Summer School: Proofs, arguments and dialogues:
history, epistemology and logic of justification practices.
University of Tübingen, 10/8/2022

Automated Reasoning and Proof Assistants for Mathematics

Angeliki Koutsoukou-Argyraki

Department of Computer Science and
Technology
(Computer Laboratory)
University of Cambridge, UK

UNIVERSITY OF
CAMBRIDGE

Supported by the ERC Advanced Grant ALEXANDRIA, Project 742178

erc
European Research Council
Established by the European Commission

Automated: algorithmic, automatic, interactive (between user and machine).

Reasoning: finding new conclusions from premises (axioms, definitions, assumptions) using rules of inference.

Plan of this course:

(PART A) A general introduction on proof assistants (interactive theorem provers) focussing on their use for the formalisation of mathematics; a discussion on the state of the art and recent advances. (Focus on Isabelle/HOL).

(PART B) Formalisation of Aristotle’s Assertoric Syllogistic in Isabelle/HOL.

A bit of history

Leibniz (1666)

“Dissertatio de arte combinatoria”: proposes the development of a symbolic language that could express any rational thought (characteristica universalis) and a mechanical method to determine its truth (calculus ratiocinator). To resolve any dispute: “Let us calculate!”/ “Calculemus!”

Boole (1847)

“The mathematical analysis of logic”: propositional logic.

Frege (1879)

“Begriffsschrift”: an expressive formal language equipped with logical axioms and rules of inference.

A bit of history

Whitehead and Russell (1910-1913)

“Principia Mathematica”: (logicism) goal to express all mathematical propositions in symbolic logic & solve paradoxes of set theory.Developed type theory.

Hilbert (1920)

Formalism and Hilbert’s program: All mathematical statements should be written in a precise formal language, follow from a provably consistent finite system of axioms, according to well-defined rules. Completeness, Consistency, Conservation, Decidability.

Note: Gödel’s Incompleteness Theorems (1931)

A bit of history

de Bruijn (late 1960s)

AUTOMATH: a predecessor of modern proof assistants based on type theory. Used Curry–Howard correspondence. Late 1970’s: van Benthem Jutting translated Landau’s “Foundations of Analysis” into AUTOMATH.

The QED Manifesto (1994)

A proposal for a central computer-based library of all known mathematics fully formalised and formally verified (automatically checked by computers)

The project was soon abandoned.

(Or was it?)

Today

Modern proof assistants (interactive theorem provers)

Software tools for formal verification/ the development of formal proofs by user-computer interaction. A human user writes the proof in a formal language via an interactive interface to be checked by a computer. Intermediate proof steps are often given by automation.

A variety of proof assistants available, based on different logical formalisms:
Based on: set theory (e.g. Mizar, Metamath); simple type theory (e.g. HOL4, HOL Light, Isabelle); dependent type theory (e.g. Coq, Agda,Lean, PVS).
Extensive libraries of formalised mathematics available.

For a direct comparison with examples, see, e.g. the webpage maintained by Wiedijk, “Formalising 100 theorems”.

“We believe that when later generations look back at the development of mathematics one will recognise four important steps:

(1) the Egyptian-Babylonian-Chinese phase, in which correct computations
were made, without proofs;
(2) the ancient Greeks with the development of “proof”;
(3) the end of the nineteenth century when mathematics became “rigorous”;
(4) the present, when mathematics (supported by computer) finally becomes
fully precise and fully transparent.”

Barendregt, H. and Wiedijk, F. (The challenge of computer mathematics, Philos. Trans. - Royal Soc., Math. Phys. Eng. Sci. 36(1835):2351-2375 (2005)).

Why formalise mathematics?

* Verification: Mathematicians can be fallible. (Example: the Fields medalist Vladimir Voevodsky started working in formalisation after discovering errors in his own work).

* (Future of?) Reviewing.

* Preserving mathematical knowledge in big libraries of formalised mathematics: databases with an enormous potential for the creation of future AI tools to assist mathematicians in the discovery(/invention) of new results.

* Deeper understanding, new insights: even familiar material can be seen in new light when using new tools. High level of detail in which a formalised proof must be written forces to think and rethink proofs and definitions.

* Educational tools.

* Last but not least: it is fulfilling and fun!

Why formalise mathematics?

...and a comment on an additional personal motivation

Work in applied proof theory- proof mining: pen-and-paper extraction of constructive/quantitative information from proofs in the form of computable bounds (requiring a logical analysis of a proof and rewriting it to make the logical form of all the statements involved explicit via revealing the hidden quantifiers).

Provokes the question:

What is it that makes a “good” proof?

* a shorter proof;

* a more “elegant” proof;

* a simpler proof (consider Hilbert’s 24th problem (1900)): “find criteria for simplicity of proofs, or, to show that certain proofs are simpler than any others.”;

* in terms of Reverse Mathematics – a proof in a weaker subsystem of Second Order Arithmetic;

* an interdisciplinary proof (e.g. a geometric proof for an algebraic problem or vice-versa would be considered to give a deeper mathematical insight);

* a proof that is easier to reuse i.e. if it provides some algorithm or technique or intermediate result that can be useful in different contexts too;

* a proof giving “better” computational content.

What do we mean by “better” computational content?

* a bound of lower complexity?

* a bound that is more precise numerically?

* a bound that is more “elegant”?

Why formalise mathematics?
A vision for the future of research mathematics:

To create an interactive assistant that would help research mathematicians in their creative work by

* providing “brainstorming”/ hints:
proof recommendations, counterexamples, proofs of auxiliary lemmas/intermediate steps;
* suggesting conjectures;
* providing information on relevant literature results;
* helping with bookkeeping on the proof structure/proof goals and details;
* formally verifying the new results.

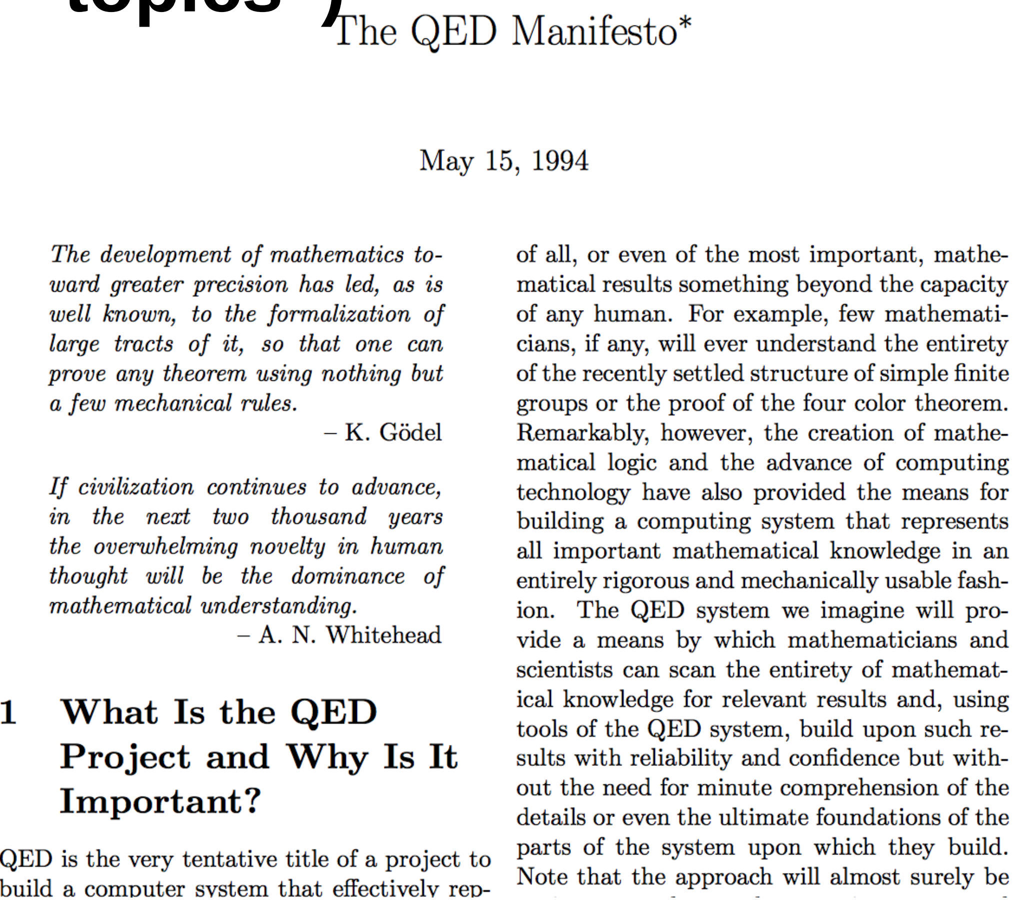
The goal is to assist mathematicians, not to replace them.

Why formalise mathematics?
A vision for the future of research mathematics:

Timothy Gowers (Fields Medal 1998) describes how a ”dialogue” between a user and a computer would ideally look like in the future to interactively assist the human mathematician to arrive at (new) conclusions. The computer would have access to an extensive database of mathematical material.

W.T. Gowers (2010). Rough Structure and Classification. In: Alon, N., Bourgain, J., Connes, A., Gromov, M., Milman, V. (eds) Visions in Mathematics. Modern Birkhäuser Classics. Birkhäuser Basel. https://doi.org/10.1007/978-3-0346-0422-2_4

Some more suggested reading (in addition to the material already given in “topics”)



The QED Manifesto*

May 15, 1994

The development of mathematics toward greater precision has led, as is well known, to the formalization of large tracts of it, so that one can prove any theorem using nothing but a few mechanical rules.

– K. Gödel

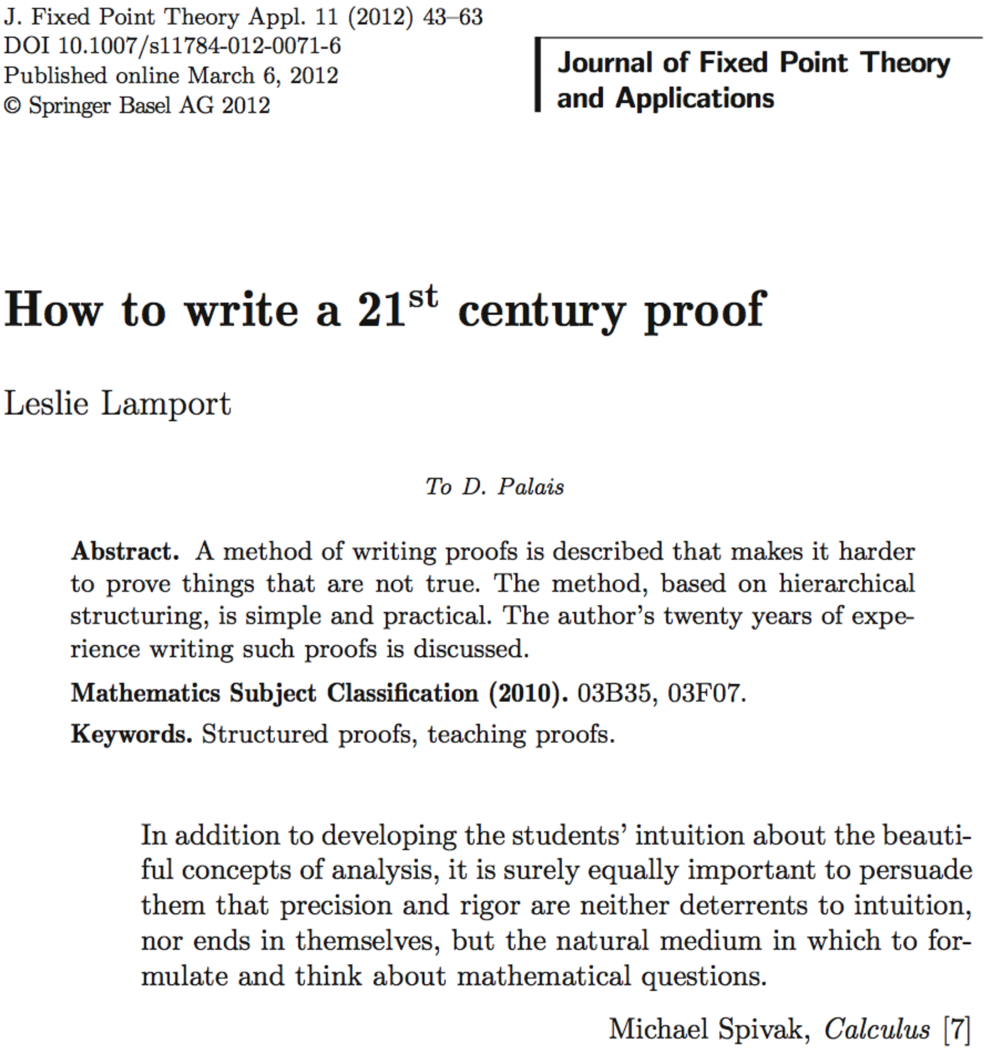
If civilization continues to advance, in the next two thousand years the overwhelming novelty in human thought will be the dominance of mathematical understanding.

– A. N. Whitehead

1 What Is the QED Project and Why Is It Important?

QED is the very tentative title of a project to build a computer system that effectively rep-

of all, or even of the most important, mathematical results something beyond the capacity of any human. For example, few mathematicians, if any, will ever understand the entirety of the recently settled structure of simple finite groups or the proof of the four color theorem. Remarkably, however, the creation of mathematical logic and the advance of computing technology have also provided the means for building a computing system that represents all important mathematical knowledge in an entirely rigorous and mechanically usable fashion. The QED system we imagine will provide a means by which mathematicians and scientists can scan the entirety of mathematical knowledge for relevant results and, using tools of the QED system, build upon such results with reliability and confidence but without the need for minute comprehension of the details or even the ultimate foundations of the parts of the system upon which they build. Note that the approach will almost surely be



J. Fixed Point Theory Appl. 11 (2012) 43–63
DOI 10.1007/s11784-012-0071-6
Published online March 6, 2012
© Springer Basel AG 2012

Journal of Fixed Point Theory and Applications

How to write a 21$^{\text{st}}$ century proof

Leslie Lamport

To D. Palais

Abstract. A method of writing proofs is described that makes it harder to prove things that are not true. The method, based on hierarchical structuring, is simple and practical. The author’s twenty years of experience writing such proofs is discussed.

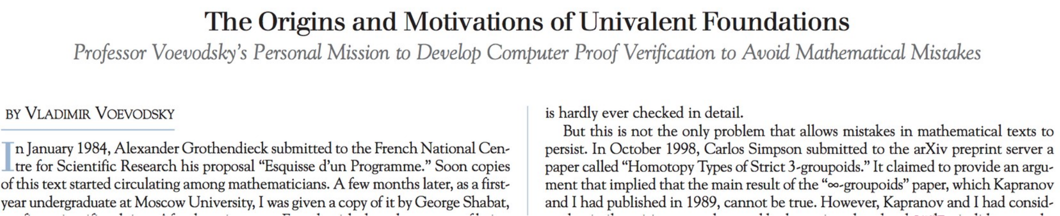
Mathematics Subject Classification (2010). 03B35, 03F07.

Keywords. Structured proofs, teaching proofs.

In addition to developing the students’ intuition about the beautiful concepts of analysis, it is surely equally important to persuade them that precision and rigor are neither deterrents to intuition, nor ends in themselves, but the natural medium in which to formulate and think about mathematical questions.

Michael Spivak, *Calculus* [7]

Some more suggested reading (in addition to the material already given in “topics”)



The Origins and Motivations of Univalent Foundations

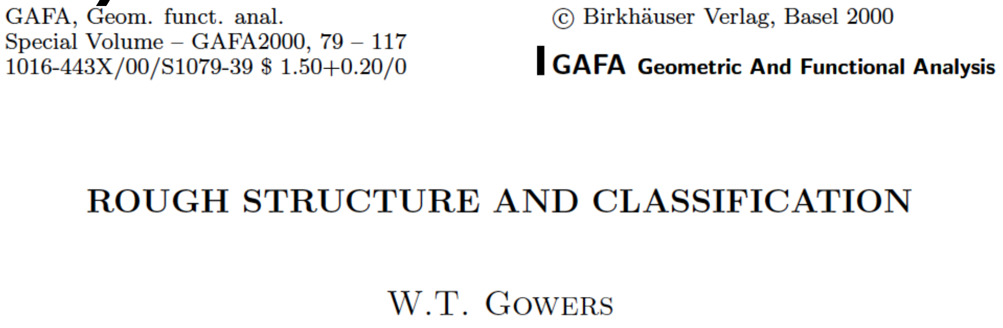
Professor Voevodsky’s Personal Mission to Develop Computer Proof Verification to Avoid Mathematical Mistakes

BY VLADIMIR VOEVODSKY

In January 1984, Alexander Grothendieck submitted to the French National Centre for Scientific Research his proposal “Esquisse d’un Programme.” Soon copies of this text started circulating among mathematicians. A few months later, as a first-year undergraduate at Moscow University, I was given a copy of it by George Shabat,

is hardly ever checked in detail.

But this is not the only problem that allows mistakes in mathematical texts to persist. In October 1998, Carlos Simpson submitted to the arXiv preprint server a paper called “Homotopy Types of Strict 3-groupoids.” It claimed to provide an argument that implied that the main result of the “∞-groupoids” paper, which Kapranov and I had published in 1989, cannot be true. However, Kapranov and I had consid-



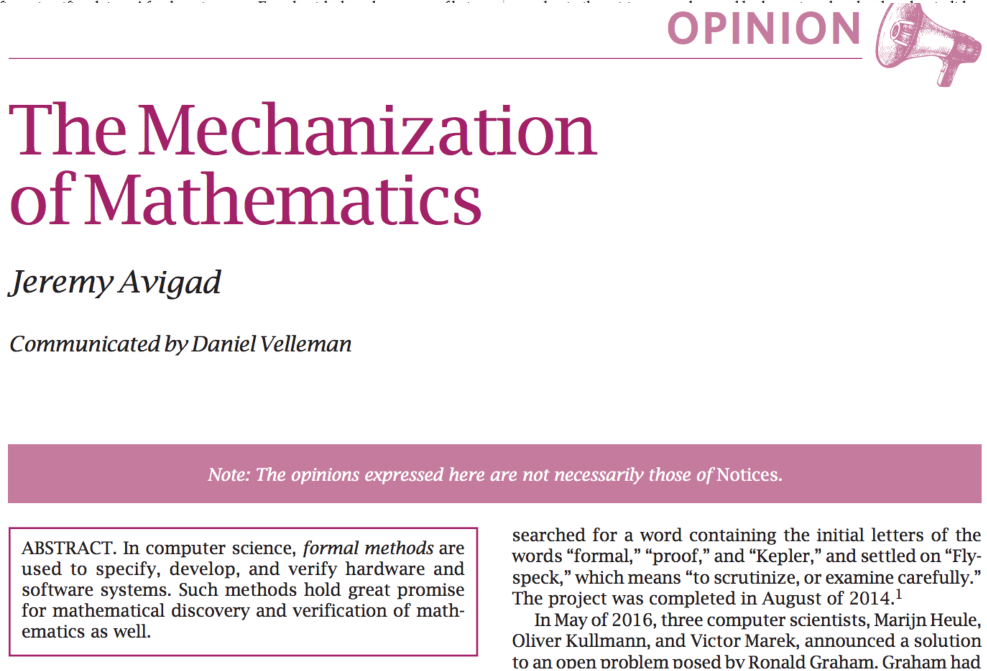
GAFA, Geom. funct. anal.
Special Volume – GAFA2000, 79 – 117
1016-443X/00/S1079-39 $ 1.50+0.20/0

© Birkhäuser Verlag, Basel 2000

GAFA Geometric And Functional Analysis

ROUGH STRUCTURE AND CLASSIFICATION

W.T. Gowers



OPINION

The Mechanization of Mathematics

Jeremy Avigad

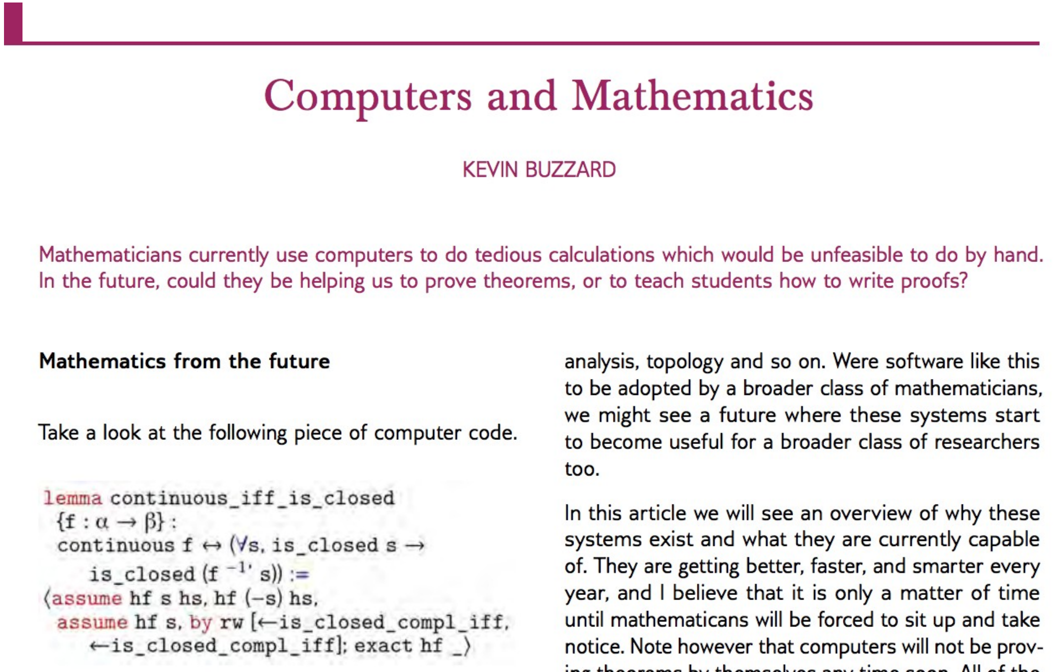
Communicated by Daniel Velleman

Note: The opinions expressed here are not necessarily those of Notices.

ABSTRACT. In computer science, *formal methods* are used to specify, develop, and verify hardware and software systems. Such methods hold great promise for mathematical discovery and verification of mathematics as well.

searched for a word containing the initial letters of the words “formal,” “proof,” and “Kepler,” and settled on “Flyspeck,” which means “to scrutinize, or examine carefully.” The project was completed in August of 2014.[1]

In May of 2016, three computer scientists, Marijn Heule, Oliver Kullmann, and Victor Marek, announced a solution to an open problem posed by Ronald Graham. Graham had



Computers and Mathematics

KEVIN BUZZARD

Mathematicians currently use computers to do tedious calculations which would be unfeasible to do by hand. In the future, could they be helping us to prove theorems, or to teach students how to write proofs?

Mathematics from the future

Take a look at the following piece of computer code.

```
lemma continuous_iff_is_closed
 {f : α → β} :
 continuous f ↔ (∀s, is_closed s →
   is_closed (f ⁻¹' s)) :=
⟨assume hf s hs, hf (−s) hs,
 assume hf s, by rw [←is_closed_compl_iff,
   ←is_closed_compl_iff]; exact hf _⟩
```

analysis, topology and so on. Were software like this to be adopted by a broader class of mathematicians, we might see a future where these systems start to become useful for a broader class of researchers too.

In this article we will see an overview of why these systems exist and what they are currently capable of. They are getting better, faster, and smarter every year, and I believe that it is only a matter of time until mathematicans will be forced to sit up and take notice. Note however that computers will not be proving theorems by themselves any time soon. All of the

Some milestones & recent advances

* Formalisation of the proof of the four-colour theorem in Coq by Gonthier (2008).

* Gonthier has also formalised the Feit–Thompson proof of the odd-order theorem in Coq (2012).

* Formalisation of the proof (1998 publ. 2005) by Hales of the Kepler conjecture (sphere packing problem) in HOL Light and Isabelle/HOL by Hales et al. (Flyspeck project, 2003-compl. 2014).

* Formalisation of Gödel's Incompleteness theorems in Isabelle/HOL by Paulson (2013).

Some milestones & recent advances

* Formalisation of an irrationality proof of ζ(3) by Apéry (evaluation of the Riemann zeta function) in Coq by Chyzak, Mahboubi, Sibut-Pinote & Tassi (2014).

* Verification of an algorithm with Isabelle/HOL to verify Tucker’s proof that the Lorenz attractor is chaotic in a rigorous mathematical sense by Immler (2015).

* Formalisation of Scholze’s perfectoid spaces in Lean by Buzzard, Commelin and Massot (2019).

* Grothendieck’s schemes in Lean by Buzzard, Hughes, Lau, Livingston, Fernández Mir, R., Morrison, S. (2020).
Independently in Isabelle/HOL by Bordg, Li and Paulson (2021).

Some milestones & recent advances

* Formalisation of a substantial amount of material in analytic number theory in Isabelle/HOL by Manuel Eberl (2019).

* The independence of the Continuum Hypothesis by Han & van Doorn in Lean (2021). Independently in Isabelle/ZF by Gunther, Pagano, Sánchez Terraf & Steinberg (2022).

* Formalisation of the solution to the cap set problem (Ellenberg & Gijswijt, 2017) by Dahmen, Hölzl and Lewis in Lean (2019).

* Szemerédi’s Regularity Lemma and Roth’s Theorem on Arithmetic Progressions in Isabelle/HOL by Edmonds, Koutsoukou-Argyraki and Paulson. Independently in Lean by Dillies and Mehta (2021).

Some milestones & recent advances

The Liquid Tensor Experiment

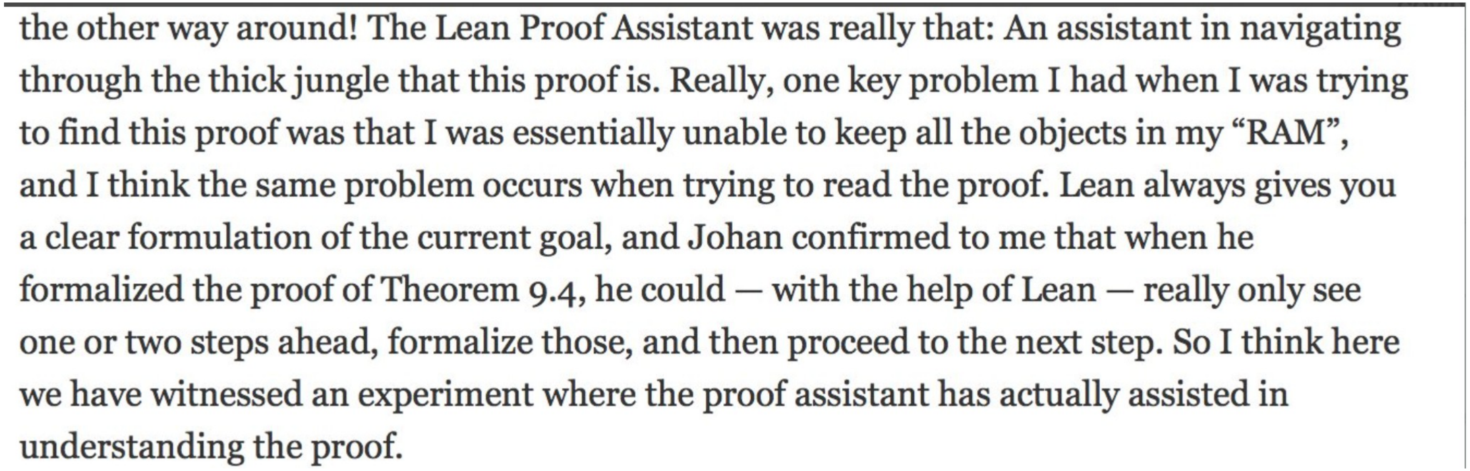
Condensed Mathematics is a theory by Clausen and Scholze (Fields Medal 2018) introducing condensed sets (an alternative notion to topological spaces).

In Dec. 2020, Scholze posed a challenge to the Xena Project Blog: to formalise the proof of a result of his he had doubts about.

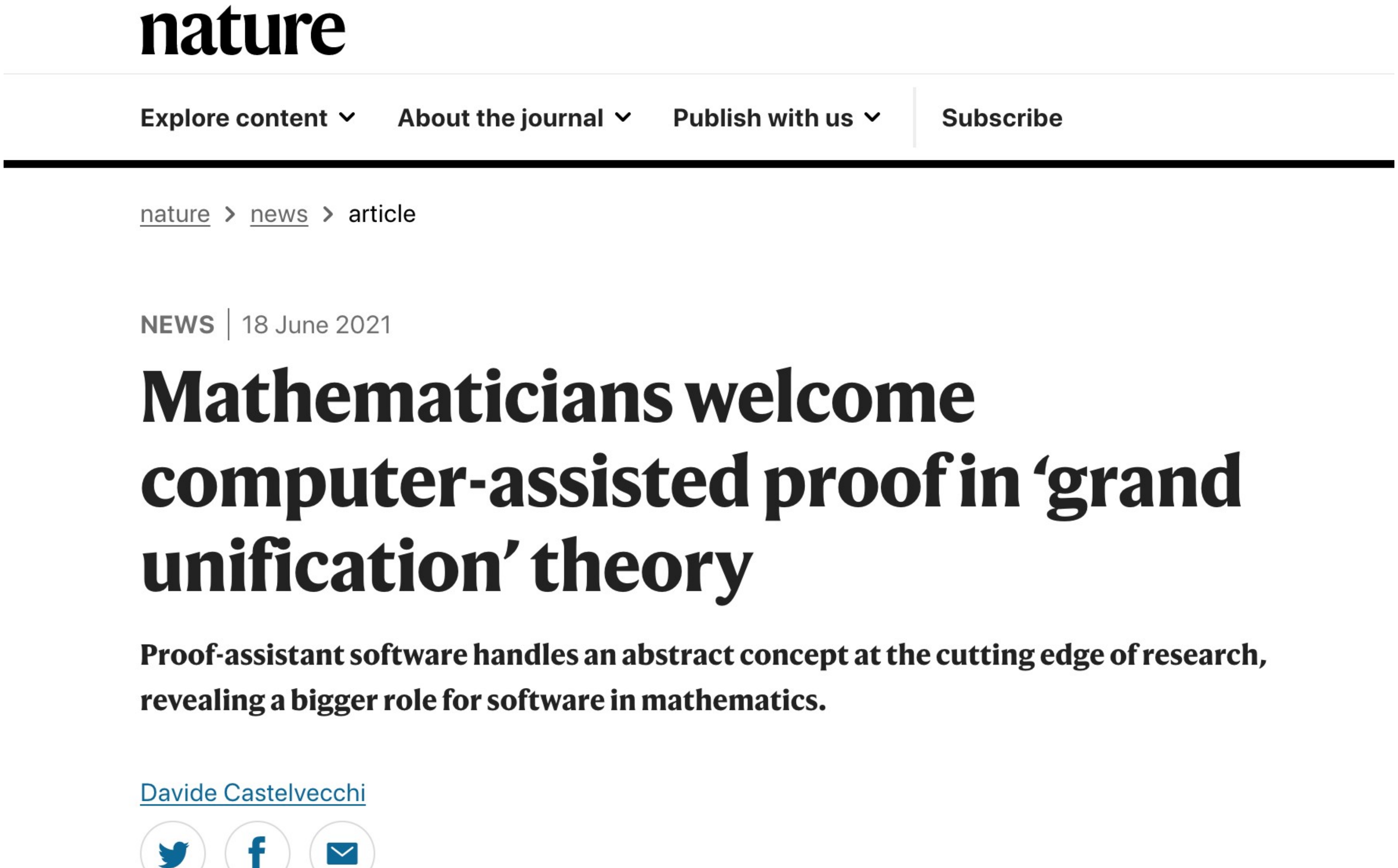
The Lean Prover Community took up the challenge: a huge collaborative effort led by Commelin succeeded to complete the proof in the summer of 2022.

Scholze had been reporting on the progress in subsequent Xena blogposts.

Scholze (June 2021, Xena Project Blog):



the other way around! The Lean Proof Assistant was really that: An assistant in navigating through the thick jungle that this proof is. Really, one key problem I had when I was trying to find this proof was that I was essentially unable to keep all the objects in my “RAM”, and I think the same problem occurs when trying to read the proof. Lean always gives you a clear formulation of the current goal, and Johan confirmed to me that when he formalized the proof of Theorem 9.4, he could — with the help of Lean — really only see one or two steps ahead, formalize those, and then proceed to the next step. So I think here we have witnessed an experiment where the proof assistant has actually assisted in understanding the proof.



nature

NEWS | 18 June 2021

Mathematicians welcome computer-assisted proof in ‘grand unification’ theory

Proof-assistant software handles an abstract concept at the cutting edge of research, revealing a bigger role for software in mathematics.

Davide Castelvecchi

Towards a new era in Mathematics?

A big shift: Formalisation was until recently an area of computer science. Now it is quickly attracting the interest of working mathematicians and mathematics students. Enthusiastic online communities and tools e.g. Zulip enable massive collaborative projects. Libraries of formal proofs are expanding at an increasingly high pace, day-by-day. Student-run projects are emerging too. Everyone welcome to join.

* The 2020 Mathematics Subject Classification includes for the first time subject classes on the formalisation of mathematics using proof assistants (68VXX).

* Kevin Buzzard, Professor at Imperial College London, an expert in arithmetic geometry and algebraic number theory who in 2017 launched the Xena project teaching undergraduate students to use the proof assistant Lean (with young mathematicians participating enthusiastically in increasing numbers) was an invited speaker at the 2022 International Congress of Mathematicians to talk about the formalisation of mathematics.

Main Obstacles

* Better automation is needed to provide proofs for intermediate proof steps (proofs are analysed in an extremely high level of detail).

* Efficient search features.

* Efficient organisation and management of libraries.

* Interoperability of proof systems, translation of proofs between proof assistants needed (Goals of the Dedukti System/ EuroProofNet COST Action).

AI/ machine learning and the future of research mathematics

Proof assistants and foundations are only one side of the story. Progress seems to require the combination of alternative approaches. An interesting analogy due to Georg Gottlob:

``rule knowledge and logical reasoning VS machine learning e.g. neural networks'' as

``left part of the brain VS right part of the brain''.

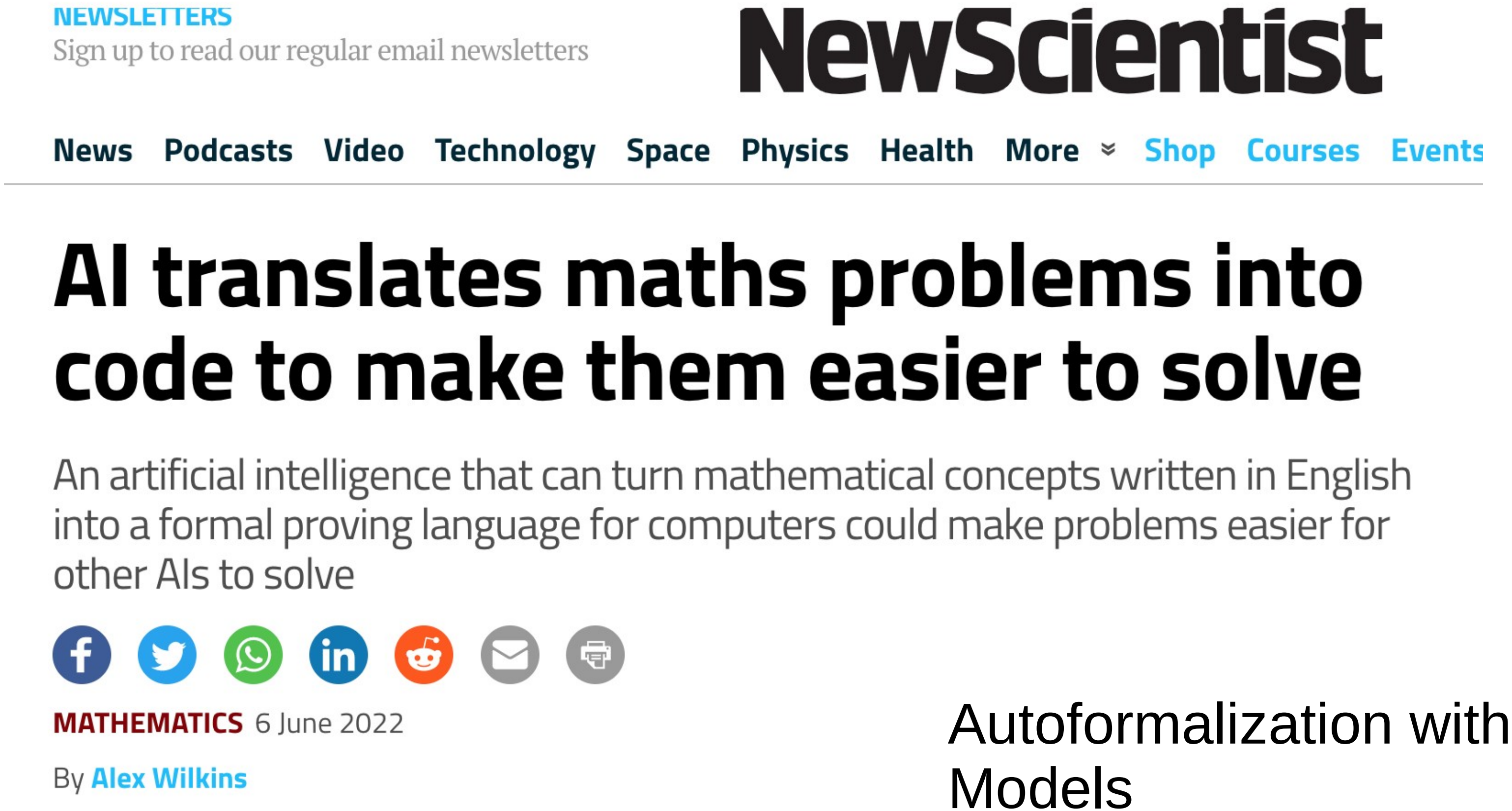
Different but complementary functions:
inducing rationality VS inducing imagination and creativity.

AI/ machine learning and the future of research mathematics

New advances in artificial intelligence and machine learning can promise novel developments in mathematical practice through their applications to automated theorem proving and proof assistants. E.g.: pattern recognition tools from machine learning can find applications in searching the libraries of formal proofs and in recognising proof patterns and providing proof recommendation methods thus enhancing automation.

The communities of machine learning and formal verification have been growing increasingly close during the past few years:

Successful conference series e.g. AITP, CICM, MATH-AI.



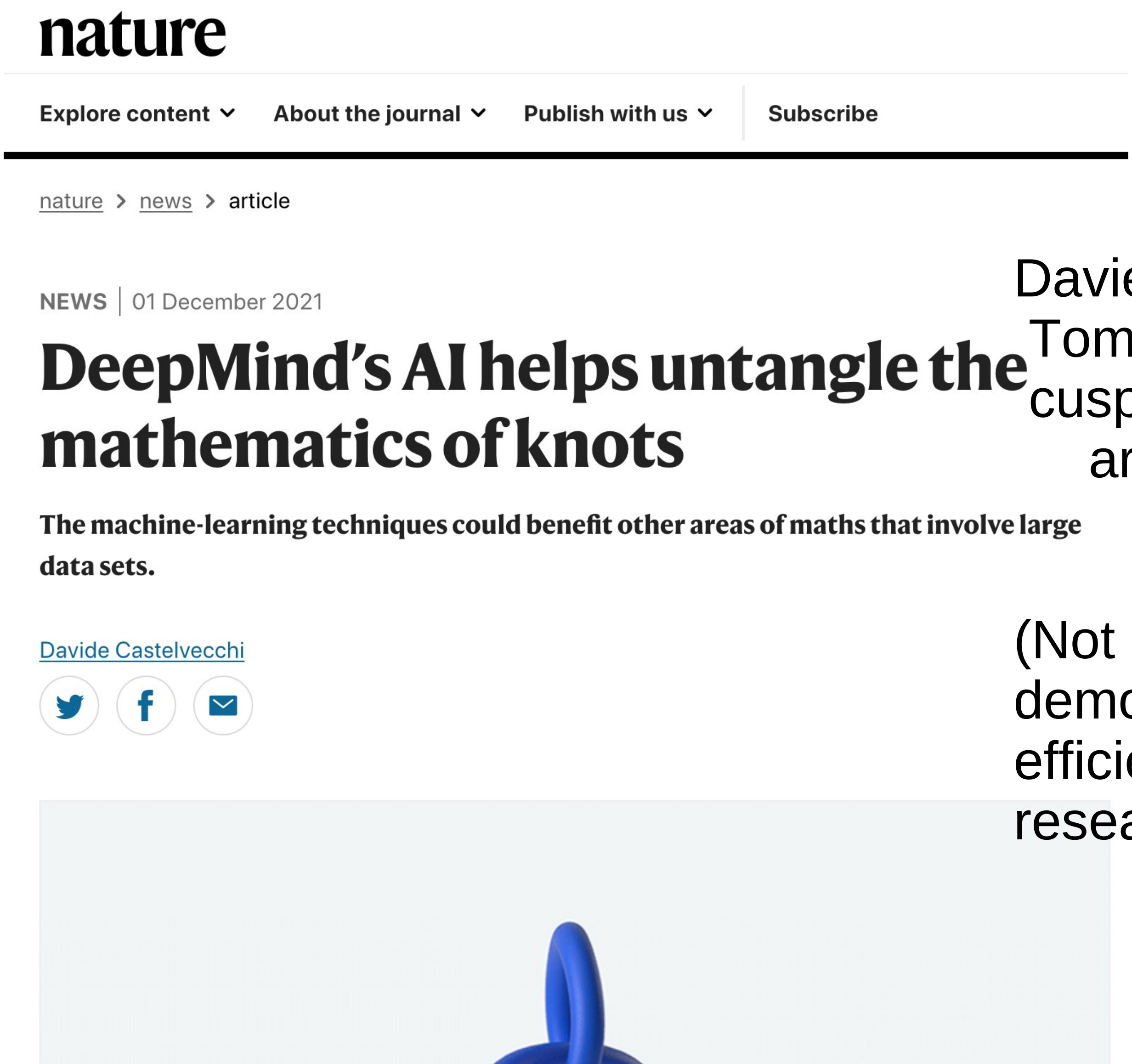
AI translates maths problems into code to make them easier to solve

An artificial intelligence that can turn mathematical concepts written in English into a formal proving language for computers could make problems easier for other AIs to solve

MATHEMATICS 6 June 2022

By Alex Wilkins

Autoformalization with Large Language Models
Wu, Y., Jiang, A. Q., Li, W., Rabe, M. N., Staats, C., Jamnik, M., Szegedy, C.
arXiv:2205.12615v1



NEWS | 01 December 2021

DeepMind's AI helps untangle the mathematics of knots

The machine-learning techniques could benefit other areas of maths that involve large data sets.

Davide Castelvecchi

Davies, A., Juhász, A., Lackenby, M., Tomasev, N., The signature and cusp geometry of hyperbolic knots, arXiv:2111.15323v1

(Not related to proof assistants but demonstrates the pattern-matching efficiency of AI to assist research mathematics.)

Isabelle – A Quick Introduction

Developed by Lawrence C. Paulson (since late 1980's),
Tobias Nipkow, Makarius Wenzel.

Interactive development of verifiable proofs

(Integrates automated reasoning tools in an interactive setting:

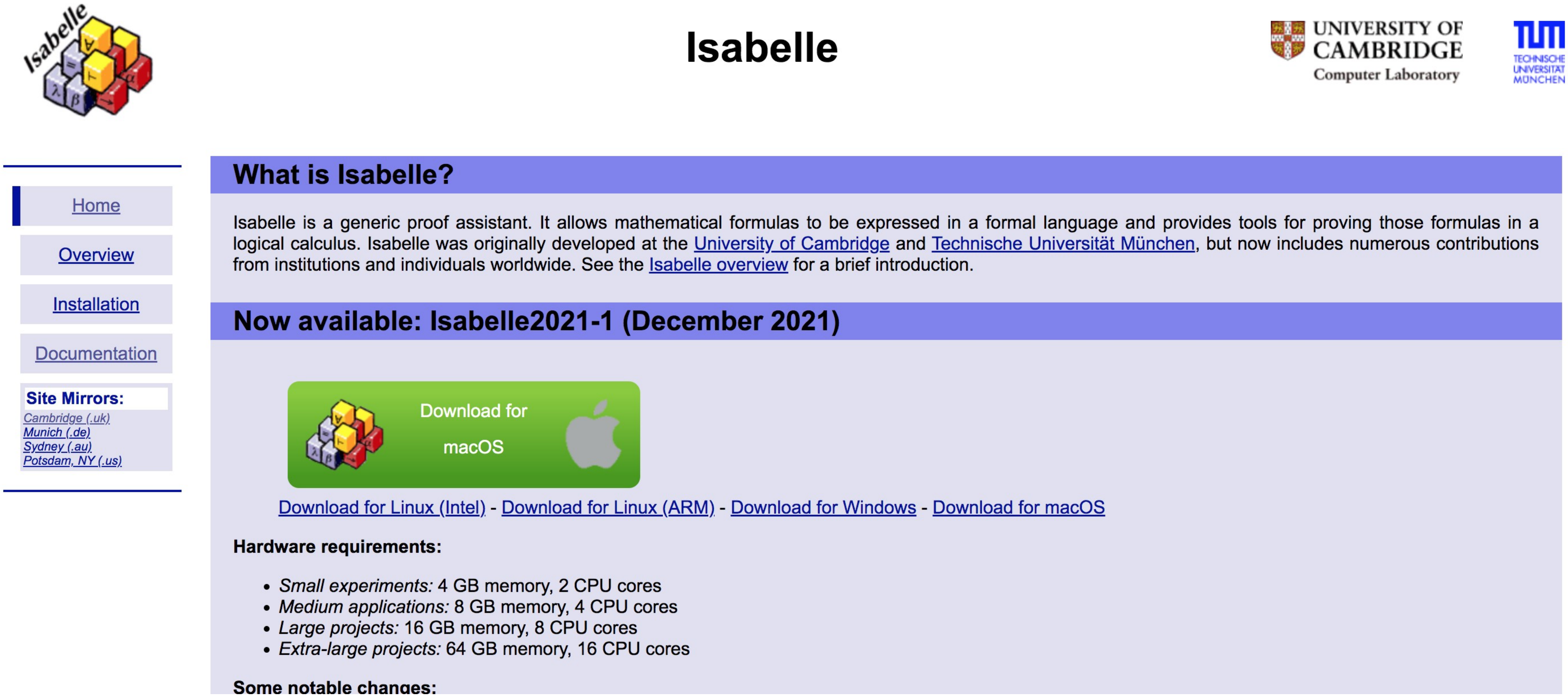
Proof scripts in Isabelle are interactive sessions between user and theorem prover)

- Isabelle/HOL: Higher Order Logic (HOL) (Includes AC; Proofs in classical logic). Simple types.
- Emphasis on producing structured, easy-to-read proofs:

  ISAR (Intelligible Semi-Automated Reasoning) proof language.
  Internal languages: ML and Scala.
- Features efficient automation (Sledgehammer and counterexample-finding tools like nitpick and Quickcheck).

Isabelle – A Quick Introduction

https://www.cl.cam.ac.uk/research/hvg/Isabelle/index.html



Isabelle

What is Isabelle?

Isabelle is a generic proof assistant. It allows mathematical formulas to be expressed in a formal language and provides tools for proving those formulas in a logical calculus. Isabelle was originally developed at the University of Cambridge and Technische Universität München, but now includes numerous contributions from institutions and individuals worldwide. See the Isabelle overview for a brief introduction.

Now available: Isabelle2021-1 (December 2021)

Download for Linux (Intel) - Download for Linux (ARM) - Download for Windows - Download for macOS

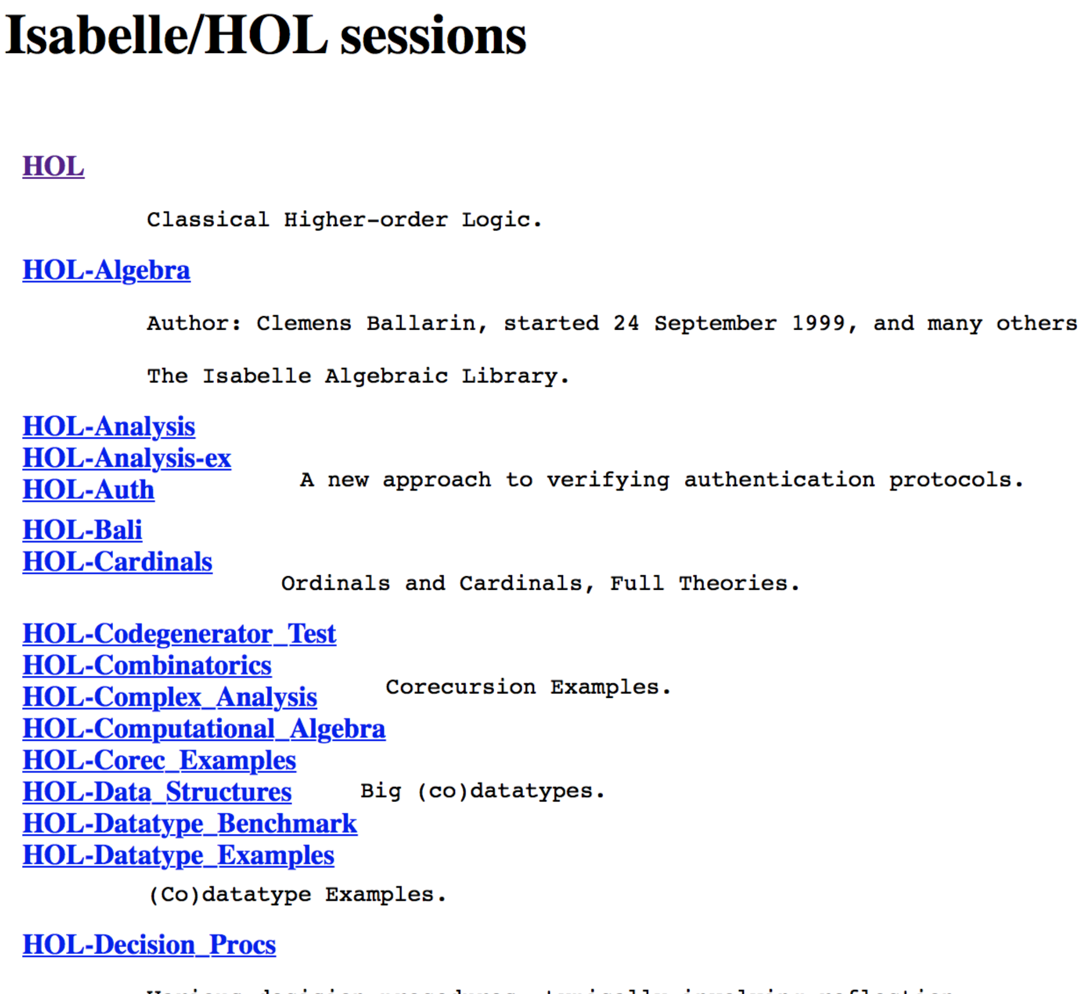
Hardware requirements:

- *Small experiments:* 4 GB memory, 2 CPU cores
- *Medium applications:* 8 GB memory, 4 CPU cores
- *Large projects:* 16 GB memory, 8 CPU cores
- *Extra-large projects:* 64 GB memory, 16 CPU cores

Some notable changes:

Isabelle – A Quick Introduction

https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/library/HOL/index.html



Isabelle/HOL sessions

HOL

Classical Higher-order Logic.

HOL-Algebra

Author: Clemens Ballarin, started 24 September 1999, and many others

The Isabelle Algebraic Library.

HOL-Analysis

HOL-Analysis-ex

HOL-Auth A new approach to verifying authentication protocols.

HOL-Bali

HOL-Cardinals

Ordinals and Cardinals, Full Theories.

HOL-Codegenerator_Test

HOL-Combinatorics

HOL-Complex_Analysis Corecursion Examples.

HOL-Computational_Algebra

HOL-Corec_Examples

HOL-Data_Structures Big (co)datatypes.

HOL-Datatype_Benchmark

HOL-Datatype_Examples

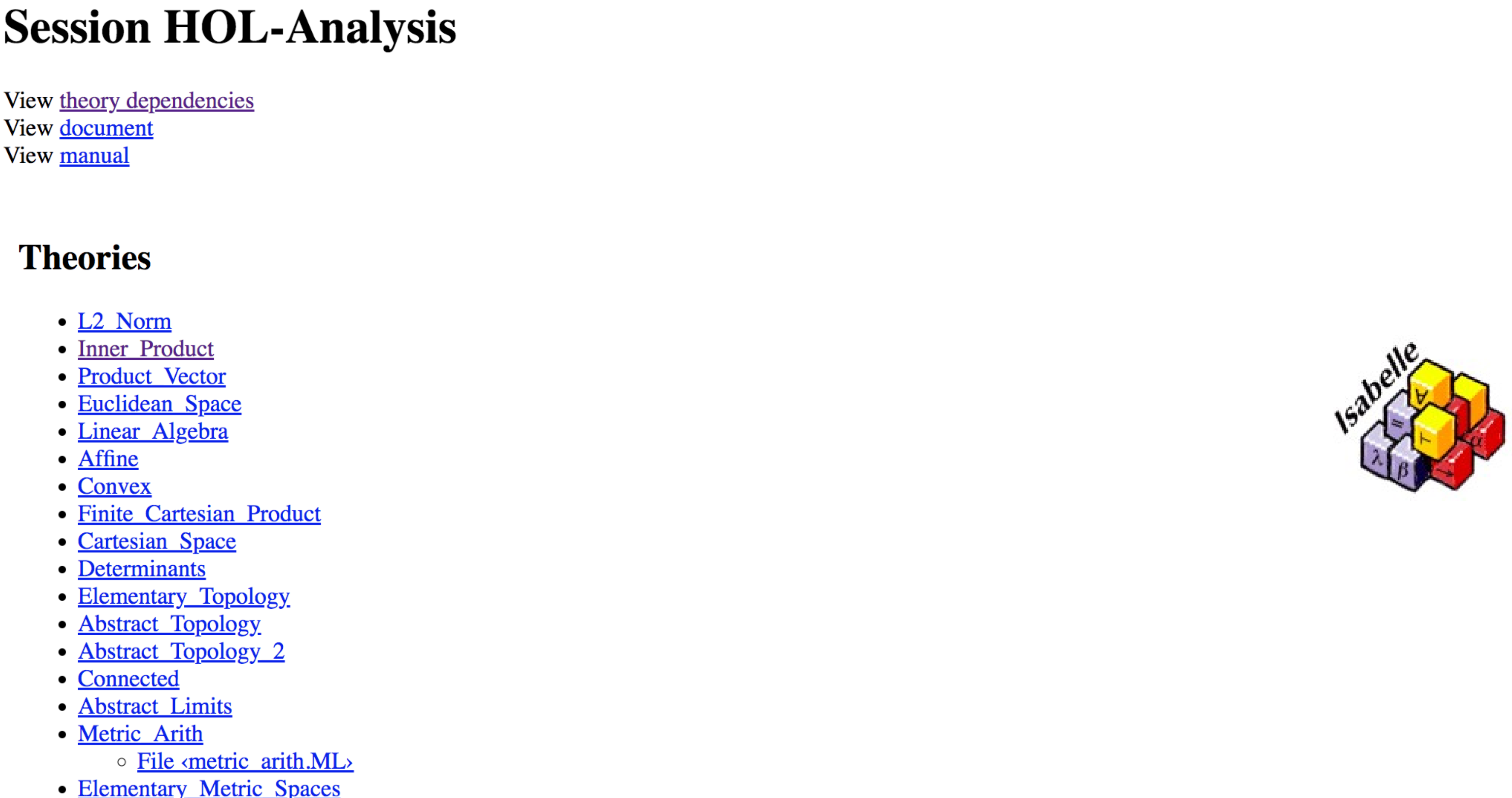
(Co)datatype Examples.

HOL-Decision_Procs

Various decision procedures, typically involving reflection.

Isabelle – A Quick Introduction

https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/library/HOL/HOL-Analysis/index.html



Session HOL-Analysis

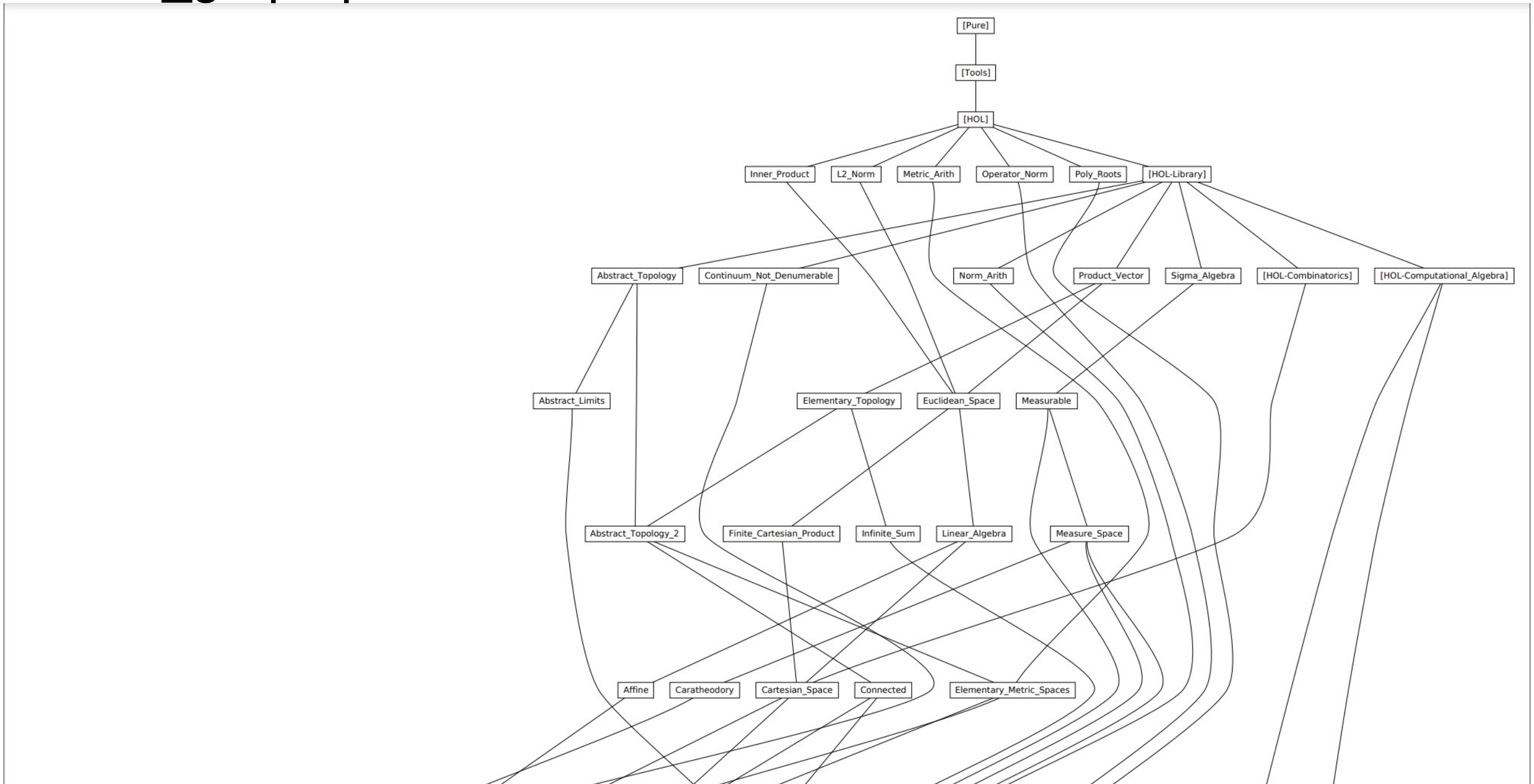
View theory dependencies
View document
View manual

Theories

- L2_Norm
- Inner_Product
- Product_Vector
- Euclidean_Space
- Linear_Algebra
- Affine
- Convex
- Finite_Cartesian_Product
- Cartesian_Space
- Determinants
- Elementary_Topology
- Abstract_Topology
- Abstract_Topology_2
- Connected
- Abstract_Limits
- Metric_Arith
    - File ‹metric_arith.ML›
- Elementary_Metric_Spaces

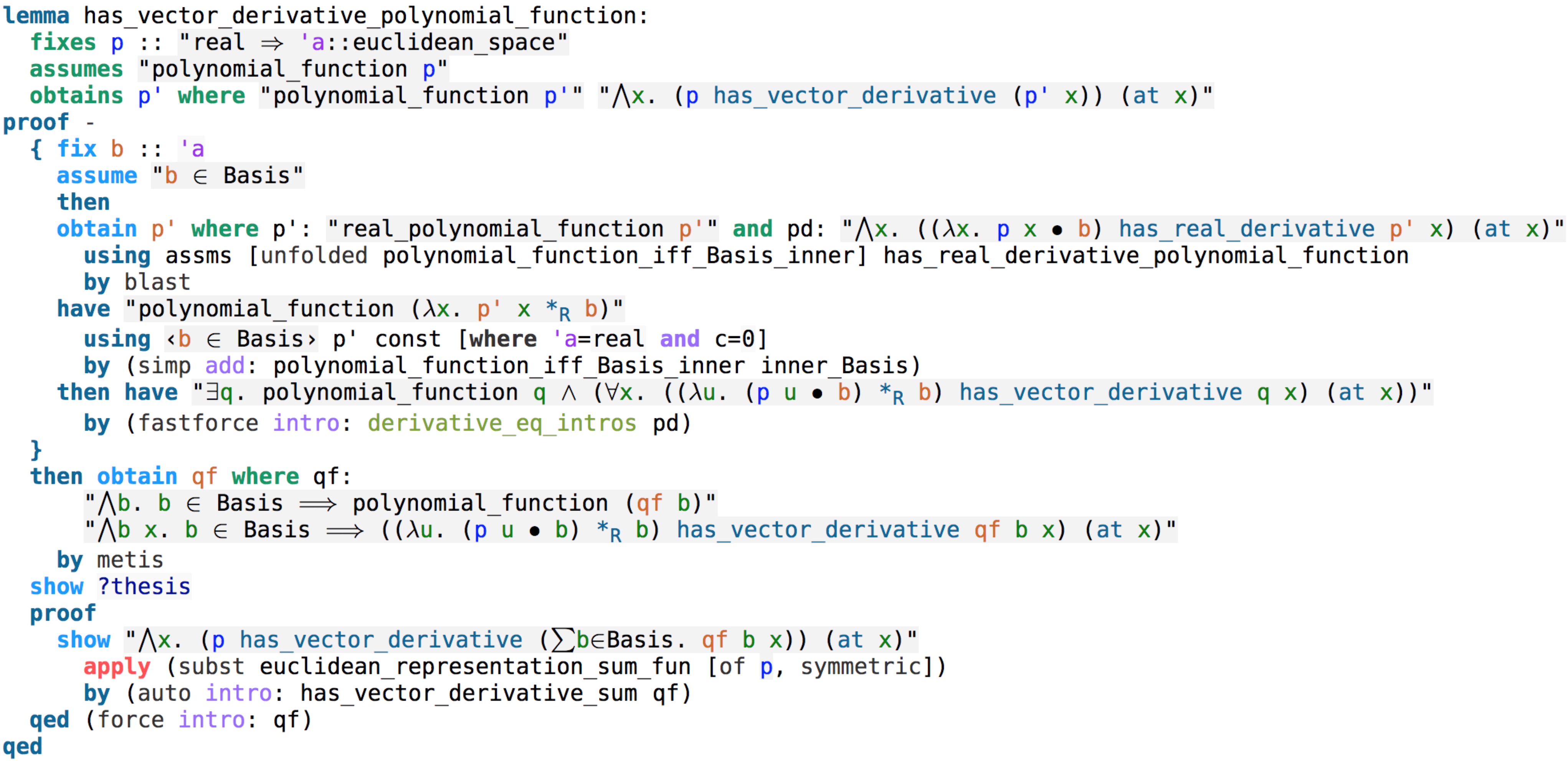
Isabelle – A Quick Introduction

Theory dependencies in the Analysis library
https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/library/HOL/HOL-Analysis/session_graph.pdf



Example of a structured proof in Isabelle/HOL

(from Theory Weierstrass_Theorems in the Isabelle Analysis Library)



```
lemma has_vector_derivative_polynomial_function:
  fixes p :: "real ⇒ 'a::euclidean_space"
  assumes "polynomial_function p"
  obtains p' where "polynomial_function p'" "⋀x. (p has_vector_derivative (p' x)) (at x)"
proof -
  { fix b :: 'a
    assume "b ∈ Basis"
    then
    obtain p' where p': "real_polynomial_function p'" and pd: "⋀x. ((λx. p x ∙ b) has_real_derivative p' x) (at x)"
      using assms [unfolded polynomial_function_iff_Basis_inner] has_real_derivative_polynomial_function
      by blast
    have "polynomial_function (λx. p' x *R b)"
      using ‹b ∈ Basis› p' const [where 'a=real and c=0]
      by (simp add: polynomial_function_iff_Basis_inner inner_Basis)
    then have "∃q. polynomial_function q ∧ (∀x. ((λu. (p u ∙ b) *R b) has_vector_derivative q x) (at x))"
      by (fastforce intro: derivative_eq_intros pd)
  }
  then obtain qf where qf:
      "⋀b. b ∈ Basis ⟹ polynomial_function (qf b)"
      "⋀b x. b ∈ Basis ⟹ ((λu. (p u ∙ b) *R b) has_vector_derivative qf b x) (at x)"
    by metis
  show ?thesis
  proof
    show "⋀x. (p has_vector_derivative (∑b∈Basis. qf b x)) (at x)"
      apply (subst euclidean_representation_sum_fun [of p, symmetric])
      by (auto intro: has_vector_derivative_sum qf)
  qed (force intro: qf)
qed
```

Isabelle – A Quick Introduction

The Archive of Formal Proofs

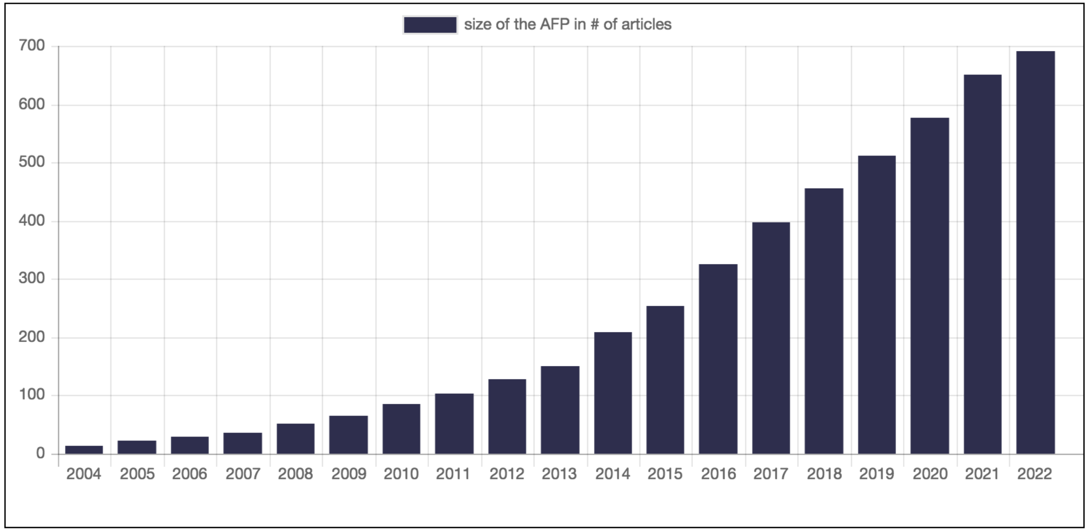
https://www.isa-afp.org/index.html

A vast collection of formalised material in Mathematics, Computer Science and Logic.

Currently:

Number of Entries: 690
Number of Authors: 424
Number of Lemmas: ~212,
Lines of Code: ~3,436,100

Growth in number of articles:



The ALEXANDRIA Project at Cambridge

Large Scale Formal Proof for the Working Mathematician

https://www.cl.cam.ac.uk/~lp15/Grants/Alexandria/
(since Autumn 2017)

UNIVERSITY OF CAMBRIDGE

- Expanding the body of formalised material on the Archive of Formal Proofs and the Isabelle Libraries.
- Case studies to explore the limits of formalisation
- Tools for managing large bodies of formal Mathematical Knowledge (Intelligent Search/ Computer-aided Knowledge Discovery).
- Automated and semi-automated environments and tools to aid *working mathematicians.*

erc
European Research Council
Established by the European Commission

PI: Lawrence C. Paulson FRS
Postdocs: Wenda Li, Anthony Bordg, Yiannos Stathopoulos, Angeliki Koutsoukou-Argyraki. PhD Student: Chelsea Edmonds.
Many external collaborators and interns.

Aristotle's Assertoric Syllogistic

* Source: Robin Smith; *Aristotle's Logic*, Stanford Encyclopedia of Philosophy, first published 18/3/2000, substantive revision 17/2/2017, available on:

https://plato.stanford.edu/entries/aristotle-logic/

* Formal Proof Development: Angeliki Koutsoukou-Argyraki; *Aristotle's Assertoric Syllogistic*, Archive of Formal Proofs, first published 08/10/2019, available on: https://www.isa-afp.org/entries/Aristotles_Assertoric_Syllogistic.html

(Only ~200 lines of Isar code!)

Back to the origins :-)

Aristotle’s Assertoric Syllogistic

Syllogisms are structures of sentences each of which can meaningfully be called true or false (assertions “apophanseis”).

A deduction is speech (logos) in which, certain things having been supposed, something different from those supposed results of necessity because of their being so. (Prior Analytics I.2, 24b18–20).

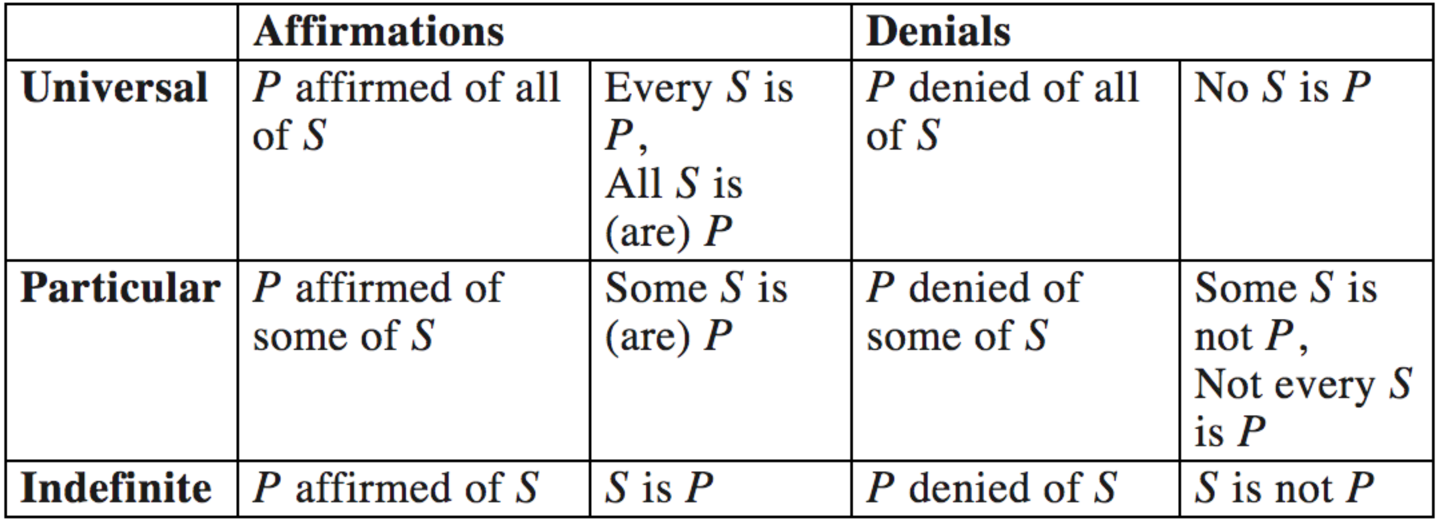
Aristotle’s Assertoric Syllogistic

Assertions (apophanseis): every such sentence must have the same structure: Subject (individual/universal) ; predicate (only universal); must either affirm or deny the predicate of the subject.

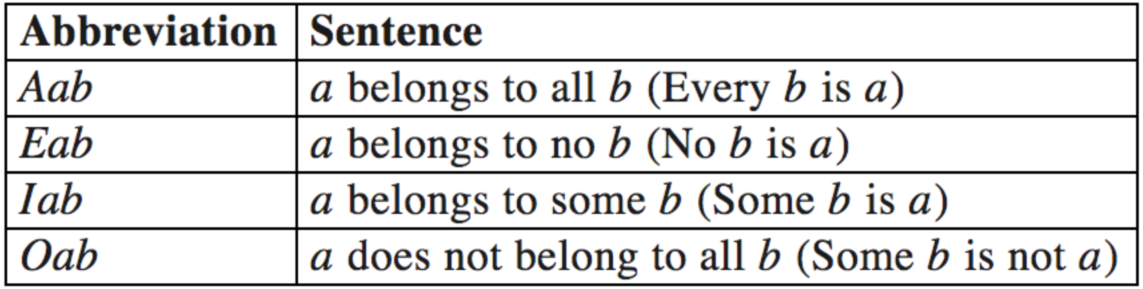
Aristotle treats individual predications and general predications as similar in logical form (“Socrates is an animal”, “Humans are animals”).
When the subject is a universal, predication can be either universal or particular.

Aristotle’s Assertoric Syllogistic

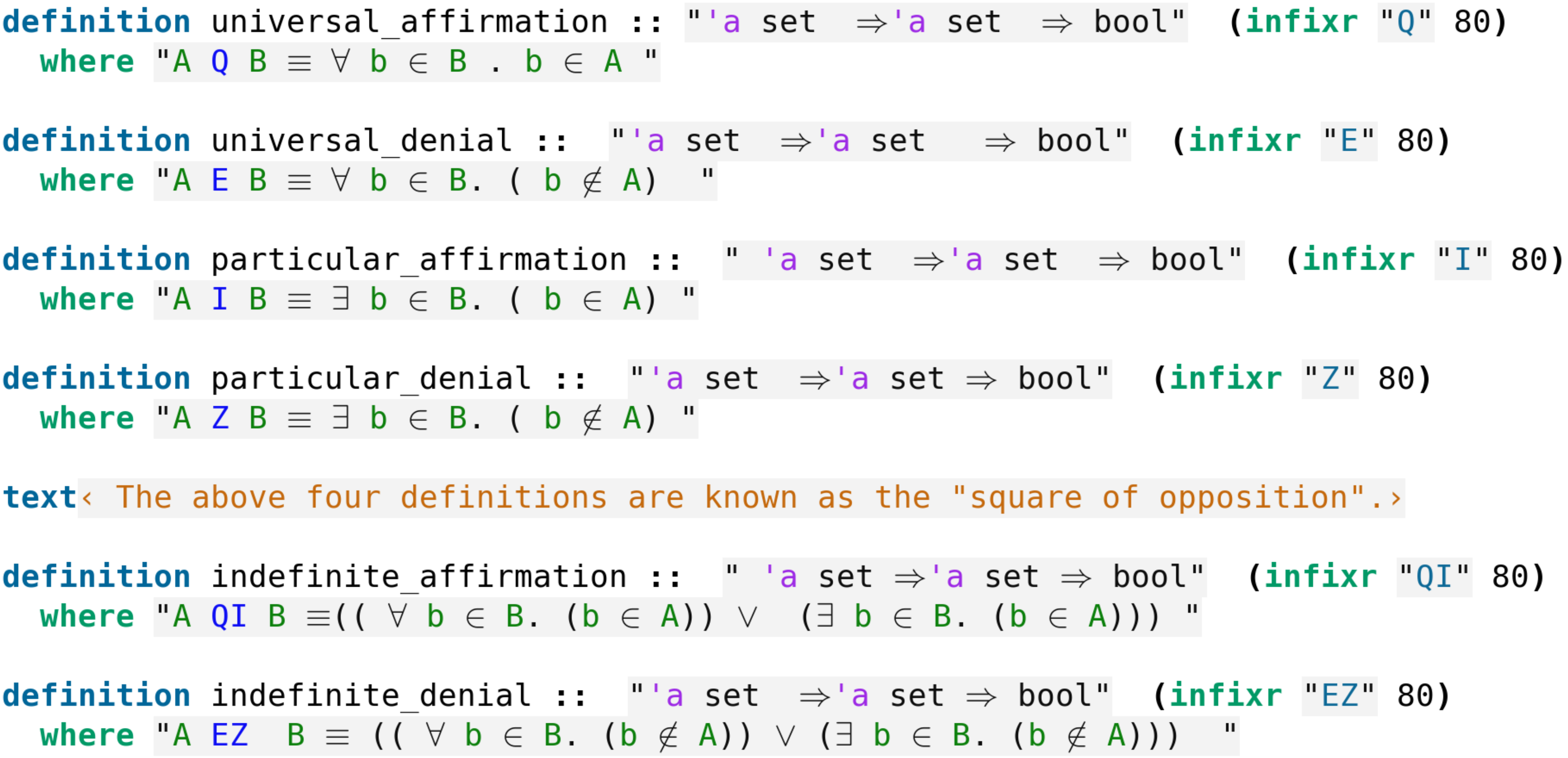
* Source: Robin Smith; *Aristotle’s Logic*, Stanford Encyclopedia of Philosophy, first published 18/3/2000, substantive revision 17/2/2017, available on: https://plato.stanford.edu/entries/aristotle-logic/



	Affirmations		**Denials**	
Universal	*P* affirmed of all of *S*	Every *S* is *P*, All *S* is (are) *P*	*P* denied of all of *S*	No *S* is *P*
Particular	*P* affirmed of some of *S*	Some *S* is (are) *P*	*P* denied of some of *S*	Some *S* is not *P*, Not every *S* is *P*
Indefinite	*P* affirmed of *S*	*S* is *P*	*P* denied of *S*	*S* is not *P*



Abbreviation	**Sentence**
Aab	*a* belongs to all *b* (Every *b* is *a*)
Eab	*a* belongs to no *b* (No *b* is *a*)
Iab	*a* belongs to some *b* (Some *b* is *a*)
Oab	*a* does not belong to all *b* (Some *b* is not *a*)



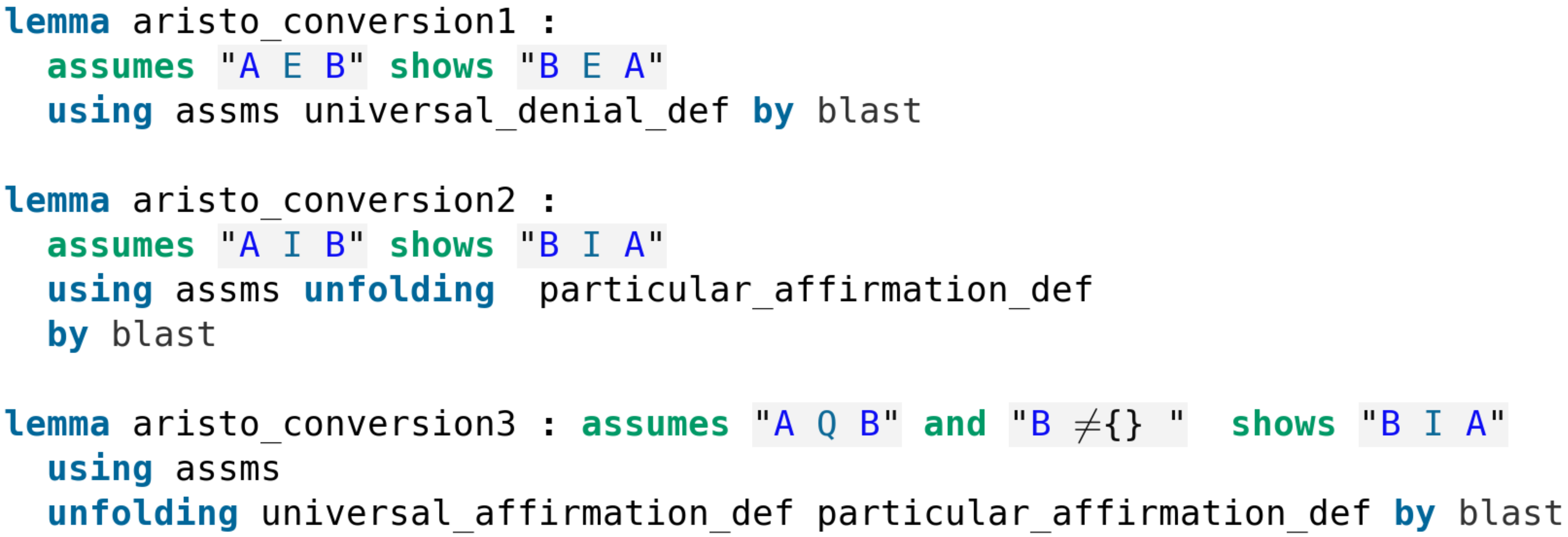
```
definition universal_affirmation :: "'a set  ⇒'a set  ⇒ bool"  (infixr "Q" 80)
  where "A Q B ≡ ∀ b ∈ B . b ∈ A "

definition universal_denial ::  "'a set  ⇒'a set   ⇒ bool"  (infixr "E" 80)
  where "A E B ≡ ∀ b ∈ B. ( b ∉ A)   "

definition particular_affirmation ::  " 'a set  ⇒'a set  ⇒ bool"  (infixr "I" 80)
  where "A I B ≡ ∃ b ∈ B. ( b ∈ A) "

definition particular_denial ::  "'a set  ⇒'a set ⇒ bool"  (infixr "Z" 80)
  where "A Z B ≡ ∃ b ∈ B. ( b ∉ A) "

text‹ The above four definitions are known as the "square of opposition".›

definition indefinite_affirmation ::  " 'a set ⇒'a set ⇒ bool"  (infixr "QI" 80)
  where "A QI B ≡(( ∀ b ∈ B. (b ∈ A)) ∨  (∃ b ∈ B. (b ∈ A))) "

definition indefinite_denial ::  "'a set  ⇒'a set ⇒ bool"  (infixr "EZ" 80)
  where "A EZ  B ≡ (( ∀ b ∈ B. (b ∉ A)) ∨ (∃ b ∈ B. (b ∉ A)))  "
```

(Note: Aristotle would never consider A to be an 1-element set)

$$Eab \rightarrow Eba$$

$$Iab \rightarrow Iba$$

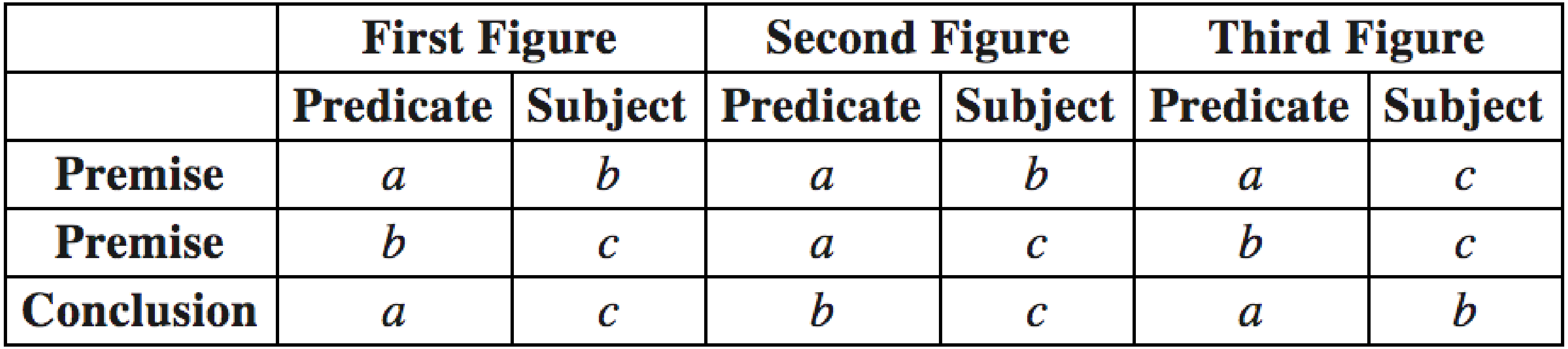
$$Aab \rightarrow Iba$$



```
lemma aristo_conversion1 :
  assumes "A E B" shows "B E A"
  using assms universal_denial_def by blast

lemma aristo_conversion2 :
  assumes "A I B" shows "B I A"
  using assms unfolding  particular_affirmation_def
  by blast

lemma aristo_conversion3 : assumes "A Q B" and "B ≠{} "  shows "B I A"
  using assms
  unfolding universal_affirmation_def particular_affirmation_def by blast
```

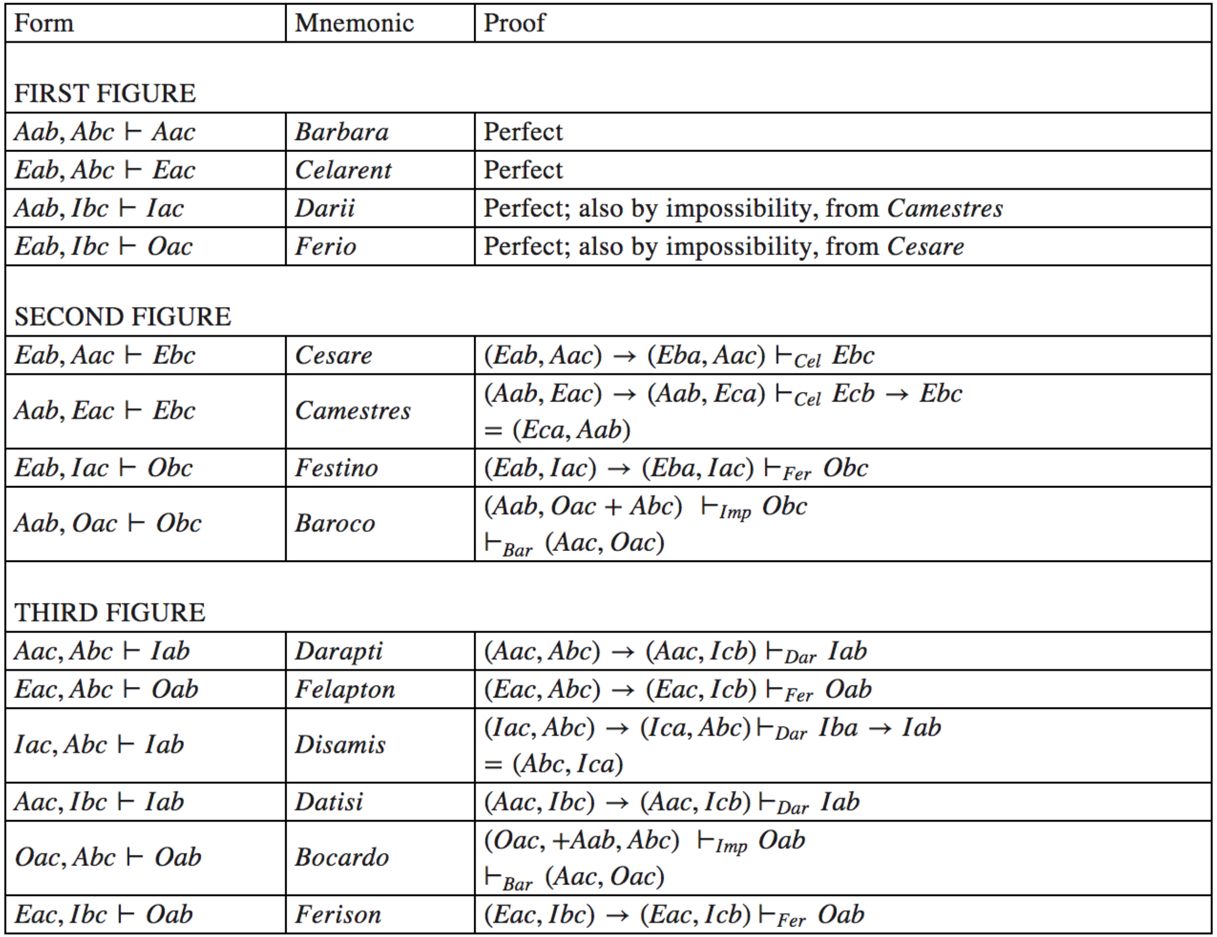
Aristotle’s Assertoric Syllogistic: the Deductions in the Figures (Moods)

* Source: Robin Smith; *Aristotle’s Logic*, Stanford Encyclopedia of Philosophy, first published 18/3/2000, substantive revision 17/2/2017, available on: https://plato.stanford.edu/entries/aristotle-logic/



	First Figure		**Second Figure**		**Third Figure**	
	Predicate	**Subject**	**Predicate**	**Subject**	**Predicate**	**Subject**
Premise	*a*	*b*	*a*	*b*	*a*	*c*
Premise	*b*	*c*	*a*	*c*	*b*	*c*
Conclusion	*a*	*c*	*b*	*c*	*a*	*b*

Aristotle’s Assertoric Syllogistic: the Deductions in the Figures (Moods)



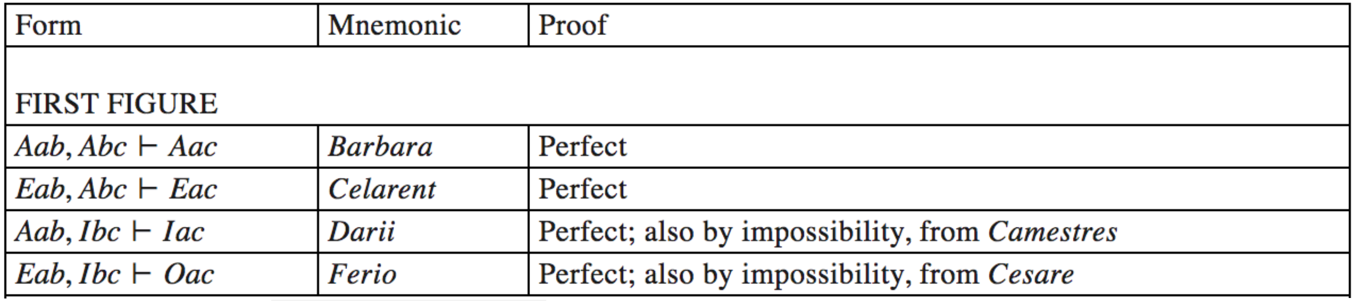
Form	Mnemonic	Proof
FIRST FIGURE		
$Aab, Abc \vdash Aac$	*Barbara*	Perfect
$Eab, Abc \vdash Eac$	*Celarent*	Perfect
$Aab, Ibc \vdash Iac$	*Darii*	Perfect; also by impossibility, from *Camestres*
$Eab, Ibc \vdash Oac$	*Ferio*	Perfect; also by impossibility, from *Cesare*
SECOND FIGURE		
$Eab, Aac \vdash Ebc$	*Cesare*	$(Eab, Aac) \to (Eba, Aac) \vdash_{Cel} Ebc$
$Aab, Eac \vdash Ebc$	*Camestres*	$(Aab, Eac) \to (Aab, Eca) \vdash_{Cel} Ecb \to Ebc$ $= (Eca, Aab)$
$Eab, Iac \vdash Obc$	*Festino*	$(Eab, Iac) \to (Eba, Iac) \vdash_{Fer} Obc$
$Aab, Oac \vdash Obc$	*Baroco*	$(Aab, Oac + Abc) \ \vdash_{Imp} Obc$ $\vdash_{Bar} (Aac, Oac)$
THIRD FIGURE		
$Aac, Abc \vdash Iab$	*Darapti*	$(Aac, Abc) \to (Aac, Icb) \vdash_{Dar} Iab$
$Eac, Abc \vdash Oab$	*Felapton*	$(Eac, Abc) \to (Eac, Icb) \vdash_{Fer} Oab$
$Iac, Abc \vdash Iab$	*Disamis*	$(Iac, Abc) \to (Ica, Abc) \vdash_{Dar} Iba \to Iab$ $= (Abc, Ica)$
$Aac, Ibc \vdash Iab$	*Datisi*	$(Aac, Ibc) \to (Aac, Icb) \vdash_{Dar} Iab$
$Oac, Abc \vdash Oab$	*Bocardo*	$(Oac, +Aab, Abc) \ \vdash_{Imp} Oab$ $\vdash_{Bar} (Aac, Oac)$
$Eac, Ibc \vdash Oab$	*Ferison*	$(Eac, Ibc) \to (Eac, Icb) \vdash_{Fer} Oab$

Table of the Deductions in the Figures

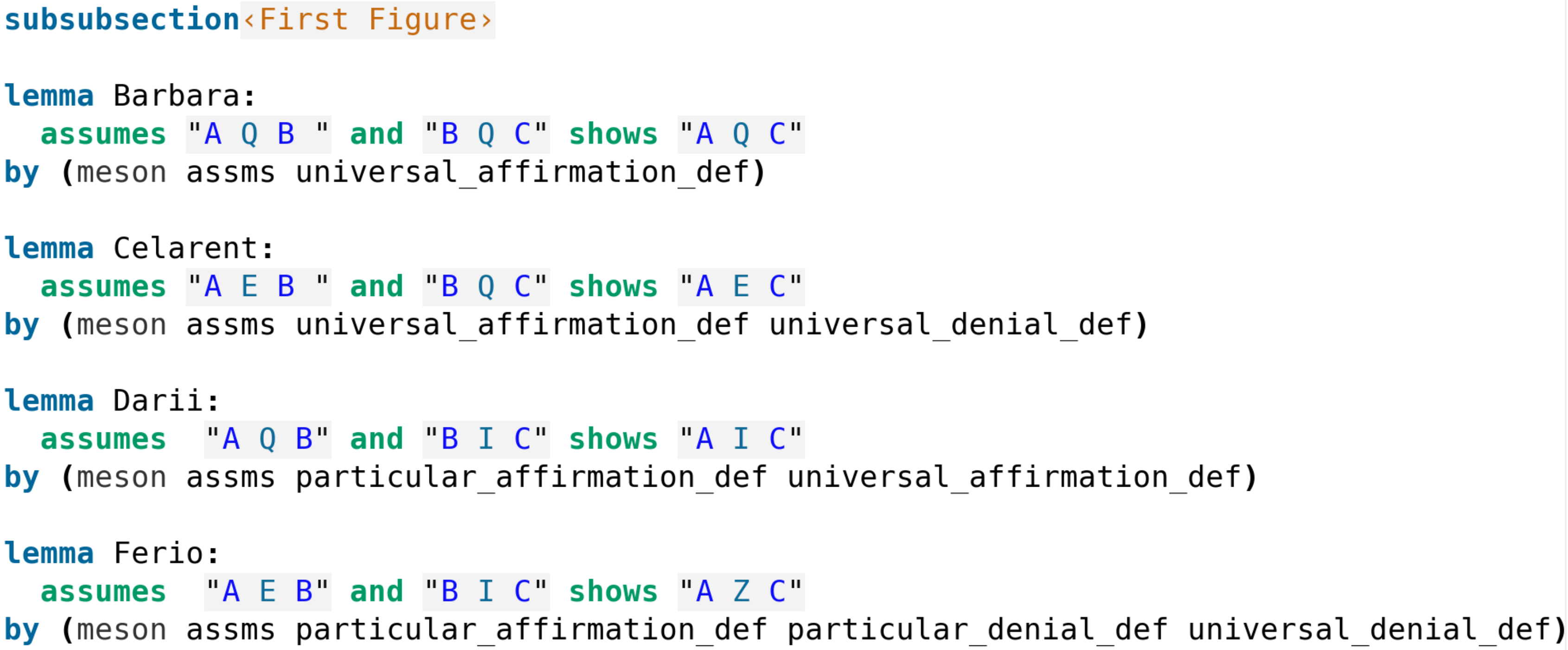
* Source: Robin Smith; *Aristotle’s Logic*, Stanford Encyclopedia of Philosophy, first published 18/3/2000, substantive revision 17/2/2017, available on: https://plato.stanford.edu/entries/aristotle-logic/

Aristotle’s Assertoric Syllogistic: the Deductions in the Figures (“Moods”)

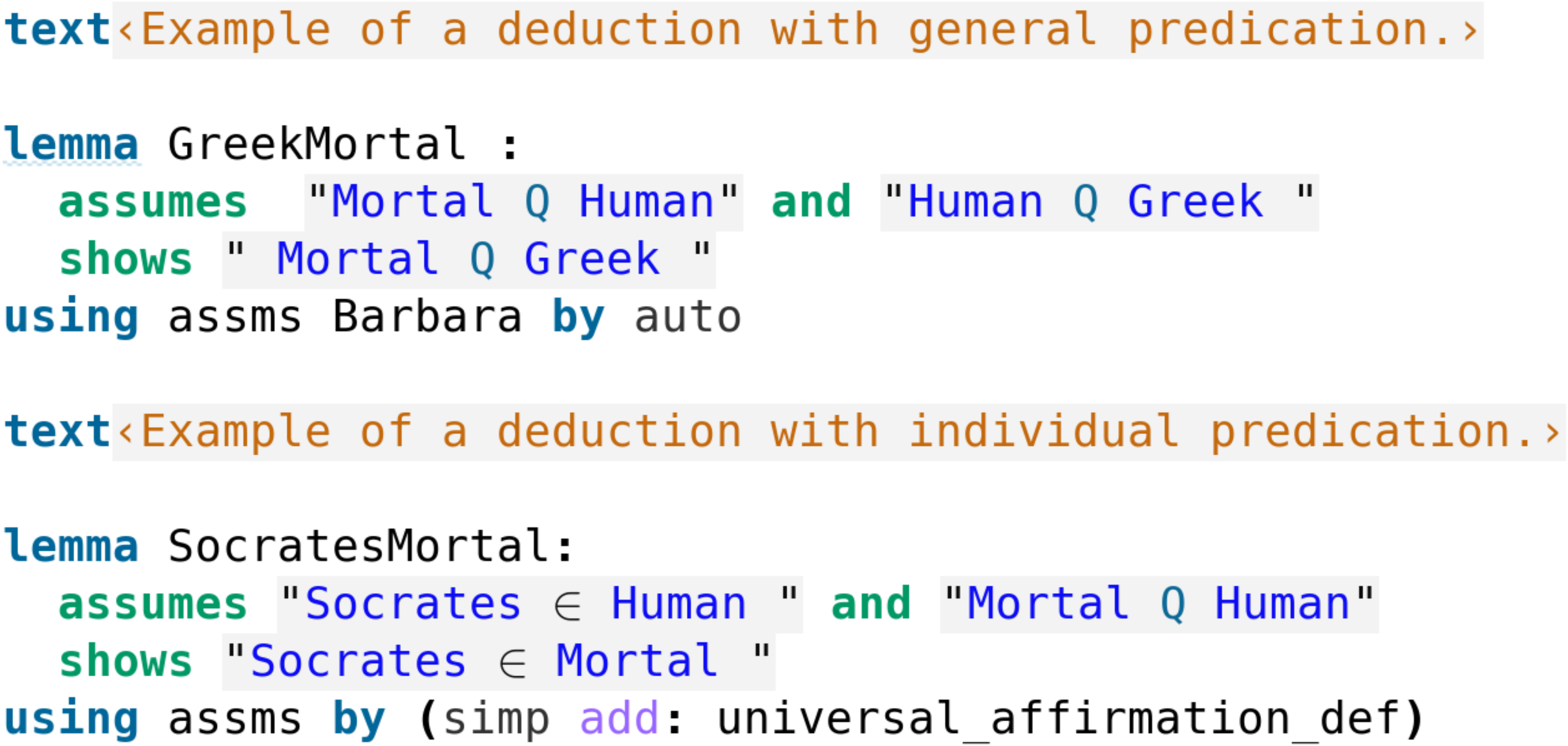
* Source: Robin Smith; *Aristotle’s Logic*, Stanford Encyclopedia of Philosophy, https://plato.stanford.edu/entries/aristotle-logic/



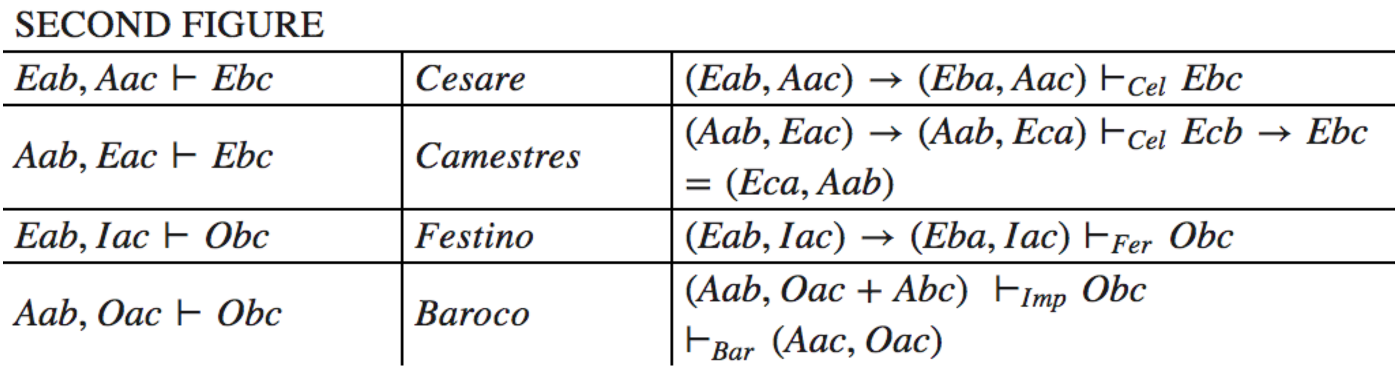
Form	Mnemonic	Proof
FIRST FIGURE		
$Aab, Abc \vdash Aac$	*Barbara*	Perfect
$Eab, Abc \vdash Eac$	*Celarent*	Perfect
$Aab, Ibc \vdash Iac$	*Darii*	Perfect; also by impossibility, from *Camestres*
$Eab, Ibc \vdash Oac$	*Ferio*	Perfect; also by impossibility, from *Cesare*



```
subsubsection‹First Figure›

lemma Barbara:
  assumes "A Q B " and "B Q C" shows "A Q C"
by (meson assms universal_affirmation_def)

lemma Celarent:
  assumes "A E B " and "B Q C" shows "A E C"
by (meson assms universal_affirmation_def universal_denial_def)

lemma Darii:
  assumes  "A Q B" and "B I C" shows "A I C"
by (meson assms particular_affirmation_def universal_affirmation_def)

lemma Ferio:
  assumes  "A E B" and "B I C" shows "A Z C"
by (meson assms particular_affirmation_def particular_denial_def universal_denial_def)
```



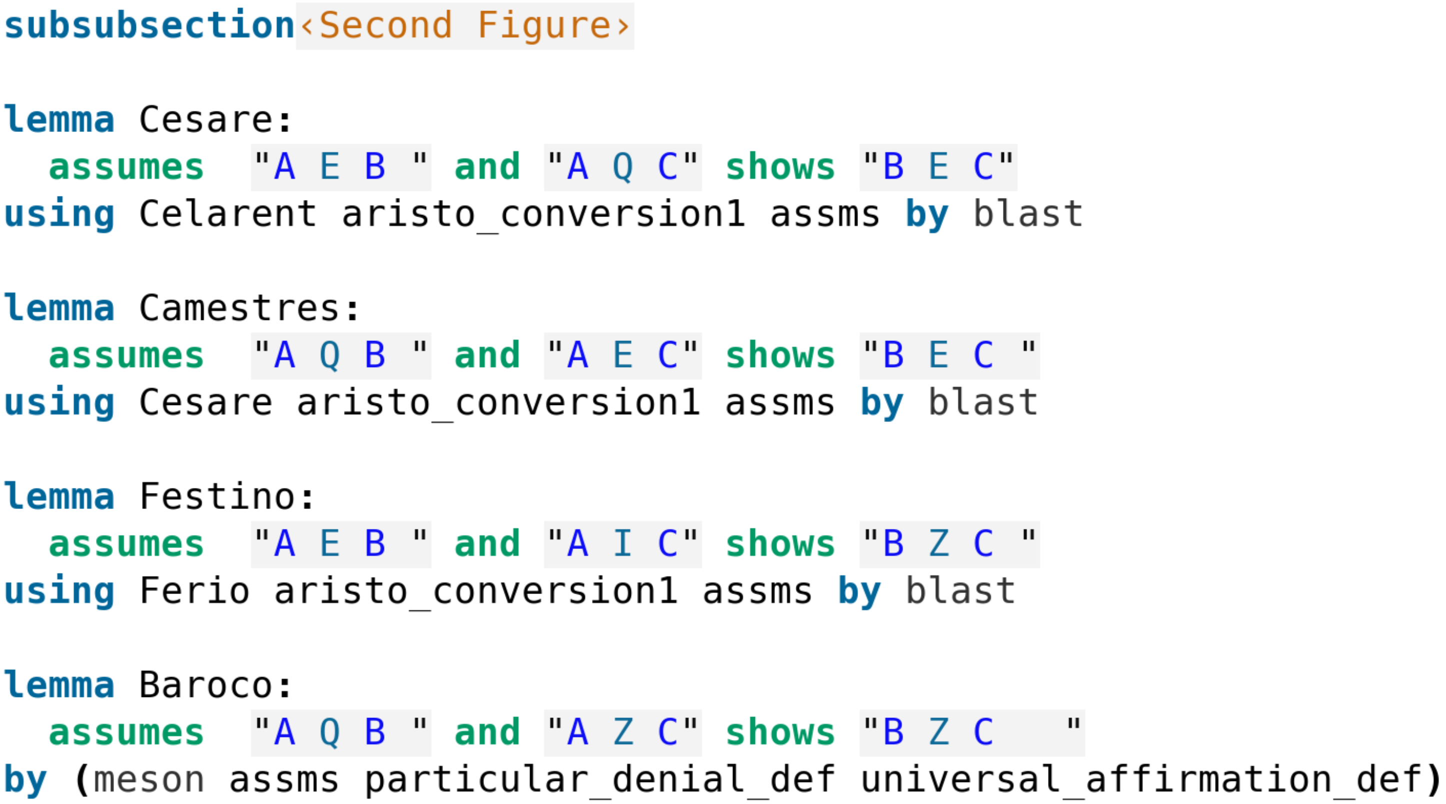
```
text‹Example of a deduction with general predication.›

lemma GreekMortal :
  assumes  "Mortal Q Human" and "Human Q Greek "
  shows " Mortal Q Greek "
using assms Barbara by auto

text‹Example of a deduction with individual predication.›

lemma SocratesMortal:
  assumes "Socrates ∈ Human " and "Mortal Q Human"
  shows "Socrates ∈ Mortal "
using assms by (simp add: universal_affirmation_def)
```



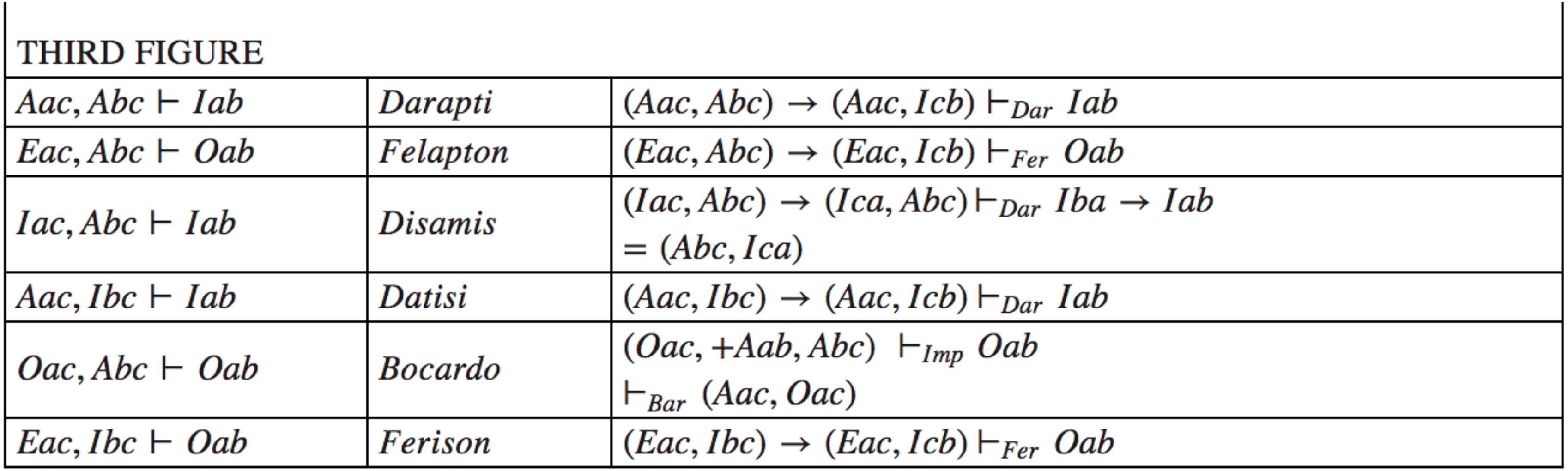
SECOND FIGURE

$Eab, Aac \vdash Ebc$	*Cesare*	$(Eab, Aac) \rightarrow (Eba, Aac) \vdash_{Cel} Ebc$
$Aab, Eac \vdash Ebc$	*Camestres*	$(Aab, Eac) \rightarrow (Aab, Eca) \vdash_{Cel} Ecb \rightarrow Ebc = (Eca, Aab)$
$Eab, Iac \vdash Obc$	*Festino*	$(Eab, Iac) \rightarrow (Eba, Iac) \vdash_{Fer} Obc$
$Aab, Oac \vdash Obc$	*Baroco*	$(Aab, Oac + Abc) \ \vdash_{Imp} Obc \vdash_{Bar} (Aac, Oac)$

* Source: Robin Smith; *Aristotle’s Logic*, Stanford Encyclopedia of Philosophy, https://plato.stanford.edu/entries/aristotle-logic/



```
subsubsection‹Second Figure›

lemma Cesare:
  assumes  "A E B " and "A Q C" shows "B E C"
using Celarent aristo_conversion1 assms by blast

lemma Camestres:
  assumes  "A Q B " and "A E C" shows "B E C "
using Cesare aristo_conversion1 assms by blast

lemma Festino:
  assumes  "A E B " and "A I C" shows "B Z C "
using Ferio aristo_conversion1 assms by blast

lemma Baroco:
  assumes  "A Q B " and "A Z C" shows "B Z C   "
by (meson assms particular_denial_def universal_affirmation_def)
```



THIRD FIGURE

$Aac, Abc \vdash Iab$	*Darapti*	$(Aac, Abc) \rightarrow (Aac, Icb) \vdash_{Dar} Iab$
$Eac, Abc \vdash Oab$	*Felapton*	$(Eac, Abc) \rightarrow (Eac, Icb) \vdash_{Fer} Oab$
$Iac, Abc \vdash Iab$	*Disamis*	$(Iac, Abc) \rightarrow (Ica, Abc) \vdash_{Dar} Iba \rightarrow Iab$ $= (Abc, Ica)$
$Aac, Ibc \vdash Iab$	*Datisi*	$(Aac, Ibc) \rightarrow (Aac, Icb) \vdash_{Dar} Iab$
$Oac, Abc \vdash Oab$	*Bocardo*	$(Oac, +Aab, Abc) \ \vdash_{Imp} Oab$ $\vdash_{Bar} (Aac, Oac)$
$Eac, Ibc \vdash Oab$	*Ferison*	$(Eac, Ibc) \rightarrow (Eac, Icb) \vdash_{Fer} Oab$

* Source: Robin Smith; *Aristotle’s Logic*, Stanford Encyclopedia of Philosophy, https://plato.stanford.edu/entries/aristotle-logic/

```
subsubsection‹Third Figure›

lemma Darapti:
  assumes  "A Q C " and "B Q C" and "C ≠{}"   shows "A I B "
  using Darii assms unfolding  universal_affirmation_def particular_affirmation_def
  by blast

lemma Felapton:
  assumes  "A E C" and "B Q C"  and  "C ≠{}"   shows "A Z B"
 using Festino aristo_conversion1 aristo_conversion3 assms by blast

lemma Disamis:
  assumes  "A I C" and "B Q C" shows "A I B"
  using Darii aristo_conversion2 assms by blast

lemma Datisi:
  assumes  "A Q C" and "B I C" shows "A I B"
  using Disamis aristo_conversion2 assms by blast

lemma Bocardo:
  assumes  "A Z C" and "B Q C" shows "A Z B"
 by (meson assms particular_denial_def universal_affirmation_def)

lemma Ferison:
  assumes  "A E C " and "B I C" shows "A Z B   "
using Ferio aristo_conversion2 assms by blast
```

Aristotle’s Assertoric Syllogistic

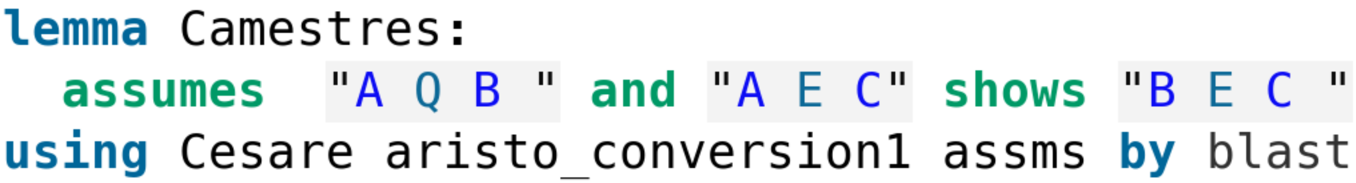
A metatheorem by Aristotle:

All deductions can be reduced to Barbara/ Celarent.

Observations

1) Using Isabelle’s automation (Sledgehammer),
the proofs of the deductions in the Figures are straightforward (one-line)
The de Bruijn factor would be < 1 !

Example: Compare

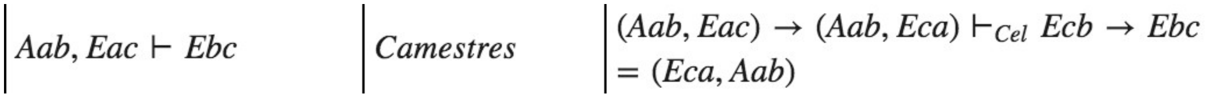


```
lemma Camestres:
  assumes  "A Q B " and "A E C" shows "B E C "
using Cesare aristo_conversion1 assms by blast
```

(note: Cesare
reduces to Celarent)

with the original proof:

Aristotle’s proof of Camestres



$Aab, Eac \vdash Ebc$	*Camestres*	$(Aab, Eac) \rightarrow (Aab, Eca) \vdash_{Cel} Ecb \rightarrow Ebc$ $= (Eca, Aab)$

“If a belongs to every b (:= every b is a) but to no c (:=no c is a), then neither will b belong to any c (:=no c is b). For if a belongs to no c (:= no c is a) , then neither does c belong to any a (:= no a is c); but a belonged to every b (:=every b is a); therefore, c will belong to no b (:= no b is c) (for the first figure has come about). And since the privative converts, neither will b belong to any c (:=no c is b).”

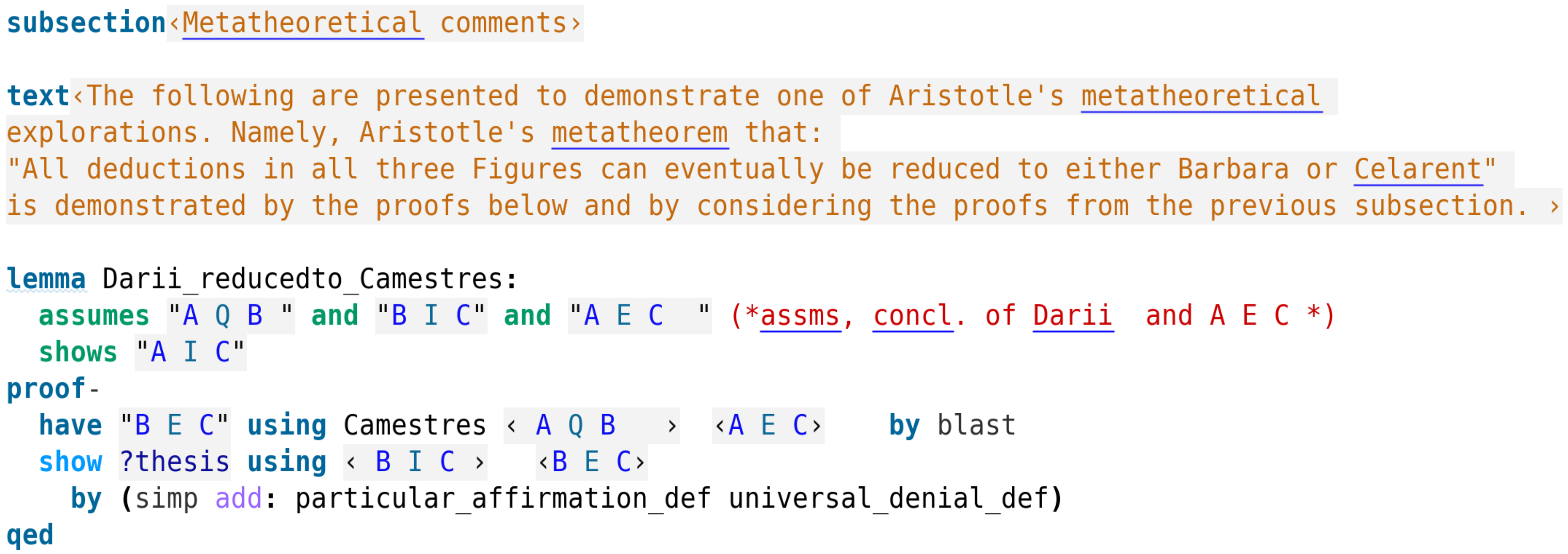
Written as:

(1) Aab, (2) Eac, To prove: Ebc.

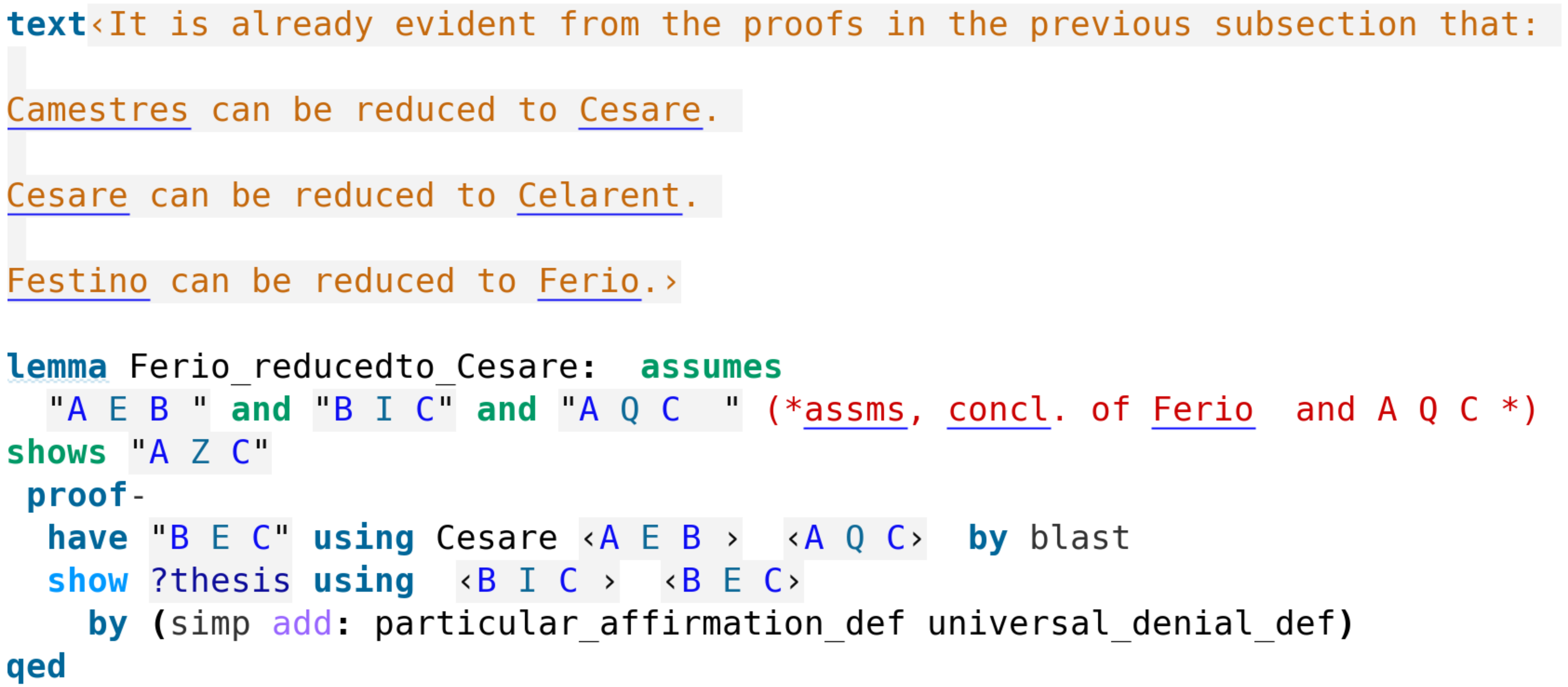
(3) Eac (from (2))
(4) Eca (from (3) and conversion)
(5) Aab (from (1))
(6) Ecb (from (4), (5) and Celarent)
(7) Ebc (from (6) and conversion)

Observations

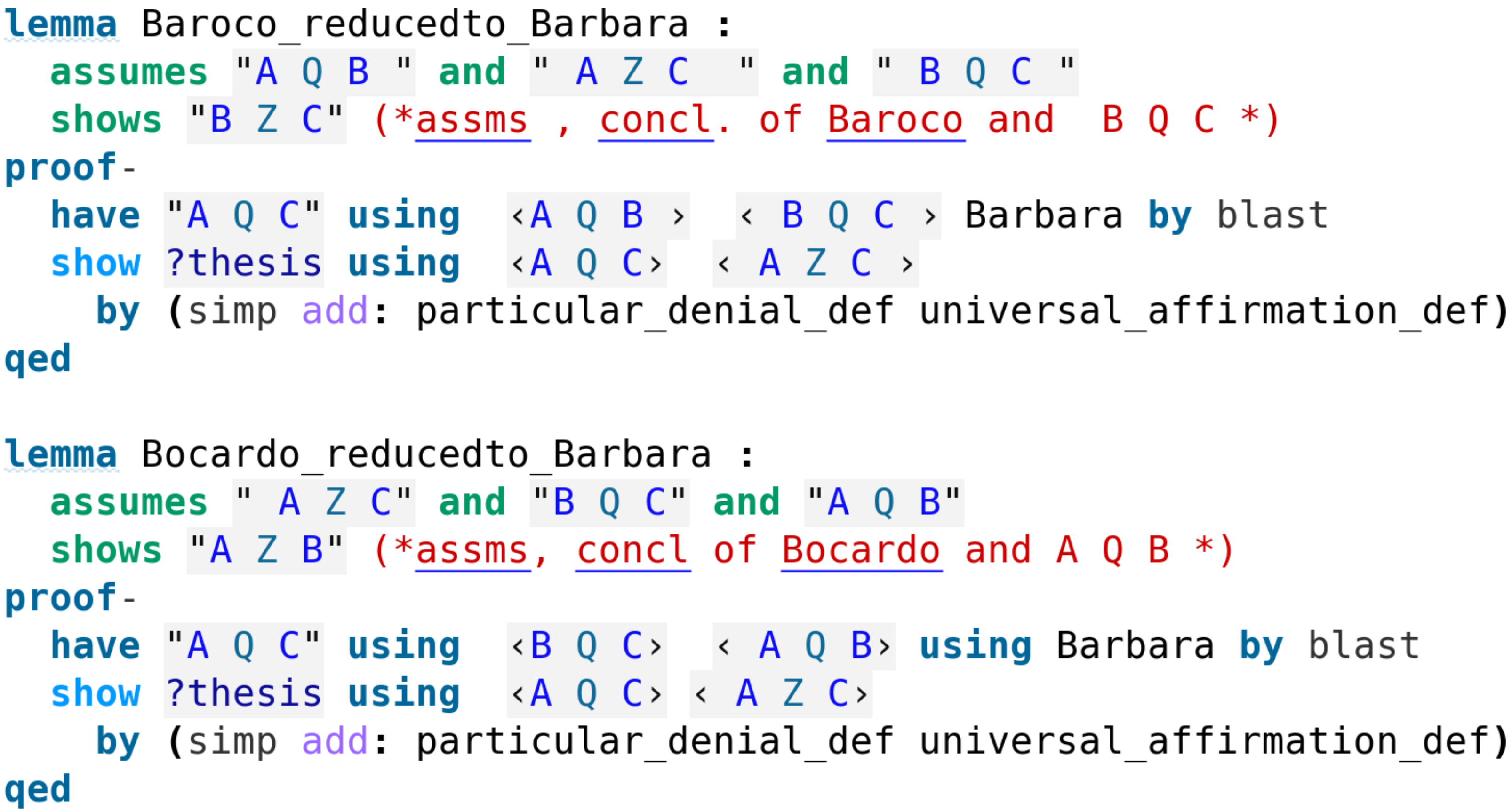
2) The metatheorem that all deductions can be reduced to Barbara/ Celarent can be seen easily from the formal proofs:



```
subsection‹Metatheoretical comments›

text‹The following are presented to demonstrate one of Aristotle's metatheoretical
explorations. Namely, Aristotle's metatheorem that:
"All deductions in all three Figures can eventually be reduced to either Barbara or Celarent"
is demonstrated by the proofs below and by considering the proofs from the previous subsection. ›

lemma Darii_reducedto_Camestres:
  assumes "A Q B " and "B I C" and "A E C  " (*assms, concl. of Darii  and A E C *)
  shows "A I C"
proof-
  have "B E C" using Camestres ‹ A Q B   ›  ‹A E C›    by blast
  show ?thesis using ‹ B I C ›   ‹B E C›
    by (simp add: particular_affirmation_def universal_denial_def)
qed
```



```
text‹It is already evident from the proofs in the previous subsection that:

Camestres can be reduced to Cesare.

Cesare can be reduced to Celarent.

Festino can be reduced to Ferio.›

lemma Ferio_reducedto_Cesare:  assumes
  "A E B " and "B I C" and "A Q C   " (*assms, concl. of Ferio  and A Q C *)
shows "A Z C"
 proof-
  have "B E C" using Cesare ‹A E B ›  ‹A Q C›  by blast
  show ?thesis using  ‹B I C ›  ‹B E C›
    by (simp add: particular_affirmation_def universal_denial_def)
qed
```

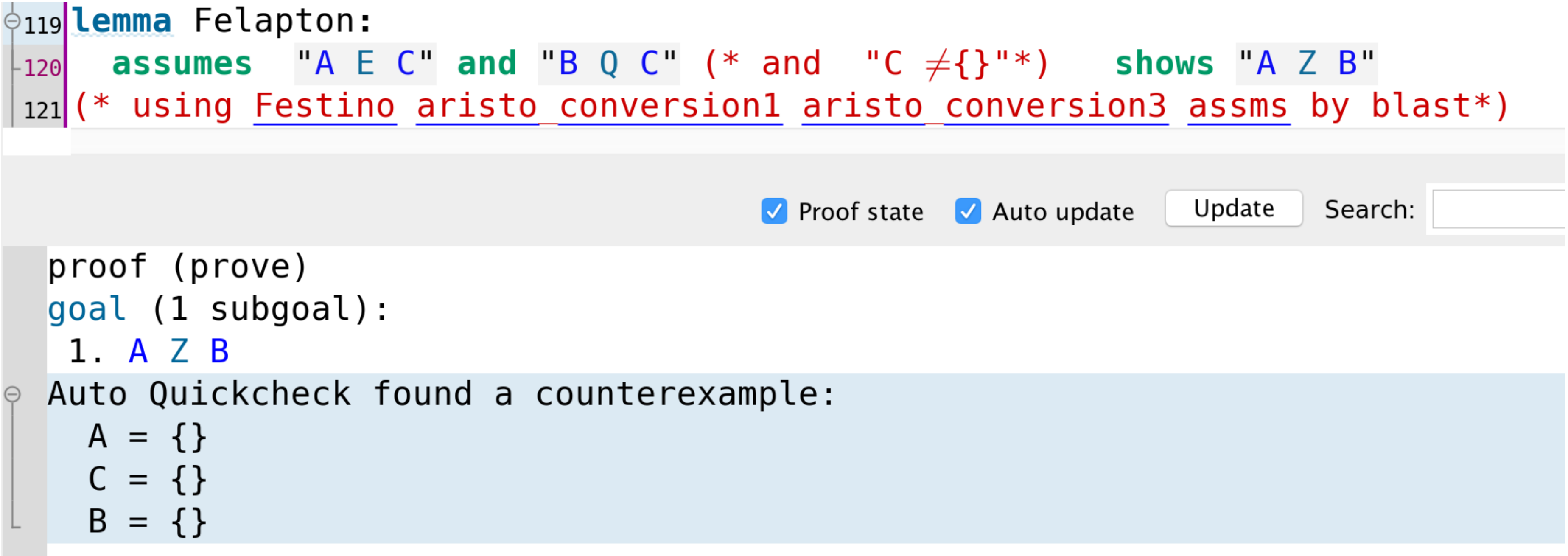


```
lemma Baroco_reducedto_Barbara :
  assumes "A Q B " and " A Z C  " and " B Q C "
  shows "B Z C" (*assms , concl. of Baroco and  B Q C *)
proof-
  have "A Q C" using  ‹A Q B ›  ‹ B Q C › Barbara by blast
  show ?thesis using  ‹A Q C›  ‹ A Z C ›
    by (simp add: particular_denial_def universal_affirmation_def)
qed

lemma Bocardo_reducedto_Barbara :
  assumes " A Z C" and "B Q C" and "A Q B"
  shows "A Z B" (*assms, concl of Bocardo and A Q B *)
proof-
  have "A Q C" using  ‹B Q C›  ‹ A Q B› using Barbara by blast
  show ?thesis using  ‹A Q C› ‹ A Z C›
    by (simp add: particular_denial_def universal_affirmation_def)
qed
```

```
text‹Finally, it is already evident from the proofs in the previous subsection that :

 Darapti can be reduced to Darii.

 Felapton can be reduced to Festino.

 Disamis can be reduced to Darii.

 Datisi can be reduced to Disamis.

 Ferison can be reduced to Ferio. ›

text‹In conclusion, the aforementioned deductions have thus been shown to be reduced to either
Barbara or Celarent as follows:

Baroco  $\Rightarrow$ Barbara

Bocardo $\Rightarrow$ Barbara

Felapton $\Rightarrow$ Festino $\Rightarrow$ Ferio $\Rightarrow$ Cesare $\Rightarrow$ Celarent

Datisi $\Rightarrow$ Disamis $\Rightarrow$ Darii $\Rightarrow$ Camestres $\Rightarrow$ Cesare

Darapti $\Rightarrow$ Darii

Ferison $\Rightarrow$ Ferio
›
```

Observations

3) The assumption that sets at hand must be nonempty is picked up by Isabelle’s counterexample tools. (Example)



```
lemma Felapton:
  assumes  "A E C" and "B Q C" (* and  "C ≠{}"*)   shows "A Z B"
(* using Festino aristo_conversion1 aristo_conversion3 assms by blast*)
```

☑ Proof state ☑ Auto update

```
proof (prove)
goal (1 subgoal):
 1. A Z B
Auto Quickcheck found a counterexample:
  A = {}
  C = {}
  B = {}
```

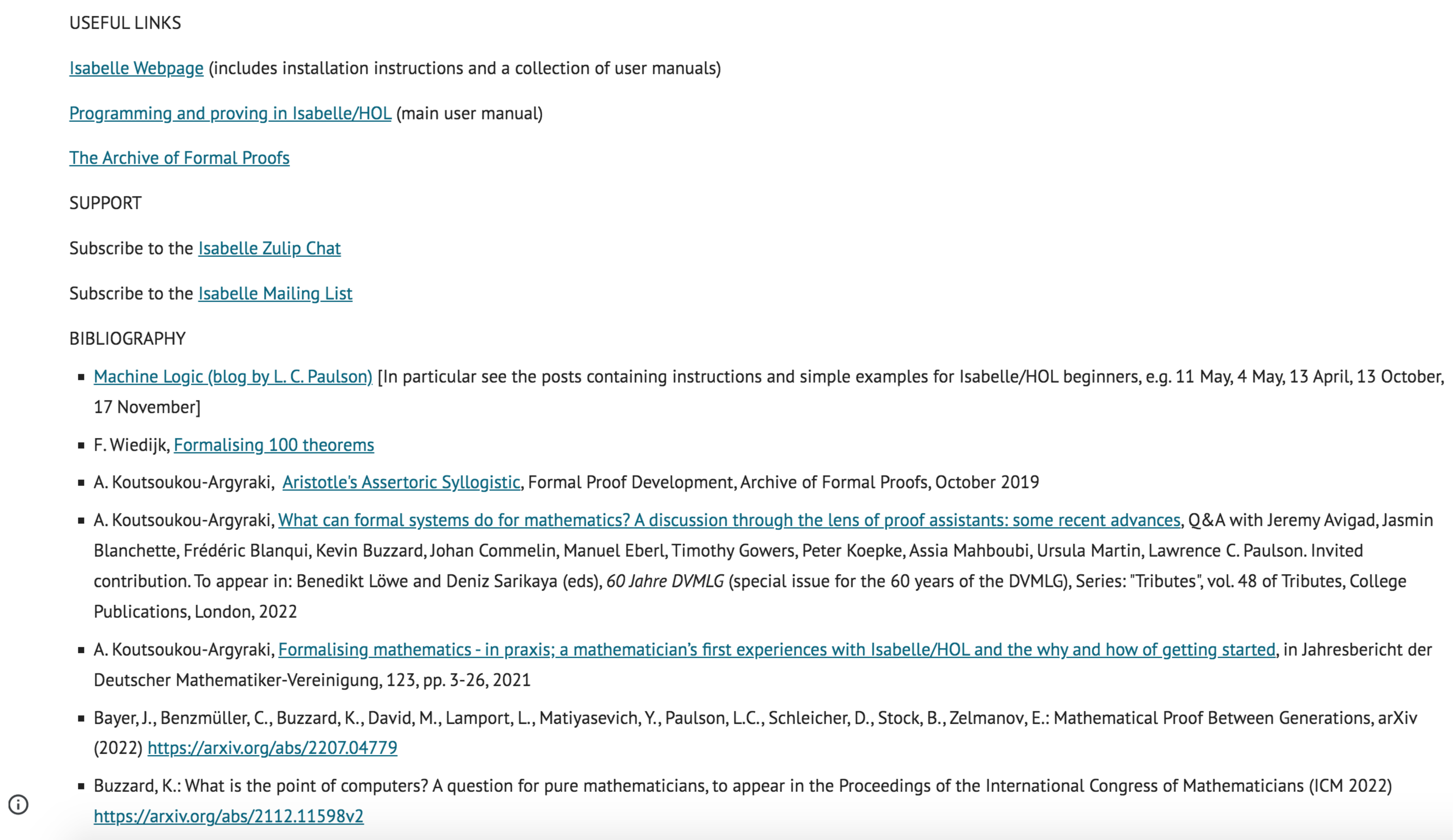
```
lemma Felapton:
  assumes  "A E C" and "B Q C"  and  "C ≠{}"  shows "A Z B"
  using Festino aristo_conversion1 aristo_conversion3 assms by blast
```

Topics for presentation

I) Explore the Archive of Formal Proofs.
Focus on developments of your choice according to your own interests.
Describe your experiences.

(II) Install Isabelle (optionally: also install the Archive of Formal Proofs) and experiment with basic examples of your choice according to your interests.

Suggested material for the topics



USEFUL LINKS

Isabelle Webpage (includes installation instructions and a collection of user manuals)

Programming and proving in Isabelle/HOL (main user manual)

The Archive of Formal Proofs

SUPPORT

Subscribe to the Isabelle Zulip Chat

Subscribe to the Isabelle Mailing List

BIBLIOGRAPHY

- Machine Logic (blog by L. C. Paulson) [In particular see the posts containing instructions and simple examples for Isabelle/HOL beginners, e.g. 11 May, 4 May, 13 April, 13 October, 17 November]
- F. Wiedijk, Formalising 100 theorems
- A. Koutsoukou-Argyraki, Aristotle's Assertoric Syllogistic, Formal Proof Development, Archive of Formal Proofs, October 2019
- A. Koutsoukou-Argyraki, What can formal systems do for mathematics? A discussion through the lens of proof assistants: some recent advances, Q&A with Jeremy Avigad, Jasmin Blanchette, Frédéric Blanqui, Kevin Buzzard, Johan Commelin, Manuel Eberl, Timothy Gowers, Peter Koepke, Assia Mahboubi, Ursula Martin, Lawrence C. Paulson. Invited contribution. To appear in: Benedikt Löwe and Deniz Sarikaya (eds), *60 Jahre DVMLG* (special issue for the 60 years of the DVMLG), Series: "Tributes", vol. 48 of Tributes, College Publications, London, 2022
- A. Koutsoukou-Argyraki, Formalising mathematics - in praxis; a mathematician's first experiences with Isabelle/HOL and the why and how of getting started, in Jahresbericht der Deutscher Mathematiker-Vereinigung, 123, pp. 3-26, 2021
- Bayer, J., Benzmüller, C., Buzzard, K., David, M., Lamport, L., Matiyasevich, Y., Paulson, L.C., Schleicher, D., Stock, B., Zelmanov, E.: Mathematical Proof Between Generations, arXiv (2022) https://arxiv.org/abs/2207.04779
- Buzzard, K.: What is the point of computers? A question for pure mathematicians, to appear in the Proceedings of the International Congress of Mathematicians (ICM 2022) https://arxiv.org/abs/2112.11598v2

Thank you!