universität freiburg

Sur les assistants de preuve

Prof. Dr. Stefan Kebekus
Freiburg, 29. September 2025

universität freiburg

Sur les assistants de preuve ...

... for the Working Mathematician

Prof. Dr. Stefan Kebekus
Freiburg, 29. September 2025

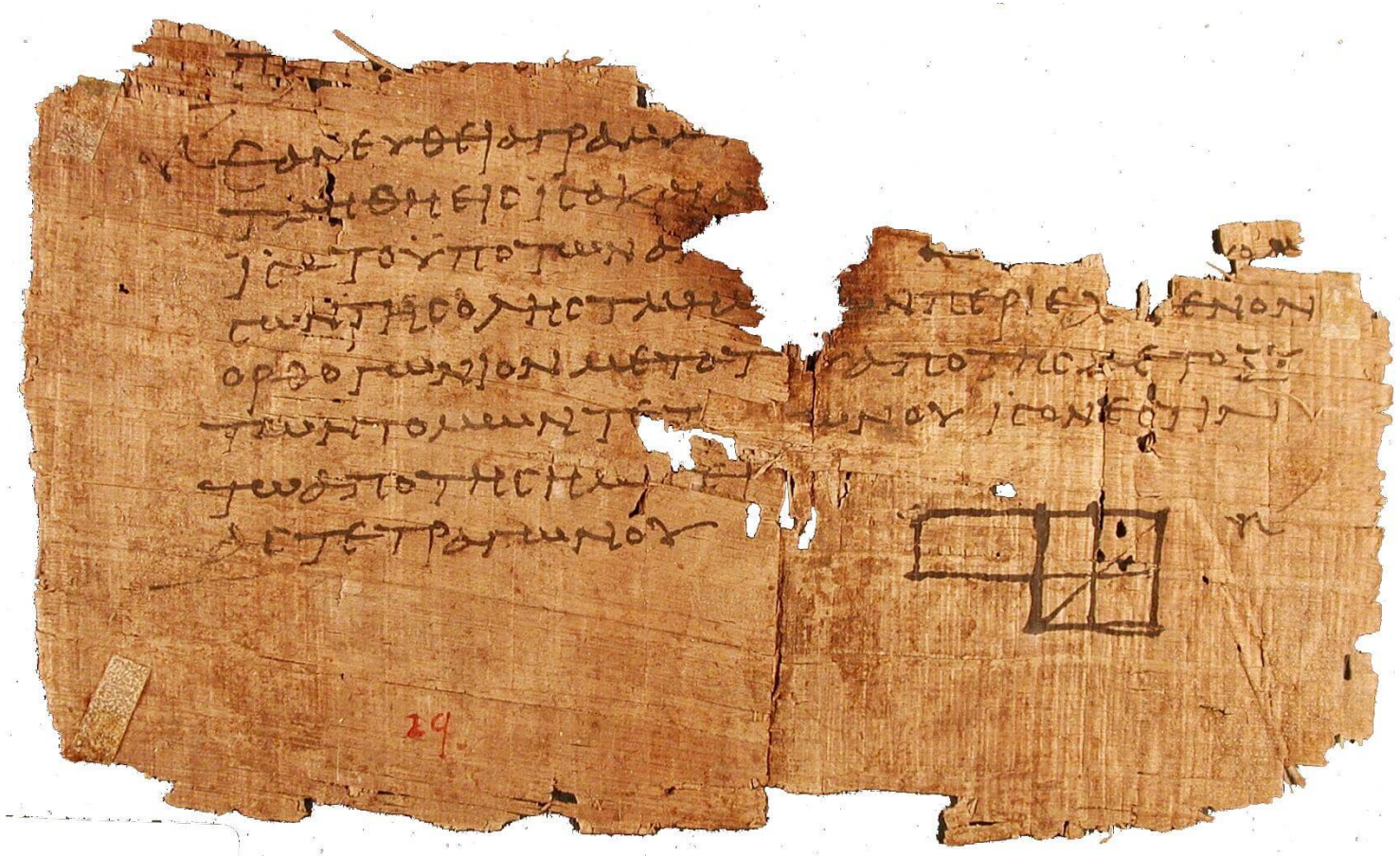
Agenda

1. Background
2. Practical Example
3. State of the Art
4. Why Formalize Mathematics?
5. Outlook

Background

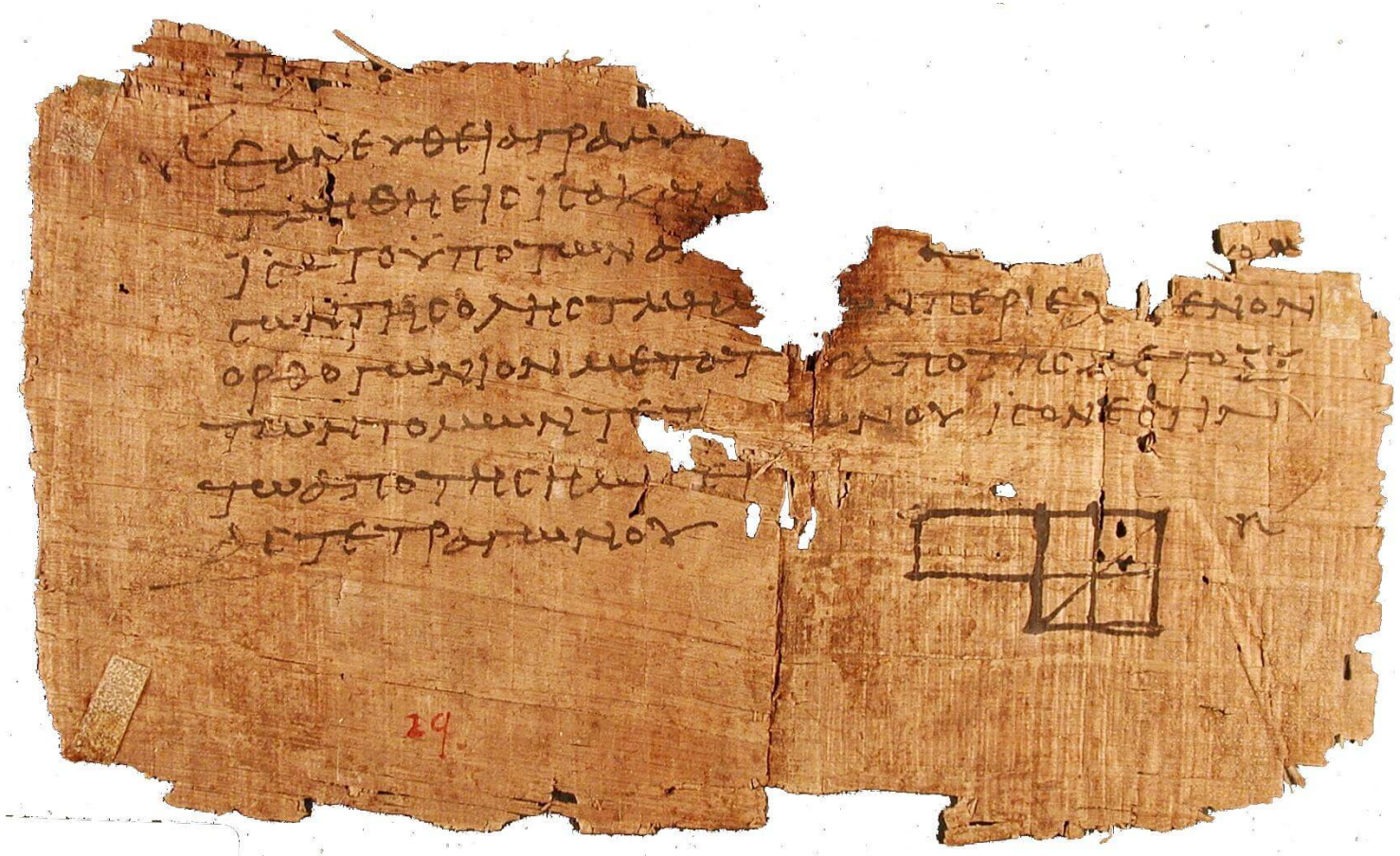
Background

Στοιχεῖα



Background

Euclid's „Elements“



Background
Euclid's „Elements“

- **Introduces the axiomatic method**
  - Specifies small number of axioms
  - Builds Mathematics from the axioms through logical inference
- **Widely recognized and taught as the „correct“ way to do mathematics**

Background
Euclid's „Elements“

- **Introduces the axiomatic method**
  - Specifies small number of axioms
  - Builds Mathematics from the axioms through logical inference
- **Widely recognized and taught as the „correct“ way to do mathematics**
- **Widely ignored by the working mathematician in practise**

Background
Principia Mathematica

$*54{\cdot}43. \quad \vdash :. \alpha, \beta \in 1 . \supset : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54{\cdot}26 . \supset \vdash :. \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

$[*51{\cdot}231] \qquad \equiv . \iota`x \cap \iota`y = \Lambda .$

$[*13{\cdot}12] \qquad \equiv . \alpha \cap \beta = \Lambda \qquad (1)$

$\vdash . (1) . *11{\cdot}11{\cdot}35 . \supset$

$\vdash :. (\exists x, y) . \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda \qquad (2)$

$\vdash . (2) . *11{\cdot}54 . *52{\cdot}1 . \supset \vdash . \text{Prop}$

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

Background
Curry-Howard Correspondence

There is a direct relationship between logical propositions and types in programming languages, and between mathematical proofs and computer programs.

Core Idea:

- Propositions correspond to types in programming.
- Proofs of a proposition correspond to programs that construct elements of the type corresponding to the proposition.
- Proving a proposition corresponds to writing a program.

Background
Curry-Howard Correspondence

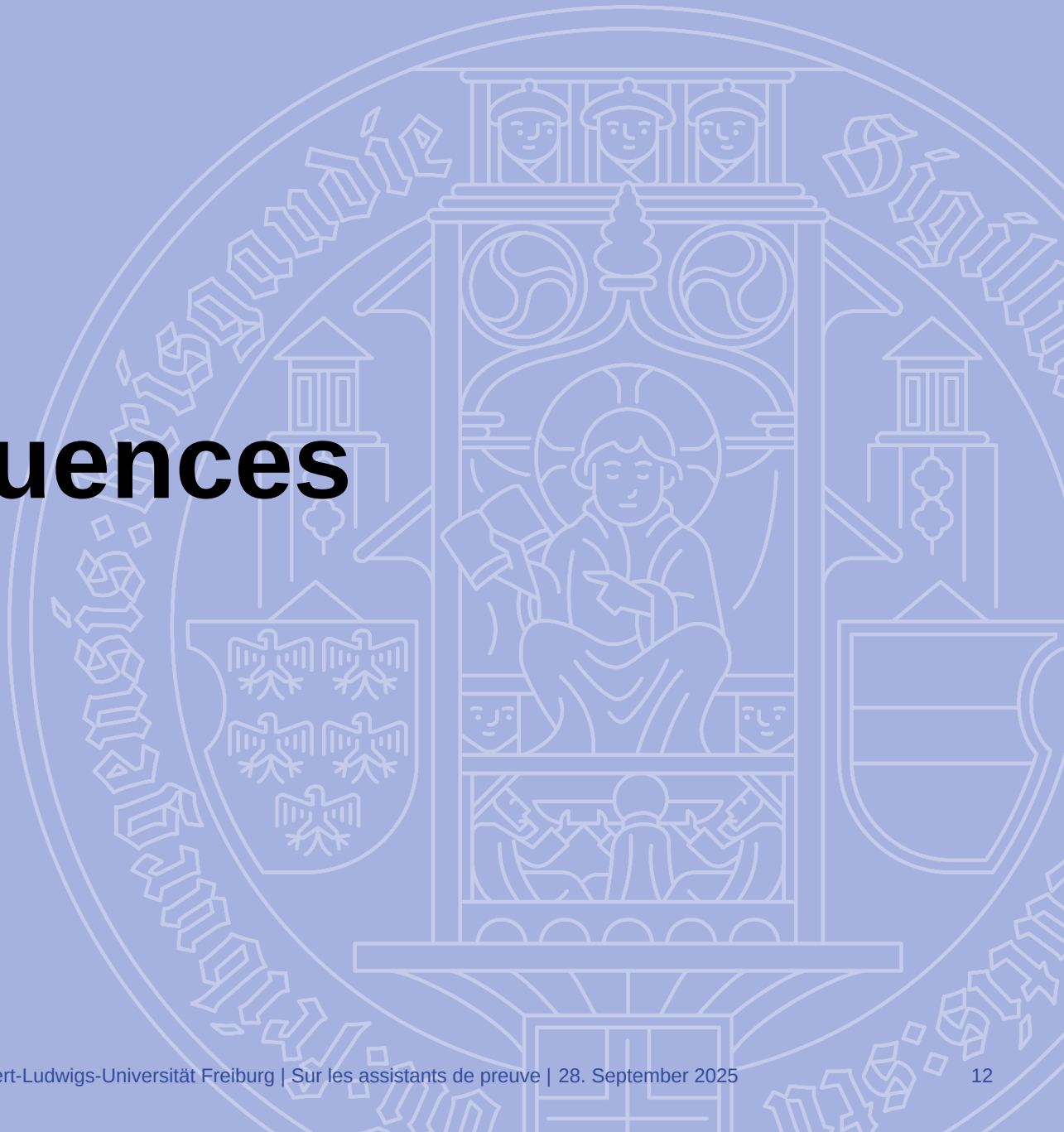
There is a direct relationship between logical propositions and types in programming languages, and between mathematical proofs and computer programs.

Want a programming language where …

- types already „look like“ their corresponding propositions in standard mathematical language, and
- programs already „look like“ their corresponding proofs in standard mathematical language.

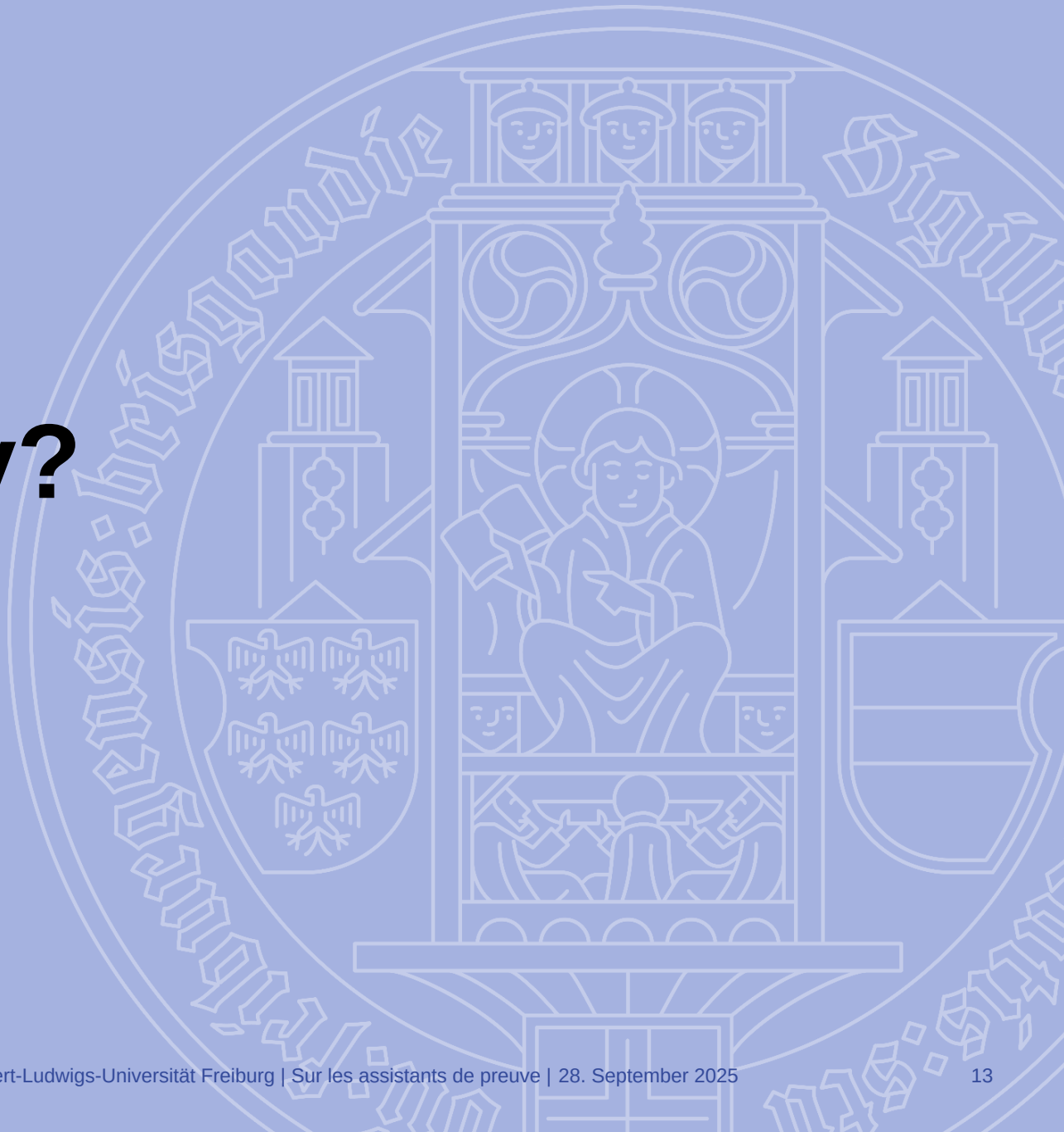
Key Point: Computers can check if source code contains a valid program!

Practical Example
Continuity and Sequences



State of the Art

Where are we today?



State of the Art
Formalization Projects / Research

Reference Projects

- Thomas Hales: Flyspec / Kepler Conjecture (Completed 2015)
- Johan Commelin: Liquid Tensor Experiment / Condensed Mathematics (Completed 2021)
- Kevin Buzzard: Fermat‘s Last Theorem (Ongoing)

Other

- Smale Sphere Eversion / Independence of the Continuum Hypothesis / Banach-Tarski Paradox

Massive push by high-profile mathematicians including Timothy Govers, Terence Tao, Akshay Venkatesh, Maryna Viazovska, …

State of the Art
Mathlib

Goal: Bourbaki in the 21st Century / Building the Mathematics Library of the Future

- Base project of most formalization efforts today
- Huge community / 500+ Contributors
- Math coverage presently uneven, but entire curriculum of Imperial College London is within reach

State of the Art
Formalization Projects / Own Experience

Goal:

- Formalize Value Distribution Theory
- Contribute the resulting code to Mathlib

Experience:

- Steep learning curve, but excellent tutorials available
- Extremely helpful and welcoming community
- Tedious work, but no fundamental problems

State of the Art
Education

Integration into the curriculum

- Discussed and (partially) implemented in numerous institutions all over the globe
- Good ressources and materials become available
- Naive implementation likely to overwhelm students

Formalization as a thesis project

- At Imperial College London: 20-30% of all MA/BA thesis projects are concerned with formalization

State of the Art
Governance and Development

LEAN

- Open Source / Apache License
- Initially developed by Microsoft Research
- Today: Development through FRO financed by US and British Philantropists

Mathlib

- Open Source / Apache License
- Development overseen by small group of extremely competent volunteer maintainers
- Developmend supported by the Mathlib Foundation, financed by US and British Philantropists
- Infrastructure financed by GitHub Inc. / Microsoft

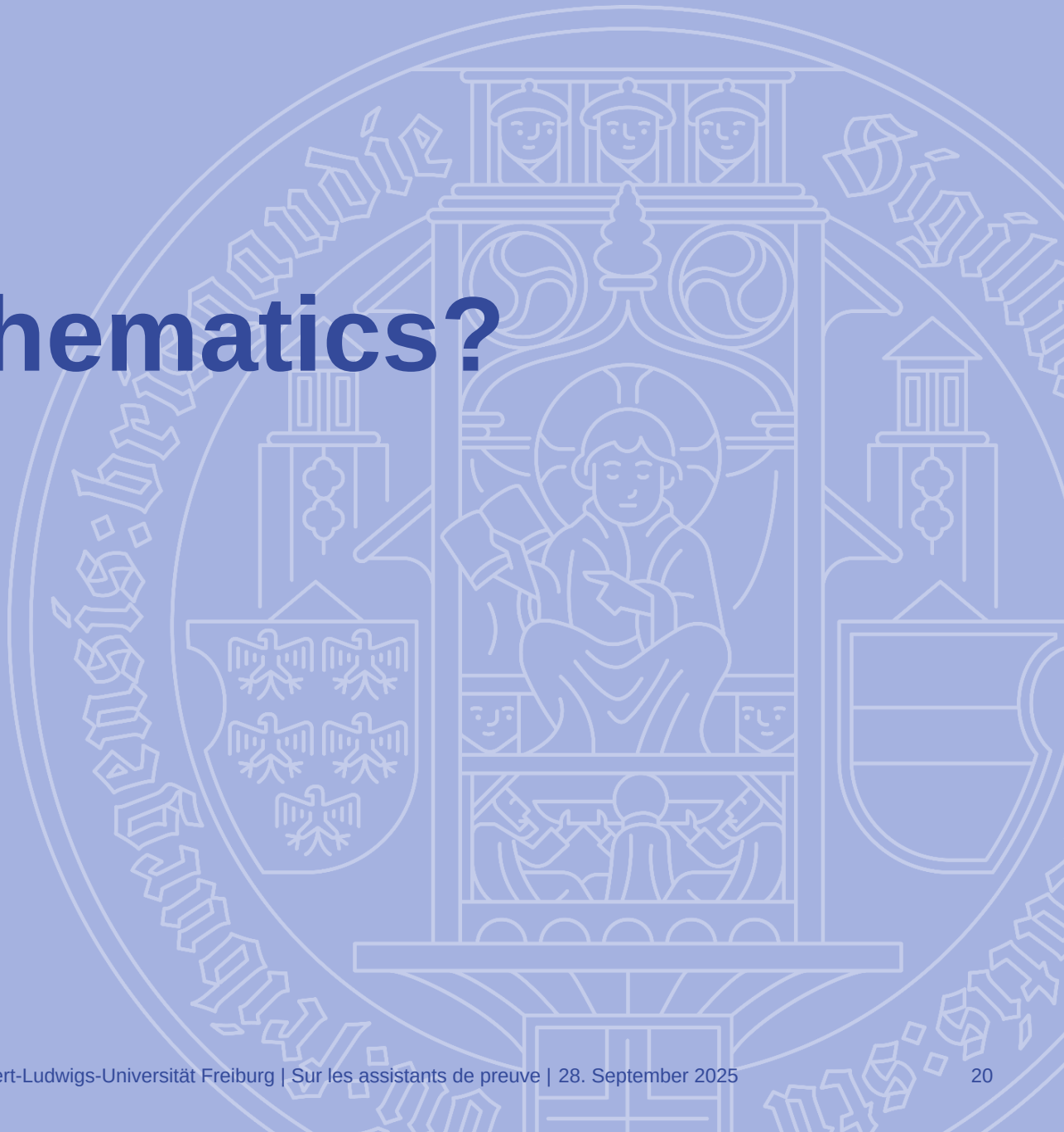
State of the Art
Artificial Intelligence / Automatic Formalization

- Performance currently underwhelming, not good enough for daily use
- Overblown claims targeted at potential investors, substantial marketing hyperbole
- Real improvements in the last two years

Massive Investments

- US Cooperations: OpenAI / Microsoft / Google / Meta / …
- US Venture Capital via AI Startups
- DARPA
- US and British Philantropists

Why Formalize Mathematics?



Why Formalize Mathematics?

- **For fun!**
- **To make knowledge available!**
- **To preserve knowledge!**

- **As a direct tool in research.**
- **To verify proofs.**

Outlook

Where are we going?

Thank you for your time!