Lean dev update

June 15 2022

Goals

Get feedback from *you*!

Introduce new team members.

Explain our roadmap.

Announce new features.

Welcome Sarah Smith

Sarah Smith is the new Program Manager for the Lean project.

Task forces

Lean 4 maintenance and bug fixes - Sebastian and Leo

Proof automation - Leo

Typed Macros - Sebastian

Code formatter - Lars

Lake - Mac

LSP & Widgets - Wojciech and Ed

Mathlib 4 - Gabriel, Mario, Ed, Arthur, Henrik, …

VS Code - Wojciech, Chris Lovett, Mariana, and Gabriel

Doc-gen4 - Henrik

Documentation

Functional Programming in Lean - David Christiansen

Lean 4 Metaprogramming - Arthur, Henrik, Siddhartha, Damiano, Jannis, …

We want to improve our documentation and website, and support more efforts.

Common infrastructure for writing all Lean documentation:

LeanInk + Doc-Gen4 + ???

Roadmap

Lean 4 first official release at the end of the summer:

- Typed Macros
- Linear arithmetic, Congruence closure + E-matching
- Widgets?
- Lake with Cloud build support?
- New tag: `lean4_release`, better tags for communicating intent?

Mathlib port - Push after Lean ICERM workshop

Roadmap cont.

After the official release the focus will continue on automation and Mathlib port.

Rob Lewis will be teaching a Discrete Math Course on top of Lean in Jan. 2023.

Goal: make sure the course will be a tremendous success.

Make sure all tactics used in Patrick Massot's course are available in Lean 4.

Module System (Sebastian): better support for staging.

Feb. 2023 - We are going to port the remaining parts of the code generator to Lean. Focus on fixing bugs, missing optimizations, and new features.

Fast reduction engine in the kernel (better support for proofs by reflection).

Stability

Modules that will be heavily modified:

- Runtime: we will add support for stack allocated result values.
- Code generator: C/C++ code will be ported to Lean.
- Do.lean: it is going to be rewritten using the approach in our ICFP paper
  - It depends on the new code generator.
- Parser: tokens should not be global, a must-have feature for DSLs
- Structural: must be extended to support mutual/nested inductives

Questions? Concerns?