofs*, Lulu.com, 2019.
- [12]. Jiang A. Q., Welleck S., Jin Peng Zhou, Lacroix T., Jiacheng Liu, Wenda Li, Jamnik M., Lample G., Yuhuai Wu. Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs. In: *Proceedings of The Eleventh International Conference on Learning Representations, 2023*.
- [13]. Jiang A. Q., Wenda Li, Han J. M., Yuhuai Wu. LISA: Language models of isabelle proofs. In: *Proceedings of the 6th Conference on Artificial Intelligence and Theorem Proving, 2021*.
- [14]. Jiang A. Q., Wenda Li, Tworowski S., Czechowski K., Odrzygóźdź T., Miłoś P., Yuhuai Wu, Jamnik M. Thor: Wielding Hammers to Integrate Language Models and Automated Theorem Provers. In: *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022*.
- [15]. First E., Rabe M. N., Ringer T., Brun Y. Baldur: Whole-Proof Generation and Repair with Large Language Models. In: *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023:), 2023*.
- [16]. Qiyuan Xu, Renxi Wang, Haonan Li, Sanan D., Watt C. IsaMini: Redesigned Isabelle Proof Language for Machine Learning. *Препринт (arXiv:2507.18885)*, 2025.
- [17]. Xiaolin Hu, Qinghua Zhou, Grechuk B., Tyukin I. Y., Sutton O. StepProof: Step-by-step verification of natural language mathematical proofs. In *Proceedings: The Thirteenth International Conference on Learning Representations, ICLR 2025, 2025*.
- [18]. Bansal K., Loos S. M., Rabe M. N., Szegedy C., Wilcox S. HOList: An Environment for Machine Learning of Higher-Order Theorem Proving. In: *Proceedings of the 36-th International Conference on Machine Learning, 2019*.
- [19]. Huang D., Dhariwal P., Dawn Song, Sutskever I. GamePad: A Learning Environment for Theorem Proving. In: *Proceedings of the International Conference on Learning Representations, 2019*.

- [20]. Paliwal A., Loos S., Rabe M., Bansal K., Szegedy C. Graph Representations for Higher-Order Logic and Theorem Proving. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), 2020.
- [21]. Kaiyu Yang, Swope A. M., Gu A., Chalamala R., Peiyang Song, Shixing Yu, Godil S., Prenger R., Anandkumar A. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. In: NeurIPS 2023 Datasets and Benchmarks (NIPS'9), article 944, 2023, 40 pages.
- [22]. Del Tredici M., McCarran J., Breen B., Mijares J. A., Yin W. W., Taylor J. M., Koppens F. H. L., Englund D. Ax-Prover: A Deep Reasoning Agentic Framework for Theorem Proving in Mathematics and Quantum Physics, Препринт (arXiv:2510.12787), 2025.
- [23]. Paulson L. C. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, vol. 5, 1989, pp. 363–397.
- [24]. Wenzel M. Isar – A Generic Interpretative Approach to Readable Formal Proof Documents. In Proceedings: Theorem Proving in Higher Order Logics, 1999. pp. 167–183.
- [25]. Huerta y Munive J. J. Reaping the first fruits of the Isabelle/RL project. In: Proceedings of the 10th Conference on Artificial Intelligence and Theorem Proving, 2025.
- [26]. Becker H., Chong N., Dockins R., Grundy J., Hu J. Z. S., Mulder I., Mulligan D. P., Mure P., Paulson L. C., Slind K. I/Q (Isabelle/Q), an Isabelle/jEdit plugin exposing proof editing/exploration capabilities as an MCP server, 2025. Available at: <https://github.com/awslabs/AutoCorrode/tree/main/iq>, accessed 10.12.2025.
- [27]. Noschinski L., Traut C. Pattern-based Subterm Selection in Isabelle. In: Proceeding of Isabelle workshop, 2014. Также препринт (arXiv:2111.04082).
- [28]. Kappelmann K. Unification Utilities for Isabelle/ML. *Archive of Formal Proofs*, 2023.
- [29]. Snyder W. E-Unification (Book chapter). *A Proof Theory for General Unification*, 1991, pp. 49–60.
- [30]. Asperti A., Ricciotti W., Sacerdoti Coen C., Tassi E. Hints in Unification. In: Proceedings of the International Conference on Theorem Proving in Higher Order Logics, 2009. pp. 84–98.
- [31]. Bohua Zhan. AUTO2, A Saturation-Based Heuristic Prover for Higher-Order Logic. In: Proceedings of the International Conference on Interactive Theorem Proving, 2016. pp. 441–456.
- [32]. Unruh Dominique. Without Loss of Generality. *Archive of Formal Proofs*, 2024.
- [33]. Blanchette J. C., Böhme S., Paulson L. C. Extending Sledgehammer with SMT Solvers. In: Proceedings of the 23rd International Conf. on Automated Deduction (CADE-23), 2011, pp. 116–130.
- [34]. Obua S., Skalberg S. Importing HOL into Isabelle/HOL. In: Proceedings of the Third International Joint Conference on Automated Reasoning, 2006. pp. 298–302.
- [35]. Adams M. The Common HOL Platform. In: Proceedings of the 4th Workshop on Proof eXchange for Theorem Proving (PxTP'15), EPTCS 186, 2015. pp. 42–56.
- [36]. Hurd J. First-order proof tactics in higher-order logic theorem provers. In: Proceedings: Design and Application of Strategies/Tactics in Higher Order Logics, NASA/CP-2003-212448 in NASA Technical Reports, 2003, pp. 56–68.
- [37]. Paulson L. C., Susanto K. W. Source-Level Proof Reconstruction for Interactive Theorem Proving. In: Proceedings: Theorem Proving in Higher Order Logics, 2007. pp. 232–245.
- [38]. The Archive of Formal Proofs. 2004–2025. Available at: <https://www.isa-afp.org/>, accessed 10.12.2025.
- [39]. Unruh D. Scala-Isabelle, a library that allows to control an Isabelle prover process from a Scala application, 2020–2025. Available at: <https://github.com/dominique-unruh/scala-isabelle>, accessed 10.12.2025.
- [40]. Jiang A. Q., Wenda Li, Han J. M., Yuhuai Wu. PISA (Portal to ISAbelle). Available at: <https://github.com/albertqjiang/Portal-to-ISAbelle>, accessed 10.12.2025.
- [41]. Xu Qiyuan. Isabelle REPL. Unofficial support for Isabelle's Read-Eval-Print-Loop, 2024–2025. Available at: <https://github.com/xqyww123/Isa-REPL>, accessed 10.12.2025.
- [42]. Bartl L., Blanchette J., Nipkow T. Exploiting Instantiations from Paramodulation Proofs in Isabelle/HOL. In: Proceedings: Automated Deduction – CADE 30: 30th International Conference on Automated Deduction, 2025, pp. 573–593.
- [43]. Blanchette J. C., Böhme S., Fleury M., Smolka S. J., Steckermeier A. Semi-intelligible Isar Proofs from Machine-Generated Proofs. *Journal of Automated Reasoning*, vol. 56, 2015, pp. 155–200.
- [44]. Haftmann F., Wenzel M. Local Theory Specifications in Isabelle/Isar. In: Proceedings: Types for Proofs and Programs, 2008, pp. 153–168.
- [45]. Linzi A. L-Mosaics and Bounded Join-Semilattices in Isabelle/HOL. Препринт (arXiv:2509.19854), 2025.

- [46]. Rao B., Eiers W., Lipizzi C. Neural Theorem Proving: Generating and Structuring Proofs for Formal Verification. In: 19th Conference on Neurosymbolic Learning and Reasoning, Proceedings of Machine Learning Research, vol. 284, 2025, pp. 1–16.
- [47]. Linzi A. Personal AI Research Log. 2025. Available at: <https://github.com/linzialessandro/ai-research-log/tree/main/formal-verification>, accessed 10.12.2025.

## **Информация об авторах / Information about authors**

Михаил Усамович МАНДРЫКИН – кандидат физико-математических наук по специальности «математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей», научный сотрудник ИСП РАН. Сфера научных интересов: формальные методы верификации программ, математическая логика, функциональное программирование, системы типов в языках программирования.

Mikhail Usamovich MANDRYKIN – Cand. Sci. (Phys.-Math.), researcher at ISP RAS. Research interests: formal methods, program verification, mathematical logic, functional programming, type systems.

Вадим Сергеевич МУТИЛИН – кандидат физико-математических наук, ведущий научный сотрудник ИСП РАН и доцент Московского физико-технического института. Сфера научных интересов: статический и динамический анализ программ.

Vadim Sergeevich MUTILIN – Cand. Sci. (Phys.-Math.), leading researcher at ISP RAS and associate professor at Moscow Institute of Physics and Technology. Main research interests: static and dynamic program analysis.

Александр Константинович ПЕТРЕНКО – доктор физико-математических наук, профессор, заведующий отделом Технологий программирования ИСП РАН, профессор кафедр Системного программирования ВМК МГУ и ФКН НИУ ВШЭ. Научные интересы: формальные методы программной инженерии, языки спецификаций и моделирования, верификация.

Alexander Konstantinovich PETRENKO – Dr. Sci. (Phys.-Math.), Prof., Head of Software Engineering Department of ISP RAS, Professor of MSU and the Faculty of Computer Science, NRU HSE. Research their use in software development and verification.

Константин Сергеевич СОРОКИН – научный сотрудник отдела компиляторных технологий ИСП РАН. Научные интересы включают компиляторные технологии, статический и динамический анализ, применение искусственного интеллекта и машинного обучения для повышения производительности труда инженеров-программистов и специалистов по контролю качества.

Konstantin Sergeevich SOROKIN is a researcher at the Compiler Technology Department of ISP RAS. Research interests include compiler technologies, static and dynamic analysis, applications of AI/ML for enhancing productivity of software engineers and QA team members.

Иван Юрьевич ТЮКИН – доктор технических наук, профессор, главный научный сотрудник ИСП РАН, профессор центра Искусственного интеллекта Сколковского Института науки и технологий. Научные интересы: математика искусственного интеллекта, адаптивный, робастный и доверительный искусственный интеллект.

Ivan Yurievich TYUKIN – Dr.Sci. (Tech.), Prof., principal research scientist of the ISP RAS, Professor of the Centre for Artificial Intelligence at Skolkovo Institute of Science and Technology. Research interests: mathematics of Artificial Intelligence, adaptive, robust and trustworthy Artificial Intelligence.