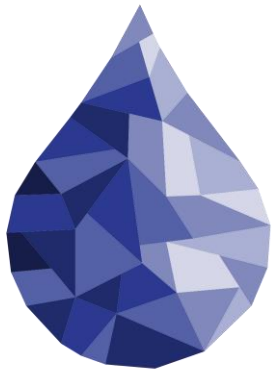


WATERPROOF

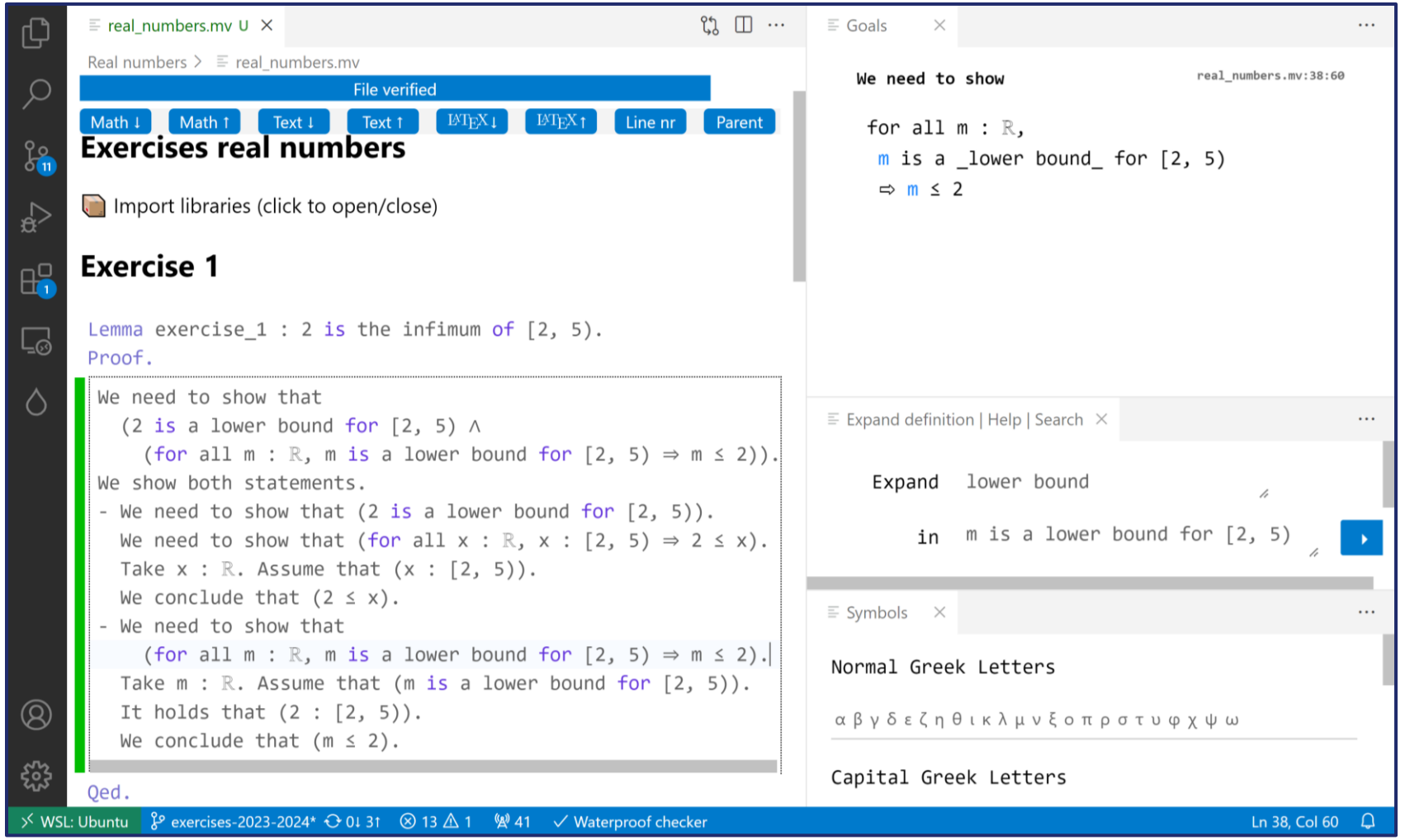
Introduction and Wishlist

Jim Portegies & Jelle Wemmenhove

27 February 2023

TU/e
EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Waterproof



https://impermeable.github.io/

Waterproof – design principle

Writing a proof in Waterproof should be as close as possible to writing a proof with pen and paper, both in terms of final result and process

Challenges and cooperation

The educational aspect brings interesting challenges, that do not come up with pure formalization

We'd like to collaborate with LiberAbaci to solve shared problems, and exchange knowledge and experiences

We are also starting a new PhD project later this year, in cooperation with Paige North and Johan Commelin, to also look into these issues

Challenges

Implicit automation

- Proofs driven by forward reasoning statements, like

> "It holds that $(\forall x: \mathbb{R}, x^2 \geq 0)$."

- Assertions implicitly checked by automation
- Implementation calls own version auto tactic
- Tunable with collections of hint databases
- Shielding of complicated statements starting with logical operator
- Heavy use of lra tactic

Implicit automation – rewrite issue

- Our automation procedure struggles with rewrites

f : $\mathbb{R} \to \mathbb{R}$ such that Hf : $\forall$a:$\mathbb{R}$, f(a) = a and x,y,z,u,v,w : $\mathbb{R}$

with Hf added to hint database

"It holds that (f(x) + y + z + u + v + w = x + y + z + u + v + w)." fails

- Should be easy: "rewrite Hf; reflexivity."
- No simple equivalent for "rewrite" in natural language
- Some libraries are designed for heavy use of rewrite tactic

Chain of (in)equalities

- Rewrite-issue often comes up with chains of (in)equalities

> "It holds that (& $25 = (-5)^2 < x^2 < z$)."

- Similar issues with inequalities
- lra very powerful, but does not work well together with lemmas in hint databases
- example Hf : ∀a:ℝ, f(a) < a

Chain of (in)equalities – support?

"It holds that ($\& 25 = (-5)^2 < x^2 < z$)."

- Own implementation, probably not optimal
- Official support?
- **Ideal** optional justification per (in)equality

User specified lemma

"By Heine_Borel it holds that ($A \subset \mathbb{R}$ is compact)."

- Implementation calls own version auto using tactic
- Intention: enforce specification lemmas
- Practice: students also use hypotheses from local context

User specified lemma – feedback

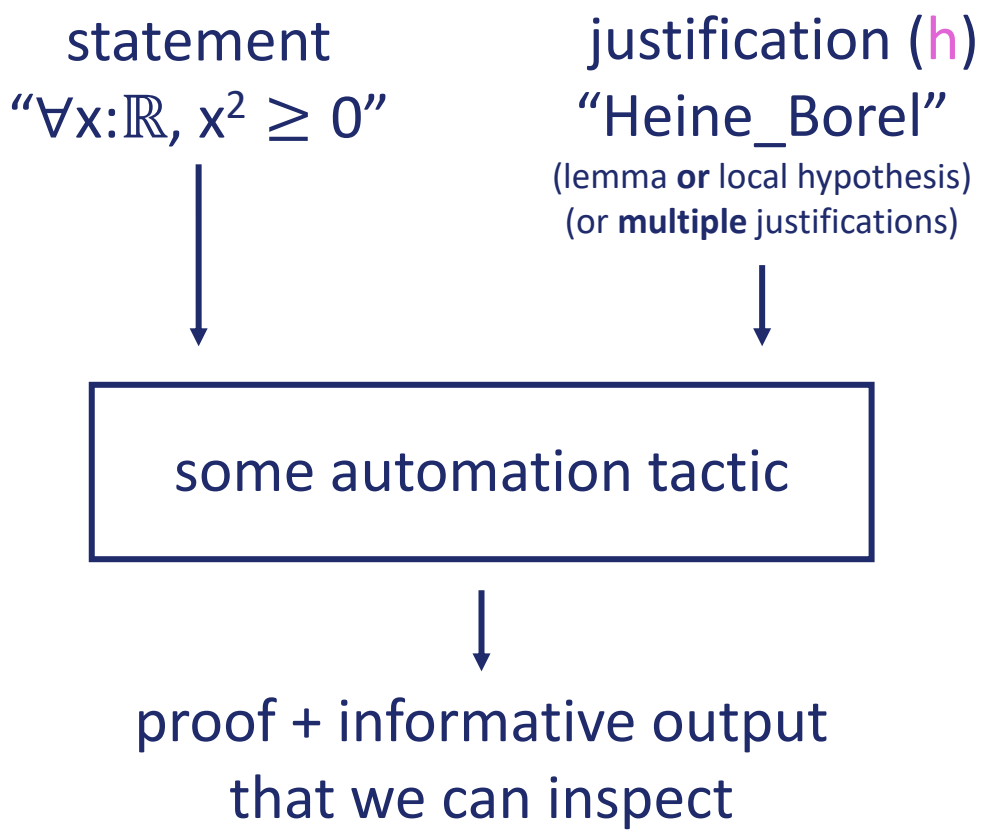
- Binary output from auto (success or fail) is not sufficient

> "By Heine_Borel it holds that (1 + 1 = 2)."
>
> proof found, but it does not use specified lemma

- Proof has to use specified lemma
- If it fails, why? --- user wants to apply this lemma!

> "By Heine_Borel it holds that ((0,1) ⊂ ℝ is compact)."
>
> lemma fails because a precondition ((0,1) ⊂ ℝ is closed) could not be shown

User specified lemma - feedback



Subsets like in normal math

- Don’t bother students with coercions or classifying predicates
- Encoding of ZF set theory in type theory not a solution
  - we want to be able to hook into existing libraries like math-comp
- Yves Bertot: numbers as nested subsets $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$

Subsets like in normal math

- We want something like this:

  - A : subset X := {x : X | P(x) }
  - a : A
  - from x : A to x : X such that P(x)
  - from x : X such that P(x) to x : A

  example

- Student project: solution via notation and tactics?

Subsets like in normal math - example

- from $x : X$ such that $P(x)$ to $x : A$
- We want to be able to do this:

Goal: $\exists y : [0,1], P(y)$

"Choose y := 1/2."

"Then indeed $(0 \leq y \leq 1)$."

... continue to prove P(y)

Naming dummy variables

- Reusing names for dummy variables is fine in Coq
- Confusing for students
- Students think same name refers to same variable

> “Take N : $\mathbb{N}$.”
>
> “It holds that ($\exists$N:$\mathbb{N}$, N > 10).”

Naming dummy variables

- We want to discourage reasoning within quantifiers

> "It holds that ($\exists x:\mathbb{R}$, $|x| < 10$)."
>
> "It holds that ($\exists x:\mathbb{R}$, $-10 < x < 10$)."

- Possible solution: not allow repeating dummy variable?
- (need to think about this some more)

Miscellaneous

- Notation used in practice
  - example: “lim_{n → ∞} n = ∞”
- Library for education (first-year bachelor students)
  - ‘reserved notation’ should not reserve too much
- Documentation
  - Current documentation quite high level
  - Easier development custom tactics and custom plugins

Wishlist overview

- Mathematical library for education, possibly including tactics
- Waterproof can use this library with little interfacing
- Library plays together nicely with Waterproof style automation
  - special care around rewrite
  - combine lra with inequality-lemmas in hint databases
  - high-quality feedback from automation procedure, including when user specifies lemma that is to be used
- Library uses common mathematical notation
- Subsets like in normal math, and $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$
- Avoid educationally-problematic naming of dummy variables
- Native support for chains of (in)equalities
- Non-expert level documentation