ИСП РАН

Формальные методы разработки и верификации программ. Горизонт 2030

Александр Константинович ПЕТРЕНКО, petrenko@ispras.ru

«Программирование и вычислительная математика»
Конференция посвящённая 100-летию со дня рождения Николая Павловича Трифонова
3 декабря 2025 г.

Что такое формальные методы

ИСП РАН

Формальные методы в программировании (ФМ) (Formal Methods – FM) – это методы, для

- моделирования,
- спецификации,
- разработки,
- верификации программного и аппаратного обеспечения, основанные на активном использовании **математического аппарата.**

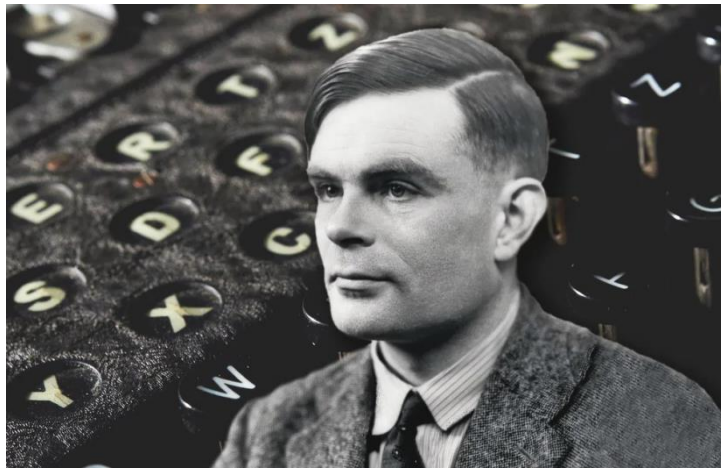
Периодизация

Предклассический период	50-60-е годы
Классический	конец 70-х – начало 2000-х
Постклассический	начало 2000-х – н.в.

Периодизация. Предклассический период

ИСП РАН

Предклассический период



А.Тьюринг

Классический

Постклассический

50-60-е годы

А. А. Ляпунов

конец 70-х – начало 2000-х

начало 2000-х – н.в.

Классический период

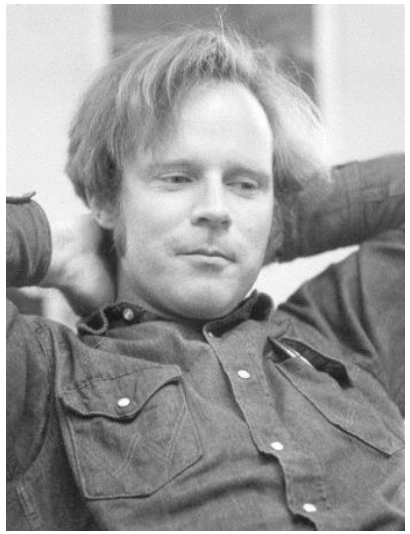
ИСП РАН

Предклассический период

Классический

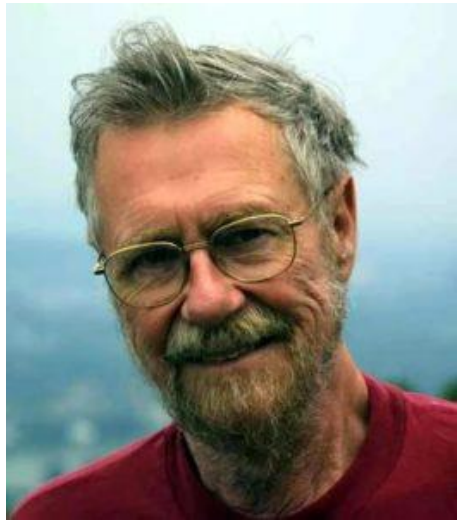
50-60-е годы

конец 70-х – начало 2000-х



Р. Флойд

С.А.Р Хоар



Э.Дейкстра

Д.Бьørнер

пред- и постусловия, инварианты, мат.индукция

VDM

Постклассический

начало 2000-х – н.в.

Периодизация

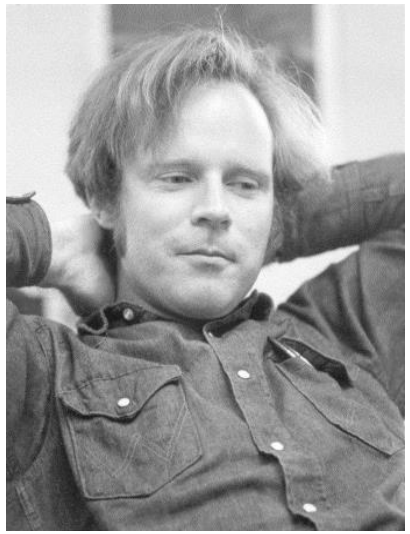
ИСП РАН

Предклассический период

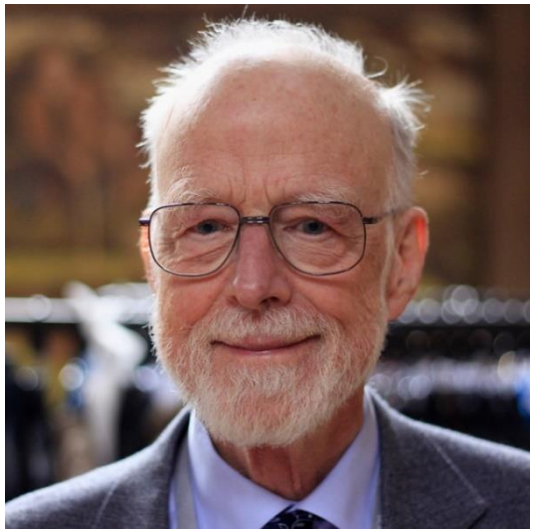
Классический

50-60-е годы

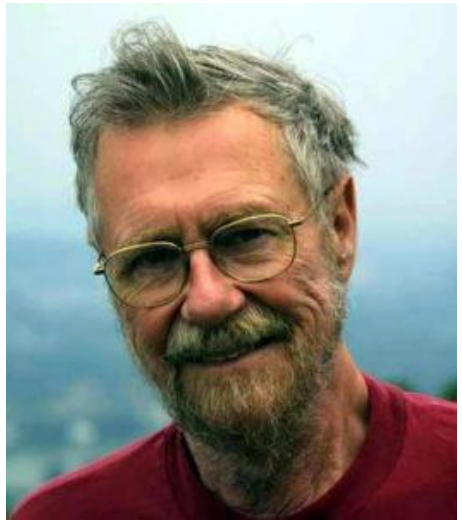
конец 70-х – начало 2000-х



Р. Флойд



Сэр Тони Хоар



Э.Дейкстра

Д.Бьørнер

пред- и постусловия, инварианты, матиндукция

VDM

Постклассический

начало 2000-х – н.в.

Периодизация

ИСП РАН

Предклассический период — 50-60-е годы

Классический — **конец 70-х – начало 2000-х**

Состояние дел:

- Число реальных успешных проектов исчисляется единицами
- Приложения: компиляторы, микроядерные операционные системы, телекоммуникационные и крипто-протоколы
- Предельный размер реализации – около 100 тыс. строк на Си или на языках моделирования

Постклассический — начало 2000-х – н.в.

Периодизация

ИСП РАН

Предклассический период — 50-60е годы

Классический — конец 70-х – начало 2000-х

Постклассический — начало 2000-х – н.в.

- Статический анализ – Coverity (США), Klockwork (Канада), Svace (Россия)
- Статическая верификация (software model checking)
  - CPAchecker (Германия), CBMC (Великобритания), LDV/Klever/CPAlocator (Россия), SDV/WDK/CodeQL (США)
- Тестирование на основе моделей (Model Based Testing) – UniTESK (Россия), SpecExplorer (США)
- Динамический анализ, фаззинг (DART, concolic testing) - SAGE (США)

Математическая и инструментальная база:

- Формальные спецификации, SAT- и SMT-солверы (Z3, cvc5, Alt-Ergo и др.) и пруверы

Инструменты РБПО*)

Разра-ботка

- **Svace и Binside: статический анализ исходного и бинарного кода**
- **Поиск клонов кода: ошибки, нарушения лицензии**
- **Дедуктивный анализ моделей безопасности, PLRDF**

Тестиро-вание

- **Контролируемое выполнение на базе эмулятора QEMU**
- **Crusher (ISP Fuzzer и SyDr): фаззинг-тестирование и динамическое символьное исполнение, расширенное тестирование, Anis,**
- **Блесна: выявление утечек чувствительных данных в памяти**

Выпуск

- **Безопасный компилятор SafeC**
  - Диверсификация кода
  - ИСП–Обфускатор: защита кода от анализа

Реагиро-вание

- **Непрерывный поиск дефектов в выпущенном ПО**
- **Анализ аварийных завершений для оценки критичности программных дефектов**

Требования ФСТЭК к процессам разработки доверенного ПО (сертифицируемым программным средствам).

- **ГОСТ Р 56939-2016** «Защита информации. Разработка безопасного программного обеспечения. Общие требования»
  Утверждён и введен в действие **01.06.2016**
- **Федеральный закон от 26.07.2017 г. № 187-ФЗ** «О безопасности критической информационной инфраструктуры Российской Федерации»
- «**Методика выявления уязвимостей** и недекларированных возможностей в программном обеспечении» утверждена ФСТЭК России 11 февраля 2019 г.
- и новые проекты стандартов

*) - РБПО – Разработка Безопасного Программного Обеспечения

Инструменты РБПО, использующие ФМ

ИСП РАН

Разработка

- **Svace и Binside: статический анализ исходного и бинарного кода**
- Поиск клонов кода: ошибки, нарушения лицензии
- **Дедуктивный анализ моделей безопасности, PLRDF**

Тестирование

- Контролируемое выполнение на базе эмулятора QEMU
- **Crusher (ISP Fuzzer и SyDr): фаззинг-тестирование и динамическое символьное исполнение, расширенное тестирование, Anis,**
- Блесна: выявление утечек чувствительных данных в памяти

Выпуск

- Безопасный компилятор SafeC
  - Диверсификация кода
  - ИСП–Обфускатор: защита кода от анализа

Реагирование

- Непрерывный поиск дефектов в выпущенном ПО
- Анализ аварийных завершений для оценки критичности программных дефектов

Требования ФСТЭК к процессам разработки доверенного ПО (сертифицируемым программным средствам).

- **ГОСТ Р 56939-2016** «Защита информации. Разработка безопасного программного обеспечения. Общие требования»
  Утверждён и введен в действие **01.06.2016**
- **Федеральный закон от 26.07.2017 г. № 187-ФЗ** «О безопасности критической информационной инфраструктуры Российской Федерации»
- «**Методика выявления уязвимостей** и недекларированных возможностей в программном обеспечении» утверждена ФСТЭК России 11 февраля 2019 г.
- и новые проекты стандартов

ГОСТ Р 15408-2013/2014 Критерии оценки безопасности ИТ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 15408-1— 2012

Информационная технология

МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ.
КРИТЕРИИ ОЦЕНКИ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Часть 1

Введение и общая модель

ISO/IEC 15408-1:2009
Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model

(IDT)

Издание официальное

Москва
Стандартинформ
2014

ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 15408-2— 2013

Информационная технология

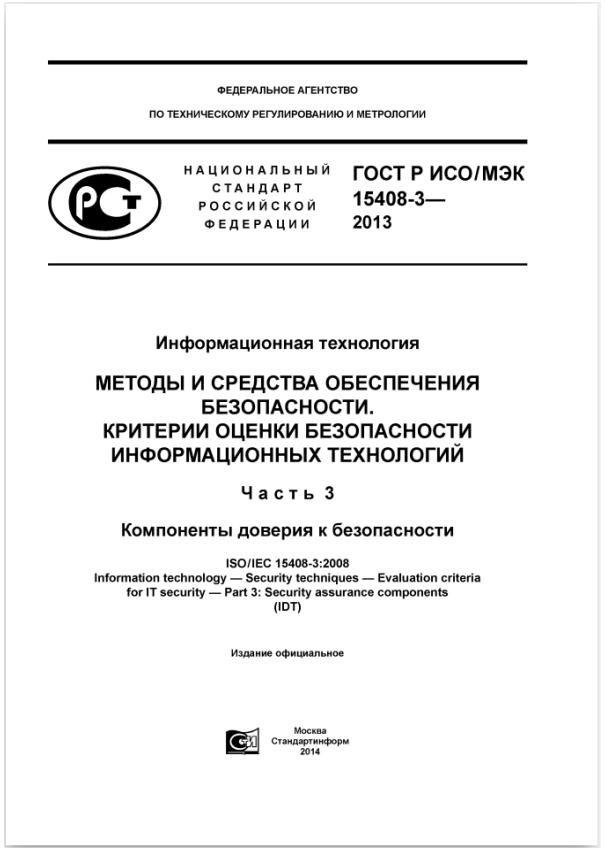
МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ.
КРИТЕРИИ ОЦЕНКИ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Часть 2 Функциональные компоненты безопасности

ISO/IEC 15408-2:2008
Information technology — Security techniques — Evaluation criteria for IT security — Part 2. Security functional components
(IDT)

Издание официальное

Москва
Стандартинформ
2014



ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 15408-3— 2013

Информационная технология

МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ.
КРИТЕРИИ ОЦЕНКИ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Часть 3

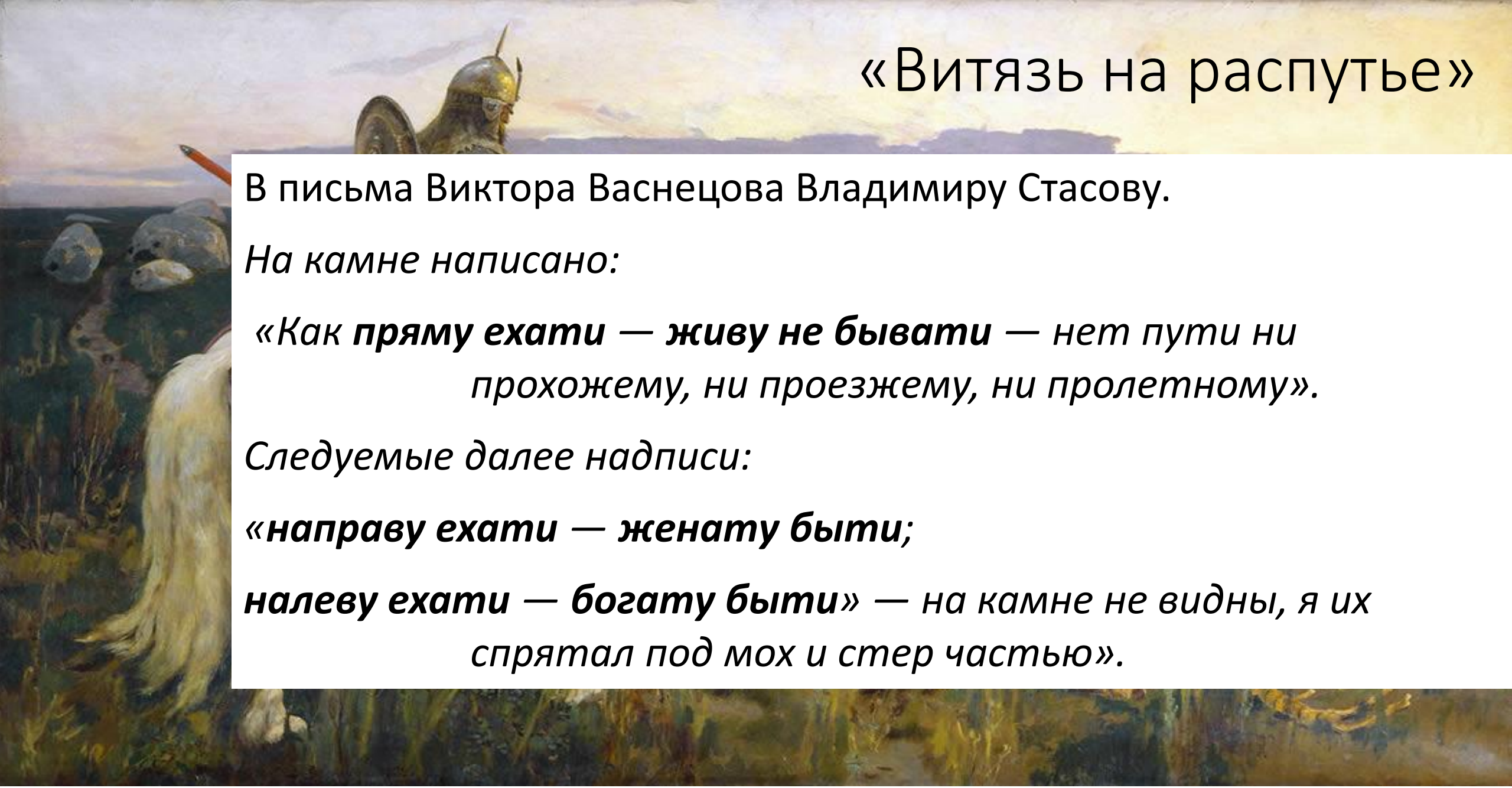
Компоненты доверия к безопасности

ISO/IEC 15408-3:2008
Information technology — Security techniques — Evaluation criteria for IT security — Part 3: Security assurance components
(IDT)

Издание официальное

Москва
Стандартинформ
2014

Горизонт 2030



«Витязь на распутье»

В письма Виктора Васнецова Владимиру Стасову.

На камне написано:

*«Как **пряму ехати — живу не бывати** — нет пути ни прохожему, ни проезжему, ни пролетному».*

Следуемые далее надписи:

*«**направу ехати — женату быти**;*

***налеву ехати — богату быти**» — на камне не видны, я их спрятал под мох и стер частью».*

Горизонт 2030. Три пути

ИСП РАН

- ***пряму ехати*** –Model Driven Engineering (программирование "сверху-вниз" - верификация моделей без верификации реальных программ)*;*
- ***направу ехати*** — небольшие, критически важные системы и методы их верификации;
- ***налеву ехати —*** интегрировать методы формальной верификации в низкоуровневые инструменты анализа программ и тестирование на основе формальных моделей.

Налеву ехати

ИСП РАН

- ***пряму ехати*** – интегрировать методы формальной верификации в манере Model Driven Engineering (программирование "сверху-вниз");
- ***направу ехати*** — небольшие, критически важные системы и методы их верификации;
- ***налеву ехати* — интегрировать** методы формальной верификации в низкоуровневые инструменты анализа программ и тестирование на основе формальных моделей (**MBT**)

  - Статическая верификация (software model checking) – **50 млн строк** на Си, формальные спецификации правил корректности (например, safety rules) – **CPAchecker, CPAlocator**
  - Генерация тестовых данных со сложными семантическими зависимостями (тестирование компиляторов, виртуальных машин, веб-приложений и проч.) – платформы на основе **dependent data types (Idris)**
  - Интеграция **тестирования на основе моделей** с инструментами фаззинг-тестирования на базе функциональных спецификаций

Направу ехати

ИСП РАН

- ***пряму ехати*** – интегрировать методы формальной верификации в манере Model Driven Engineering (программирование "сверху-вниз");
- ***направу ехати*** — небольшие, критически важные системы и методы их верификации;
  - Классическая дедуктивная верификация (**Isabelle/HOL, Coq, PLRDF**). Системы около **100 тыс. строк на Си**, с ограничениями параллельности и многопоточности – **20 чел.*лет**.
- ***налеву ехати*** — интегрировать методы формальной верификации в низкоуровневые инструменты анализа программ и тестирование на основе формальных моделей.

ГОСТ Р 59453-2021/2025 Формальная модель управления доступом. Части 1-4.

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 59453.1— 2021

Защита информации

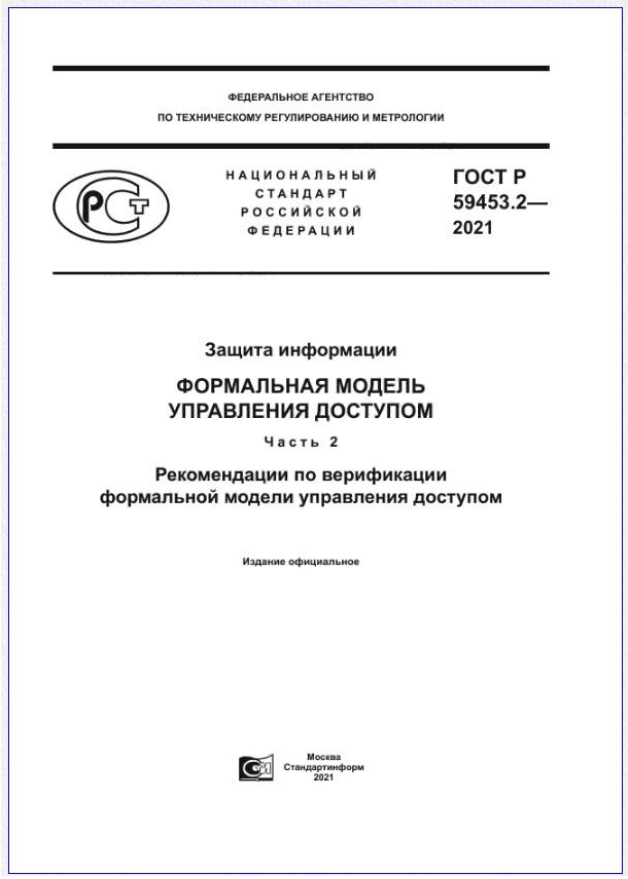
ФОРМАЛЬНАЯ МОДЕЛЬ УПРАВЛЕНИЯ ДОСТУПОМ

Часть 1

Общие положения

Издание официальное

Москва
Стандартинформ
2021



ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 59453.2— 2021

Защита информации

ФОРМАЛЬНАЯ МОДЕЛЬ УПРАВЛЕНИЯ ДОСТУПОМ

Часть 2

Рекомендации по верификации формальной модели управления доступом

Издание официальное

Москва
Стандартинформ
2021

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 59453.3— 2025

Защита информации

ФОРМАЛЬНАЯ МОДЕЛЬ УПРАВЛЕНИЯ ДОСТУПОМ

Часть 3

Рекомендации по разработке

Издание официальное

Москва
Российский институт стандартизации
2025

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 59453.4— 2025

Защита информации

ФОРМАЛЬНАЯ МОДЕЛЬ УПРАВЛЕНИЯ ДОСТУПОМ

Часть 4

Рекомендации по верификации средства защиты информации, реализующего политики управления доступом, на основе формализованных описаний модели управления доступом

Издание официальное

Москва
Российский институт стандартизации
2025

Пряму ехати

ИСП РАН

- ***пряму ехати*** –Model Driven Engineering (программирование "сверху-вниз" - верификация моделей без верификации реальных программ)*;*
  - Конструктивная информационная безопасность (КИБ) – Secure-by-Design
- ***направу ехати*** — небольшие, критически важные системы и методы их верификации;
- ***налеву ехати —*** интегрировать методы формальной верификации в низкоуровневые инструменты анализа программ и тестирование на основе формальных моделей.

Пряму ехати / КИБ

ИСП РАН

Принципы КИБ*)

- ❑ Минимизируйте поверхность атаки
- ❑ Установите безопасные значения по умолчанию
- ❑ Принцип наименьших привилегий
- ❑ Принцип эшелонированной защиты
- ❑ Безопасная обработка аварий
- ❑ Не доверяйте внешним сервисам
- ❑ Разделение обязанностей
- ❑ Избегайте безопасности через неизвестность
- ❑ Сохраняйте безопасность простой
- ❑ Правильно исправляйте дефекты в средствах безопасности

---

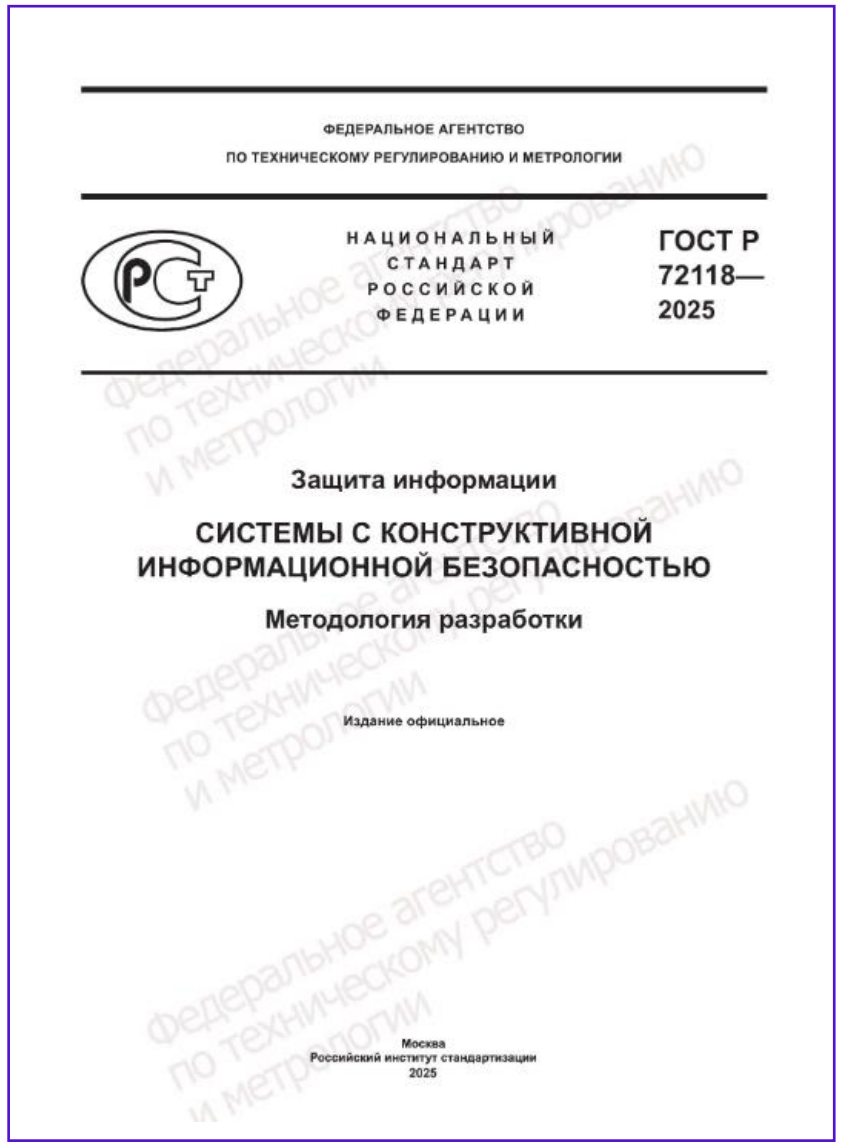
* - Secure by Design от Open Worldwide Application Security Project (OWASP) (https://owasp.org/www-project-secure-by-design-framework/)

7 принципов КИБ

ИСП РАН

Принцип изоляции
Принцип минимизации поверхности атаки
Принцип минимизации привилегий
Принцип эшелонированной
Принцип обеспечения безопасности по умолчанию
Принцип обеспечения безопасности сбоев
Принцип прозрачности решений по защите

Пряму ехати / КИБ



ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 72118— 2025

Защита информации

СИСТЕМЫ С КОНСТРУКТИВНОЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТЬЮ

Методология разработки

Издание официальное

Москва
Российский институт стандартизации
2025

Техники и методы КИБ

- Архитектурные шаблоны проектирования
  - Микроядерные ОС, виртуализация, ядра разделения, MILS, FLASK,…
- Систематизированный hardening
- Безопасные языки программирования (memory safe, secure language, функциональные языки)
- Архитектурные модели, средства верификации и анализа моделей, средств тестирования реализаций на соответствие моделям
  - Разработка и тестирование на основе моделей (MDE, MDSE, MBT), поиск архитектурных ошибок
- Репозитории архитектурных ошибок проектирования

ИИ и формальные методы

ИСП РАН

A Survey on Deep Learning for Theorem Proving

Zhaoyu Li[1], Jialiang Sun[1], Logan Murphy[1], Qidong Su[1], Zenan Li[2], Xian Zhang[3]
Kaiyu Yang[4]*, Xujie Si[1,5]
[1]University of Toronto, [2]Nanjing University, [3]Microsoft Research Asia, [4]Meta FAIR,
[5]CIFAR AI Chair
{zhaoyu, six}@cs.toronto.edu

Abstract

Theorem proving is a fundamental aspect of mathematics, spanning from informal reasoning in natural language to rigorous derivations in formal systems. In recent years, the advancement of deep learning, especially the emergence of large language models, has sparked a notable surge of research exploring these techniques to enhance the process of theorem proving. This paper presents a comprehensive survey of deep learning for theorem proving by offering (i) a thorough review of existing approaches across various tasks such as autoformalization, premise selection, proofstep generation, and proof search; (ii) an extensive summary of curated datasets and strategies for synthetic data generation; (iii) a detailed analysis of evaluation metrics and the performance of state-of-the-art methods; and (iv) a critical discussion on the persistent challenges and the promising avenues for future exploration. Our survey aims to serve as a foundational reference for deep learning approaches in theorem proving, inspiring and catalyzing further research endeavors in this rapidly growing field. A curated list of papers is available at https://github.com/zhaoyu-li/DL4TP.

1 Introduction

Proving theorems is a cornerstone of mathematics. Since the era of Euclid around 300 B.C., people have crafted theorems and proofs using a blend of natural language and mathematical symbols, meticulously evaluating their correctness through manual inspection. In the 1950s, a paradigm shift occurred with the exploration of computer-assisted proofs (Davis, 1957; Davis & Putnam, 1960), wherein a machine automatically applies deduction rules to prove assertions. These innovations laid the groundwork for the subsequent development of interactive theorem provers (Bruijn, de, 1970; Milner, 1972), enabling people to construct more intricate theorems and proofs by interacting with these systems. Building upon these advancements, later research extended the scope of theorem proving beyond mathematics, applying it to various practical applications such as software verification (Schumann, 2001) and hardware design (Kern & Greenstreet, 1999).

Exploring learning-based approaches for theorem proving has been a long-standing research focus, dating back to the 1990s (Suttner & Ertel, 1990; Denzinger et al., 1999). The recent development of deep learning, especially with the evolution of large language models (LLMs), has ignited a wave of research interest in this area again. As shown in Figure 1, the volume of papers on deep learning for theorem proving has grown approximately from 2 in 2016 to 45 in 2023, continuing to rise in 2024. However, despite such remarkable growth, this domain is characterized by a wide range of tasks, methods, datasets, and evaluations, which lack a cohesive framework to comprehend the true extent of progress and identify the underlying challenges and potential future work.

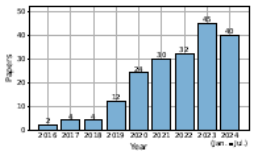


Figure 1: Papers on deep learning for theorem proving over the years (data for 2024 is up to July).

*Research conducted while Kaiyu Yang was at Caltech.

1

Сценарии использования ИИ в рамках ФМ

- ❑ Генерация спецификаций
- ❑ Генерация программ
- ❑ Генерация тестов
- ❑ Поиск клонов
- ❑ Поиск некорректного заимствования
- ❑ Анализ результатов статического и динамического анализа
- ❑ Объяснение программ
- ❑ Объяснение результатов тестирования
- ❑ Поиск причин ошибочного повеления программ (root cause analysis)
- ❑ Ассистирование при верификации программ (доказательстве теорем)
- ❑ . . .

Спасибо!