Proof assistants for teaching proof and proving, motivations and challenges

ITI IRMIA++ Seminar - 2024

Julien Narboux

March 2024

What is a proof assistant ?

A software which allows to :

- define mathematical concepts/programs
- mechanically check proofs of theorems/programs

What is a proof assistant ?

A software which allows to :

- define mathematical concepts/programs
- mechanically check proofs of theorems/programs

What it is not ?

- An automated theorem prover
- A tool that helps in finding the proofs

History of proof assistants

- ACL Boyer-Moore (75-)
- LCF Milner (72)
- Automath De Bruijn (67)
- Mizar Trybulec (73-)
- Isabelle Paulson (86-)
- Coq Huet-Coquand (84-) (ACM Software System Award 2013)
- ...
- Lean Moura (2013)

Languages to describe proofs

Imperative We give orders (called tactics) to complete the proof tree

Declarative The proof is a sequence of mathematical assertions and their justification.

```
example (p q r : Prop) : p ∧ (q ∨ r) ↔ (p ∧ q) ∨ (p ∧ r) :=
begin
  apply iff.intro,
    intro H,
    apply (or.elim (and.elim_right H)),
      intro Hq,
      apply or.intro_left,
      apply and.intro,
        exact (and.elim_left H),
      exact Hq,
    intro Hr,
    apply or.intro_right,
    apply and.intro,
    exact (and.elim_left H),
    exact Hr,
  intro H,
  apply (or.elim H),
    intro Hpq,
    apply and.intro,
      exact (and.elim_left Hpq),
    apply or.intro_left,
    exact (and.elim_right Hpq),
  intro Hpr,
  apply and.intro,
    exact (and.elim_left Hpr),
  apply or.intro_right,
  exact (and.elim_right Hpr)
end
```

```
theorem sqrt_prime_irrational:
  fixes p :: nat
  assumes "prime p"
  shows "sqrt p ∉ ℚ"
proof
  from ‹prime p› have p: "p > 1" by (rule prime_gt_1_nat)
  assume "sqrt p ∈ ℚ"
  then obtain m n :: nat
    where n: "n ≠ 0"
      and sqrt_rat: "¦sqrt p¦ = m / n"
      and "coprime m n" by (rule Rats_abs_nat_div_natE)
  have eq: "m^2 = p * n^2"
  proof -
    from n and sqrt_rat have "m = ¦sqrt p¦ * n" by simp
    then have "m^2 = (sqrt p)^2 * n^2" by (simp add: power_mult_distrib)
    also have "(sqrt p)^2 = p" by simp
    also have "... * n^2 = p * n^2" by simp
    finally show ?thesis by linarith
  qed
  have "p dvd m ∧ p dvd n"
  proof
    from eq have "p dvd m^2" ..
    with ‹prime p› show "p dvd m" by (rule prime_dvd_power)
    then obtain k where "m = p * k" ..
    with eq have "p * n^2 = p^2 * k^2" by algebra
    with p have "n^2 = p * k^2" by (simp add: power2_eq_square)
    then have "p dvd n^2" ..
    with ‹prime p› show "p dvd n" by (rule prime_dvd_power)
  qed
  then have "p dvd gcd m n" by simp
  with ‹coprime m n› have "p = 1" by simp
  with p show False by simp
qed
```

Some success of proof assistants

- CompCert (X. Leroy, ACM Award 2022)
- seL4 Micro Kernel (ACM Award 2023)
- 4 Color theorem 2007, Feit-Thompson 2012 (Gonthier)
- Flyspeck 2015 (Hales)
- Polynomial Freiman-Ruzsa 2023 (Tao)

Motivations for using a proof assistant ?

- immediate feedback : forbids non-sense statements, forbids incorrect reasoning, detect scope and freshness errors,. . .
- clarify role of statements : axiom, lemma, hypothesis, premise, conclusion, goal, . . .
- clarify theoretical status [1] (conjecture, consequence of hypotheses, sufficient condition to obtain the goal, . . .) and operational status : premise/conclusion
- clarify allowed logical rules
- impartiality
- micro-world : definitions and allowed theorems
- clarify what is a proof forbidding other types of argumentation
- *gamification*

1. Duval's terminology

Different potential goals

- To teach what is a proof
- To teach logic
- To teach software foundations
- To automate proof-checking
- To teach maths in general

Existing tools

Two communities :

1. Didactics of mathematics
2. Interactive theorem proving

ITP Community

Computer Science

- logic
- proof of programs, semantics, software foundations

U-Penn, Portland, Princeton, Harvard, Warsaw, CNAM, Lyon, Nice, Paris, Strasbourg, ...

Maths

- Bachelor - Maths : Nijmegen (ProofWeb), Nice (CoqWeb), Chambery (Phox), Paris (Lean), Strasbourg (Edukera/LeanVerbose/Deaduction)...
- ...

My experiments

M2 CS : Semantics, Hoare's logic, formalized in Coq.

M1 CS : Formalization of proofs in geometry.

M1 CS : Course about formal theorem with/about Coq.

L2 CS : Course about logic, Edukera in Logic mode (natural deduction)

L1 Maths and L1 CS : Introduction to proof

My experiments

M2 CS : Semantics, Hoare's logic, formalized in Coq.

M1 CS : **Formalization of proofs in geometry**.

M1 CS : Course about formal theorem with/about Coq.

L2 CS : Course about logic, Edukera in Logic mode (natural deduction)

L1 Maths and L1 CS : **Introduction to proof**

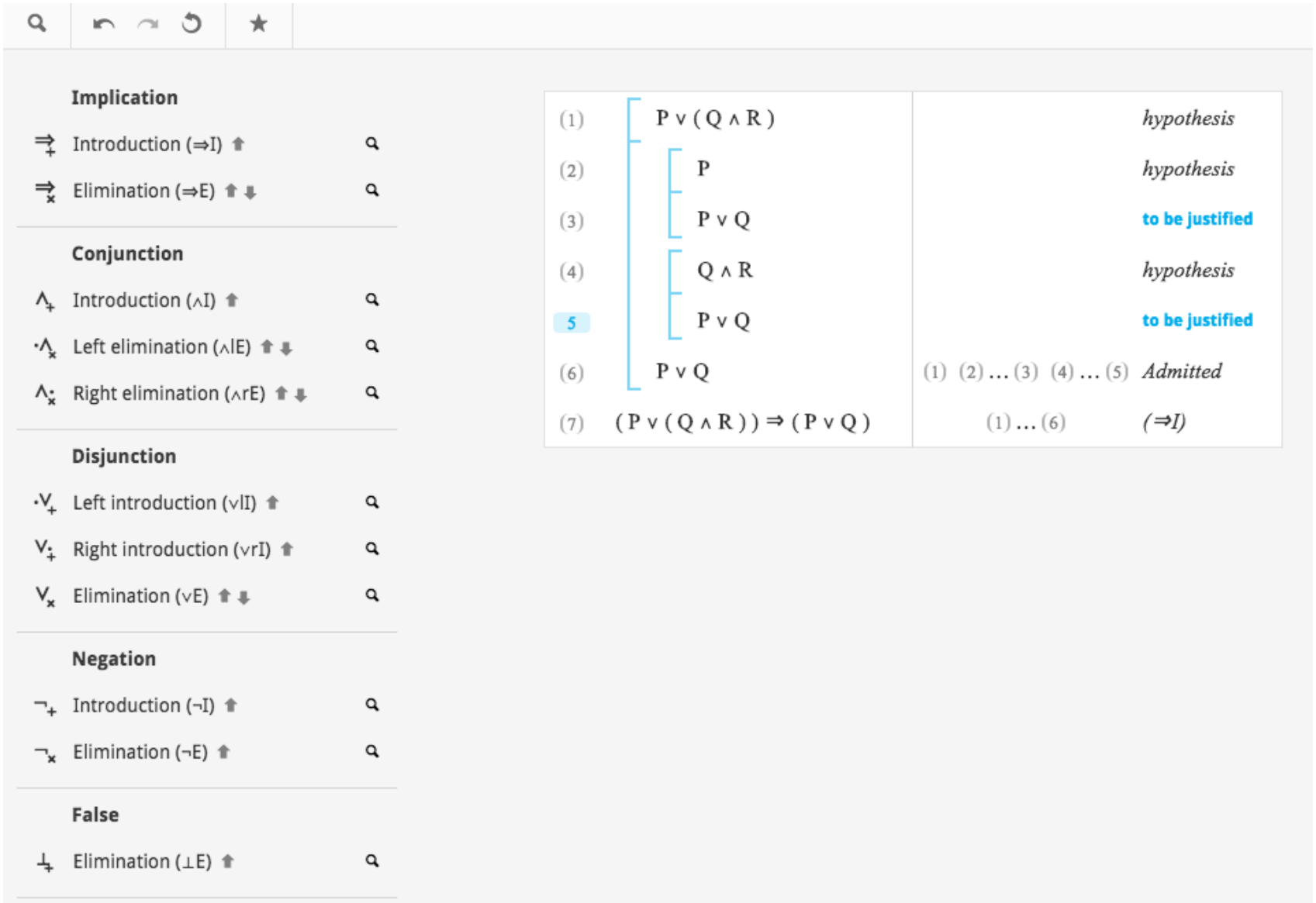
Edukera (Rognier and Duhamel)

- Web-application
- Coq is hidden inside the web page
- LCF style interaction + proof displayed in a pen and paper style.
- Some users in France (about 1000 students, 70k exercises)
- No textual input "proof by pointing", syntactically correct by construction (as using Scratch)
- Easy to learn using a tutorial
- Always correct applications of a logic rule
- Meta-variables

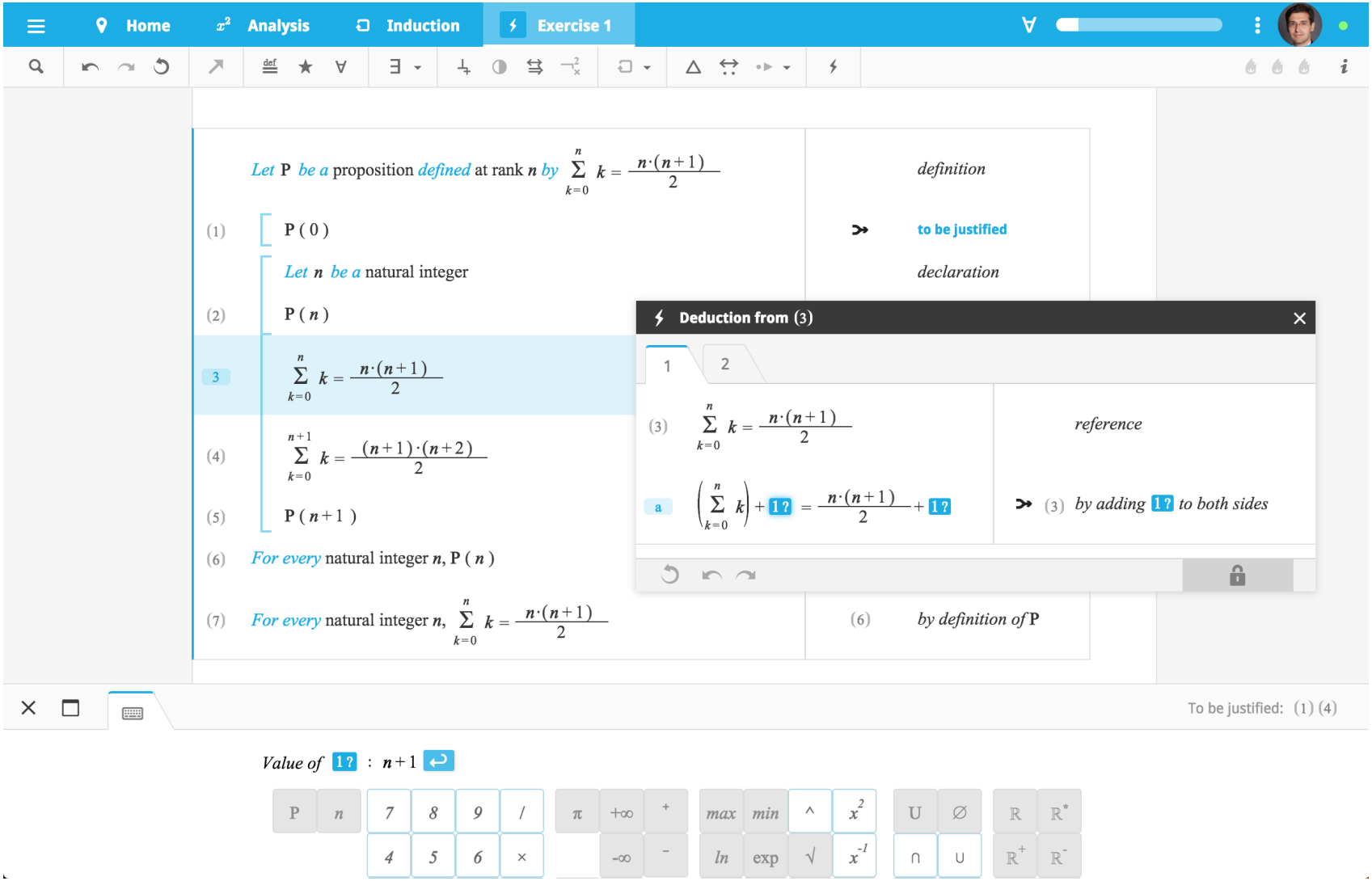
Two modes

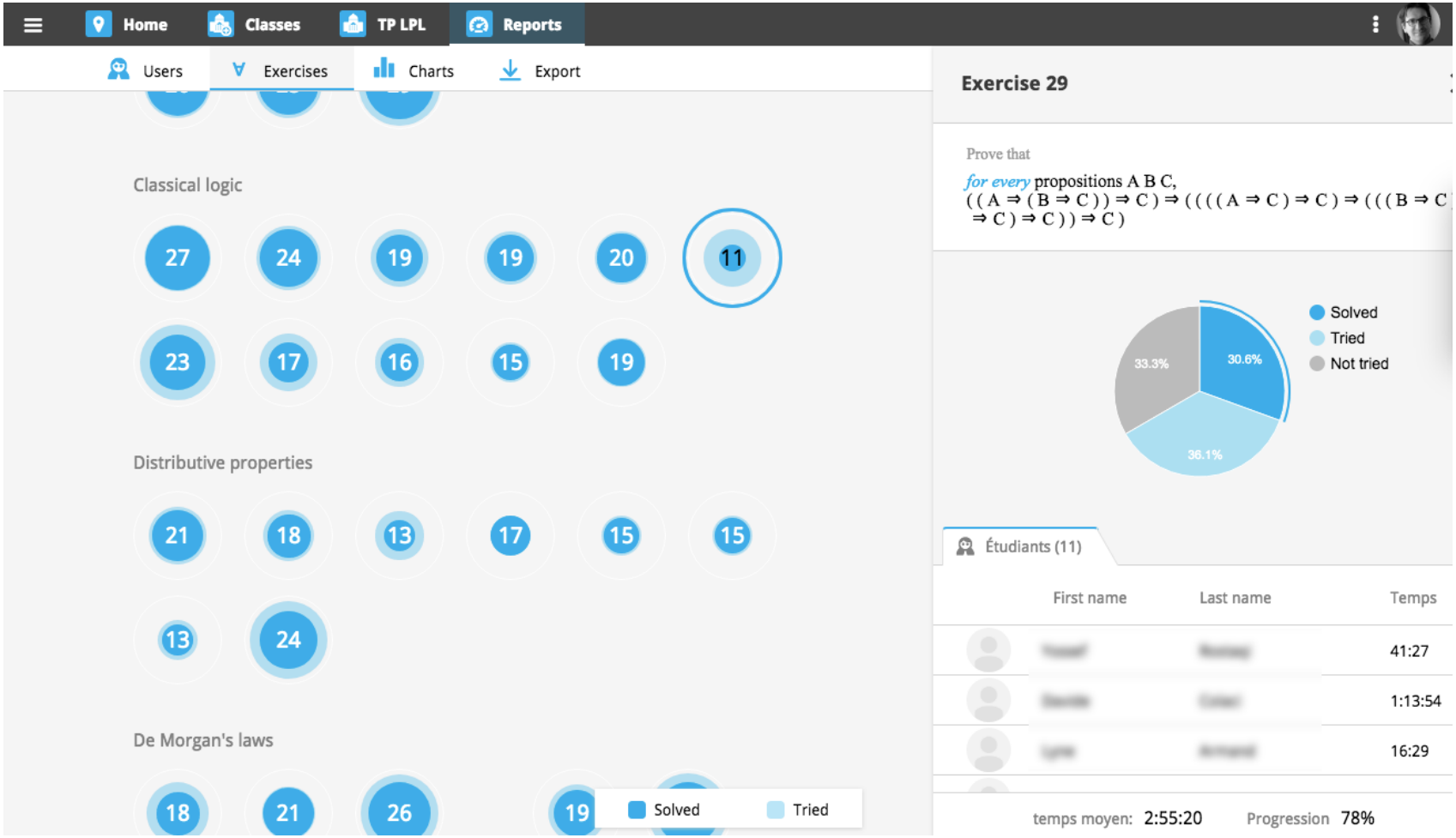
1. Logic
   - Use natural deduction rules.
   - Can display proof tree (Fitch's or Gentzen's style).
   - Backward reasoning
2. Maths
   - Forward/Backward reasoning.
   - Less fine-grained proof steps than in logic mode.

Edukera (logic mode)



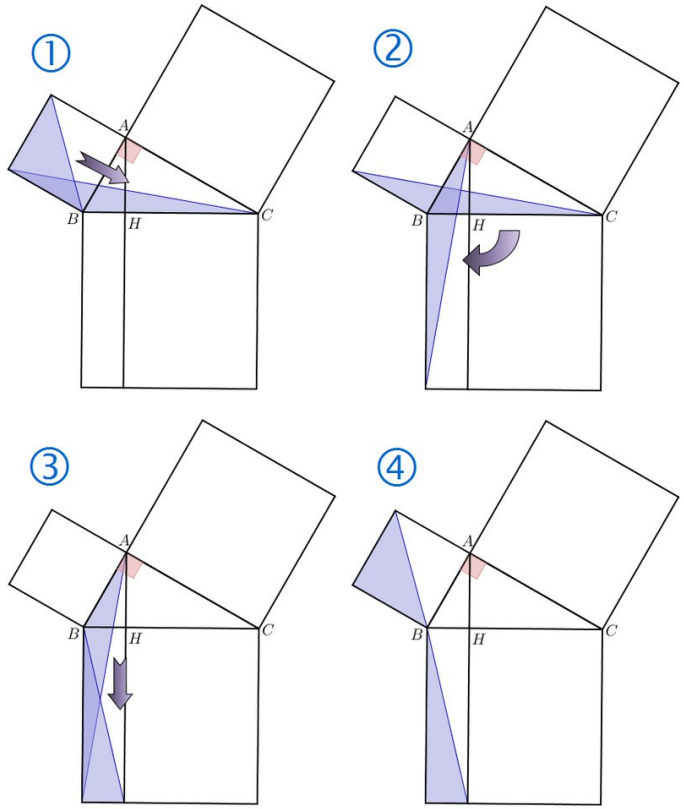
Edukera (math mode)




Home
Classes
TP LPL
Reports
Users
Exercises
Charts
Export
Classical logic
27
24
19
19
20
11
23
17
16
15
19
Distributive properties
21
18
13
17
15
15
13
24
De Morgan's laws
18
21
26
19
Solved
Tried
Exercise 29
Prove that
for every propositions A B C,
((A ⇒ (B ⇒ C)) ⇒ C) ⇒ ((((A ⇒ C) ⇒ C) ⇒ (((B ⇒ C) ⇒ C) ⇒ C)) ⇒ C)
Solved
Tried
Not tried
33.3%
30.6%
36.1%
Étudiants (11)
First name
Last name
Temps
41:27
1:13:54
16:29
temps moyen: 2:55:20
Progression 78%

Experiment at master lelvel

1. Choose a proof Pythagoras's theorem and propose a formalization.



Discussions :

- validity of proof
- proofs as explanation
- simplicity

- visual proofs
- circularity
- genericity (proof in a model, vs proof in every model)

Experiment with first-year students : Introduction to proof and proving

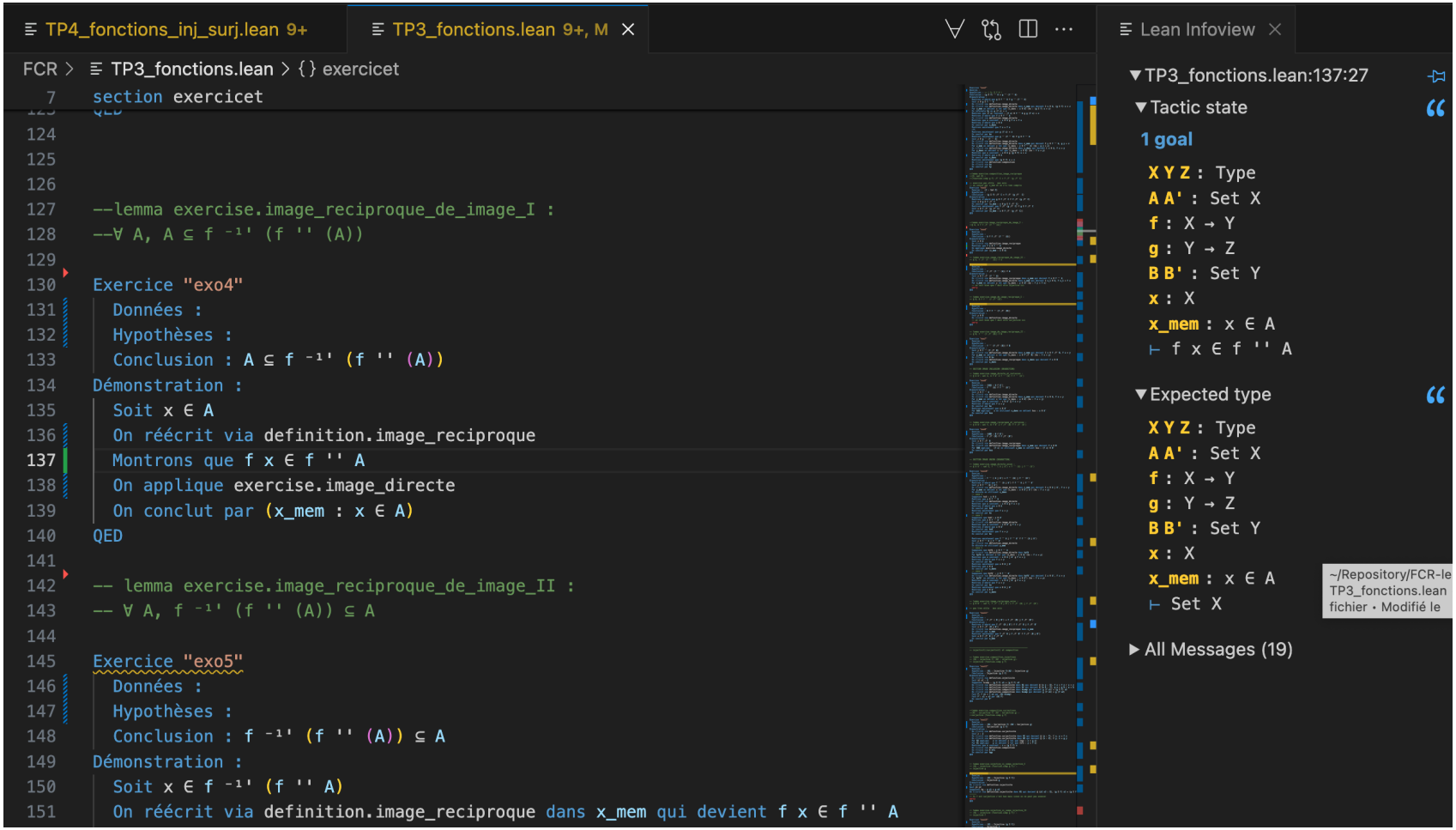
- L1 Maths and L1 CS, selected/not selected student.
- 7 groups.
- Mixed teaching team : high-school teachers, PhD students, lecturer in maths, lecturer in CS.
- Contents : vocabulary and structure of a mathematical document. Reasoning rules (without formality). Sets, Functions (direct image, inverse image, injective, surjective), Relations, induction.

Pre-experiment in Strasbourg in 2024

- Objective : introduction to proof
- 3 groups with Deaduction by Frédéric Le Roux
- 3 groups with Lean-Verbose by Patrick Massot
- On paper assessment.

Current work :

- Construction of a corpus of exercises (proofs, true/false, formalization).
- Improving software, bug reporting.
- Instrumentation of proof assistants to collect date.


TP4_fonctions_inj_surj.lean 9+
TP3_fonctions.lean 9+, M
FCR > TP3_fonctions.lean > { } exercicet
7 section exercicet
124
125
126
127 --lemma exercise.image_reciproque_de_image_I :
128 --∀ A, A ⊆ f ⁻¹' (f '' (A))
129
130 Exercice "exo4"
131 Données :
132 Hypothèses :
133 Conclusion : A ⊆ f ⁻¹' (f '' (A))
134 Démonstration :
135 Soit x ∈ A
136 On réécrit via definition.image_reciproque
137 Montrons que f x ∈ f '' A
138 On applique exercise.image_directe
139 On conclut par (x_mem : x ∈ A)
140 QED
141
142 -- lemma exercise.image_reciproque_de_image_II :
143 -- ∀ A, f ⁻¹' (f '' (A)) ⊆ A
144
145 Exercice "exo5"
146 Données :
147 Hypothèses :
148 Conclusion : f ⁻¹' (f '' (A)) ⊆ A
149 Démonstration :
150 Soit x ∈ f ⁻¹' (f '' A)
151 On réécrit via definition.image_reciproque dans x_mem qui devient f x ∈ f '' A
Lean Infoview
TP3_fonctions.lean:137:27
Tactic state
1 goal
X Y Z : Type
A A' : Set X
f : X → Y
g : Y → Z
B B' : Set Y
x : X
x_mem : x ∈ A
⊢ f x ∈ f '' A
Expected type
X Y Z : Type
A A' : Set X
f : X → Y
g : Y → Z
B B' : Set Y
x : X
x_mem : x ∈ A
⊢ Set X
~/Repository/FCR-le
TP3_fonctions.lean
fichier • Modifié le
All Messages (19)

```
Exercice "exo1"
  Données :
  Hypothèses :
  Conclusion : VraiFaux (∀ n : ℕ, (n = 0) ∨ (n = 1))
Démonstration :
  Montrons que ¬ ∀ n : ℕ, (n = 0) ∨ (n = 1)
  On pousse la négation
  Montrons que 3 convient : (3 ≠ 0) ∧ (3 ≠ 1)
  On calcule
QED

-- Plus petit que tous
-- Il existe m dans ℕ tel que pour tout n dans ℕ, m ≤ n.

Exercice "exo2"
  Données :
  Hypothèses :
  Conclusion : VraiFaux (∃ m : ℕ, (∀ n : ℕ, m ≤ n))
Démonstration :
  Montrons que ∃ m : ℕ, (∀ n : ℕ, m ≤ n )
  Montrons que 0 convient : ∀ (n : ℕ), 0 ≤ n
  Soit n
  On calcule
QED
```

In previous work [2] we studied one exercise using different PAs (Coq, Deaduction, Edukera, Lean (Verbose), Lurch) and we provided an *a priori* study of the potential impact of different features.

Exercise

Étant donnés trois ensembles A, B et C tels que $C \subseteq A$ et une fonction $f : A \to B$, montrer que si f est injective alors $f^{-1}(f(C)) = C$.

Some videos

2. Evmorfia Bartzia, Antoine Meyer et Julien Narboux (oct. 2022). "Proof assistants for undergraduate mathematics and computer science education : elements of a priori analysis". In : *INDRUM 2022 : Fourth conference of the International Network for Didactic Research in University Mathematics*. Sous la dir. de María Trigueros. Hanovre, Germany : Reinhard Hochmuth.

automatic unfolding of definitions

- fake success ? ($\rightarrow$ Lean Verbose / Coq Waterproof)
- real success since they get a proof ?

interaction style undo/redo, and automatic application of reasoning rules can lead to a trial/error strategy

Research in didactic ?

Very few results about proof assistants (Iannone, Hanna, Richard, Thoma).
Since 2024, ANR APPAM lead by Cécile Ouvrier-Buffet : didactics + maths+ computer science.
Study of student's difficulty while learning proof, impact of the use of proof assistants.

Some questions

- What competencies are exercised using PAs ?
- Does PAs improve habits ?
- Is the syntax a problem ?
- Syntax vs semantics ?
- How to transfer to proofs on paper ?
- What is the impact of foundations of PAs on student perception of proof ?

Do you want to know more ?

A thematic school : PAT

PAT 2023 : first thematic school about proof assistant for teaching, 38 participants.
PAT 2025 ? (project submitted to CNRS)

Workshops : ThEdu

ThEdu 2024, July 2, IJCAR, Nancy.

Bibliography

A Zotero group :
`www.zotero.org/groups/2621881/`