Searching for Proof Improvements with **TryAtEachStep**

David Renshaw
Lean Together
January 2025

1. **origin story**
2. big picture
3. implementation
4. results
5. future work

February 2021

PROOF ARTIFACT CO-TRAINING FOR THEOREM PROVING WITH LANGUAGE MODELS

Jesse Michael Han
University of Pittsburgh
OpenAI

Jason Rute
IBM Research*

Yuhuai Wu
Google Research
Stanford University†

Edward W. Ayers
Carnegie Mellon University‡

Stanislas Polu
OpenAI

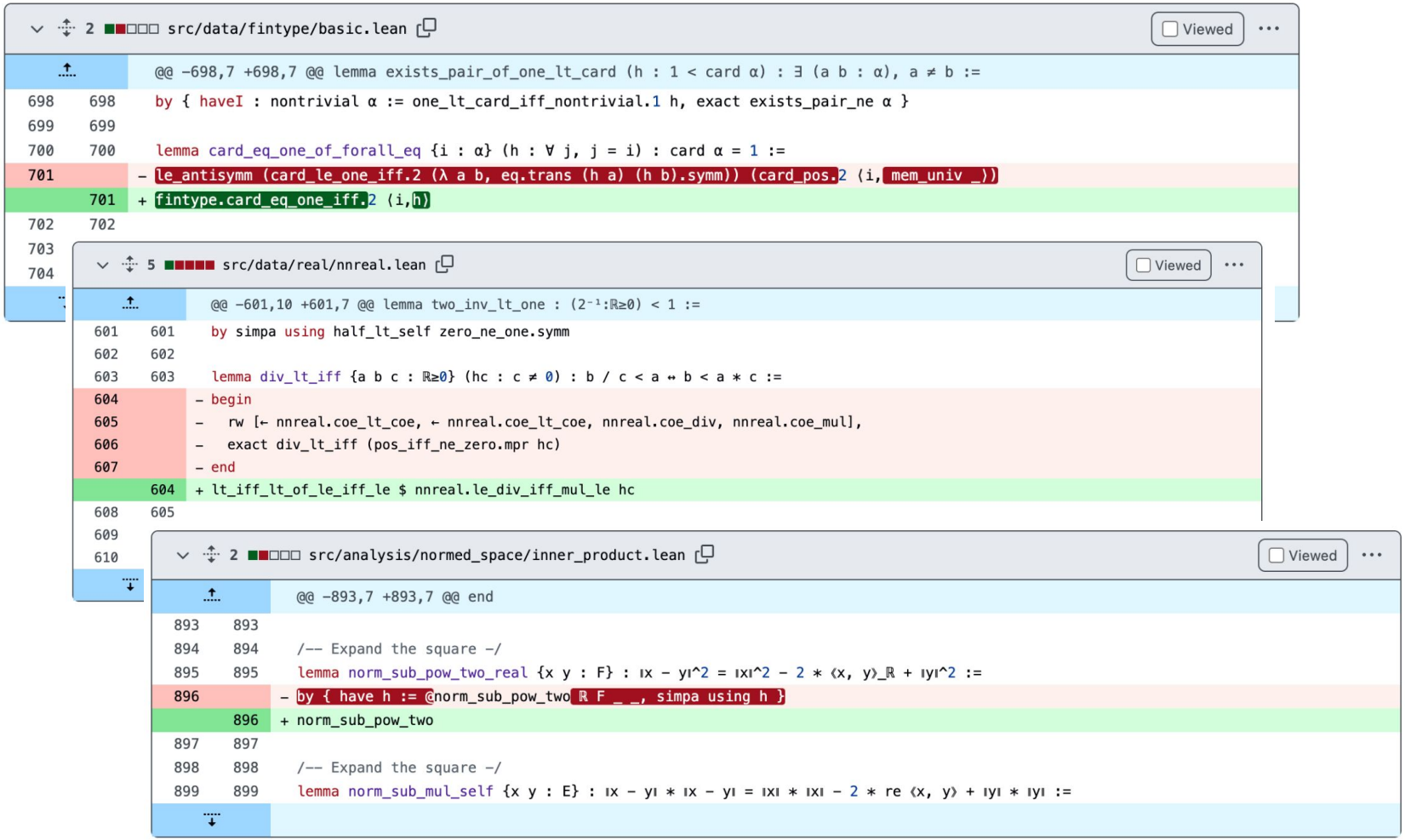
jesse-michael-han commented on Feb 9, 2021 — Collaborator

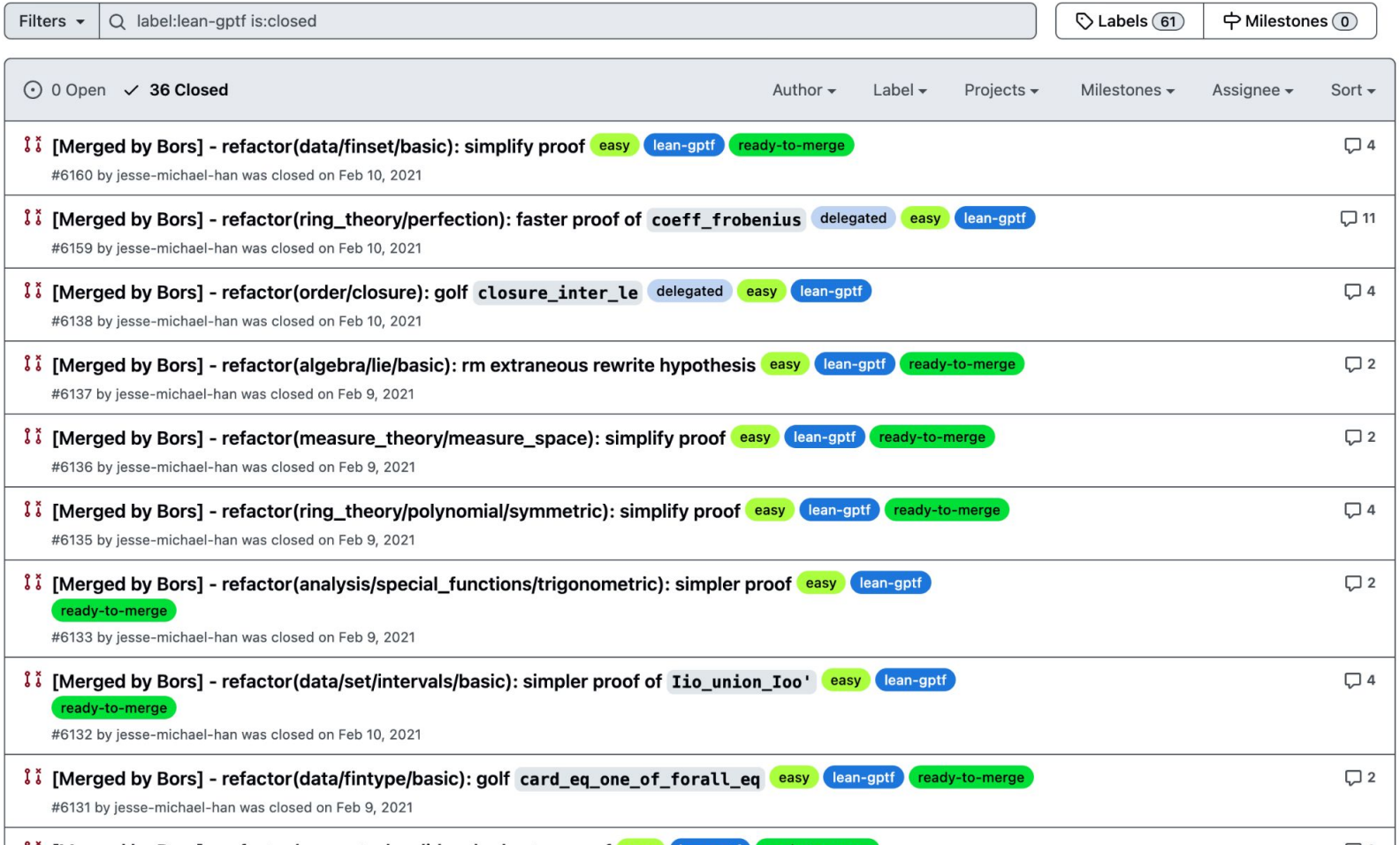
Co-authors: `lean-gptf`, Stanislas Polu

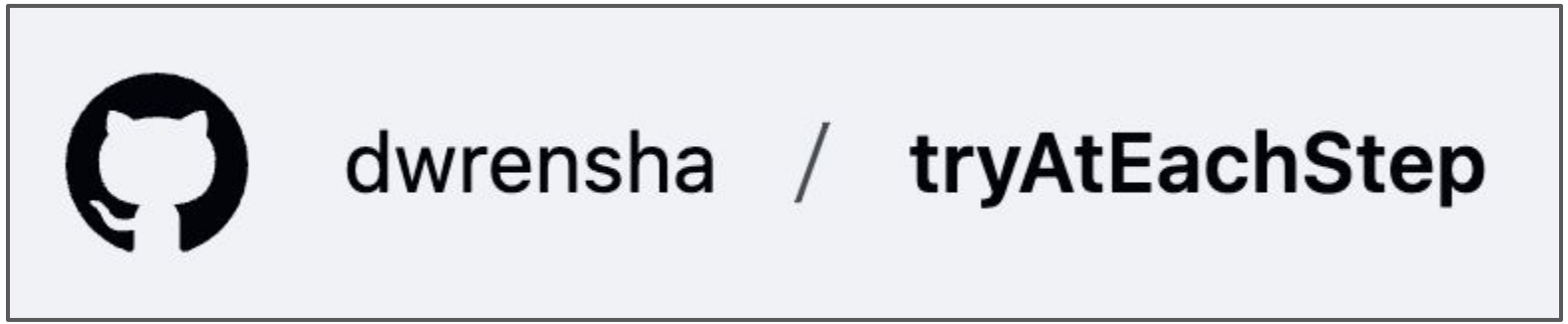
This was found by `formal-lean-wm-to-tt-m1-m2-v4-c4` when we evaluated it on theorems added to `mathlib` after we last extracted training data.

10 ■■■■□ src/algebra/lie/basic.lean — Viewed

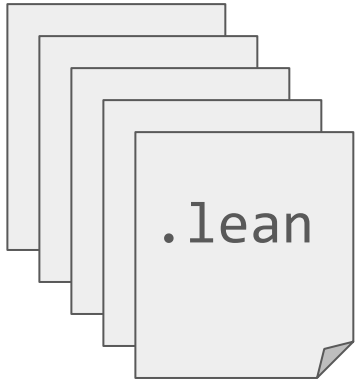
```
@@ -1577,15 +1577,7 @@ begin
 end

 @[simp] lemma map_bot_iff : I.map f = ⊥ ↔ I ≤ f.ker :=
-begin
-  rw le_ker_iff, unfold lie_ideal.map, split; intros h,
-  { rwa [eq_bot_iff, lie_submodule.lie_span_le, set.image_subset_iff, lie_submodule.bot_coe] at h,},
-  { suffices : f '' I = ↑(⊥ : lie_ideal R L'), { rw [this, lie_submodule.lie_span_eq], },
-    ext x, rw [lie_submodule.bot_coe, set.mem_singleton_iff, set.mem_image],
-    split,
-    { rintros (y, hy, hx), rw ← hx, exact h y hy, },
-    { intros hx, use 0, simp [hx], }, },
-end
+by { rw ← le_bot_iff, apply lie_ideal.map_le_iff_le_comap }

 lemma ker_eq_bot : f.ker = ⊥ ↔ function.injective f :=
 by rw [← lie_submodule.coe_to_submodule_eq_iff, ker_coe_submodule, lie_submodule.bot_coe_submodule,
```


2 src/data/fintype/basic.lean
Viewed
@@ -698,7 +698,7 @@ lemma exists_pair_of_one_lt_card (h : 1 < card α) : ∃ (a b : α), a ≠ b :=
698 698 by { haveI : nontrivial α := one_lt_card_iff_nontrivial.1 h, exact exists_pair_ne α }
699 699
700 700 lemma card_eq_one_of_forall_eq {i : α} (h : ∀ j, j = i) : card α = 1 :=
701 - le_antisymm (card_le_one_iff.2 (λ a b, eq.trans (h a) (h b).symm)) (card_pos.2 ⟨i, mem_univ _⟩)
701 + fintype.card_eq_one_iff.2 ⟨i,h⟩
702 702
703
704
5 src/data/real/nnreal.lean
Viewed
@@ -601,10 +601,7 @@ lemma two_inv_lt_one : (2⁻¹:ℝ≥0) < 1 :=
601 601 by simpa using half_lt_self zero_ne_one.symm
602 602
603 603 lemma div_lt_iff {a b c : ℝ≥0} (hc : c ≠ 0) : b / c < a ↔ b < a * c :=
604 - begin
605 - rw [← nnreal.coe_lt_coe, ← nnreal.coe_lt_coe, nnreal.coe_div, nnreal.coe_mul],
606 - exact div_lt_iff (pos_iff_ne_zero.mpr hc)
607 - end
604 + lt_iff_lt_of_le_iff_le $ nnreal.le_div_iff_mul_le hc
608 605
609
610
2 src/analysis/normed_space/inner_product.lean
Viewed
@@ -893,7 +893,7 @@ end
893 893
894 894 /-- Expand the square -/
895 895 lemma norm_sub_pow_two_real {x y : F} : ∥x - y∥^2 = ∥x∥^2 - 2 * ⟪x, y⟫_ℝ + ∥y∥^2 :=
896 - by { have h := @norm_sub_pow_two ℝ F _ _, simpa using h }
896 + norm_sub_pow_two
897 897
898 898 /-- Expand the square -/
899 899 lemma norm_sub_mul_self {x y : E} : ∥x - y∥ * ∥x - y∥ = ∥x∥ * ∥x∥ - 2 * re ⟪x, y⟫ + ∥y∥ * ∥y∥ :=


Filters
label:lean-gptf is:closed
Labels 61
Milestones 0
0 Open
36 Closed
Author
Label
Projects
Milestones
Assignee
Sort
[Merged by Bors] - refactor(data/finset/basic): simplify proof easy lean-gptf ready-to-merge
#6160 by jesse-michael-han was closed on Feb 10, 2021
4
[Merged by Bors] - refactor(ring_theory/perfection): faster proof of coeff_frobenius delegated easy lean-gptf
#6159 by jesse-michael-han was closed on Feb 10, 2021
11
[Merged by Bors] - refactor(order/closure): golf closure_inter_le delegated easy lean-gptf
#6138 by jesse-michael-han was closed on Feb 10, 2021
4
[Merged by Bors] - refactor(algebra/lie/basic): rm extraneous rewrite hypothesis easy lean-gptf ready-to-merge
#6137 by jesse-michael-han was closed on Feb 9, 2021
2
[Merged by Bors] - refactor(measure_theory/measure_space): simplify proof easy lean-gptf ready-to-merge
#6136 by jesse-michael-han was closed on Feb 9, 2021
2
[Merged by Bors] - refactor(ring_theory/polynomial/symmetric): simplify proof easy lean-gptf ready-to-merge
#6135 by jesse-michael-han was closed on Feb 9, 2021
4
[Merged by Bors] - refactor(analysis/special_functions/trigonometric): simpler proof easy lean-gptf ready-to-merge
#6133 by jesse-michael-han was closed on Feb 9, 2021
2
[Merged by Bors] - refactor(data/set/intervals/basic): simpler proof of Iio_union_Ioo' easy lean-gptf ready-to-merge
#6132 by jesse-michael-han was closed on Feb 10, 2021
4
[Merged by Bors] - refactor(data/fintype/basic): golf card_eq_one_of_forall_eq easy lean-gptf ready-to-merge
#6131 by jesse-michael-han was closed on Feb 9, 2021
2


dwrensha / tryAtEachStep

1. origin story
2. **big picture**
3. implementation
4. results
5. future work



tryAtEachStep

my_tactic

```
theorem sum3 (v : Fin 3 → ℝ) :
    ∑ i, v i = v 0 + v 1 + v 2 := by
  rw [Fin.sum_univ_succ, Fin.sum_univ_succ, Fin.sum_univ_one]
  simp only [Fin.isValue, Fin.succ_zero_eq_one, Fin.succ_one_eq_two]
  ring
```

```
Try this: exact Fin.sum_univ_three v
```

exact?

```
theorem Float.Zero.valid : ValidFinite emin 0 :=
  ⟨by
    rw [add_sub_assoc]
    apply le_add_of_nonneg_right
    apply sub_nonneg_of_le
    apply Int.ofNat_le_ofNat_of_le
    exact C.precPos,
    suffices prec ≤ 2 * emax by
      rw [← Int.ofNat_le] at this
      rw [← sub_nonneg] at *
      simp only [emin, emax] at *
      ring_nf
      rw [mul_comm]
      assumption
    le_trans C.precMax (Nat.le_mul_of_pos_left _ Nat.zero_lt_two),
    by (rw [max_eq_right]; simp [sub_eq_add_neg, Int.ofNat_zero_le])⟩
```

omega

1. origin story
2. big picture
3. **implementation**
4. results
5. future work

leanprover / **lean3**

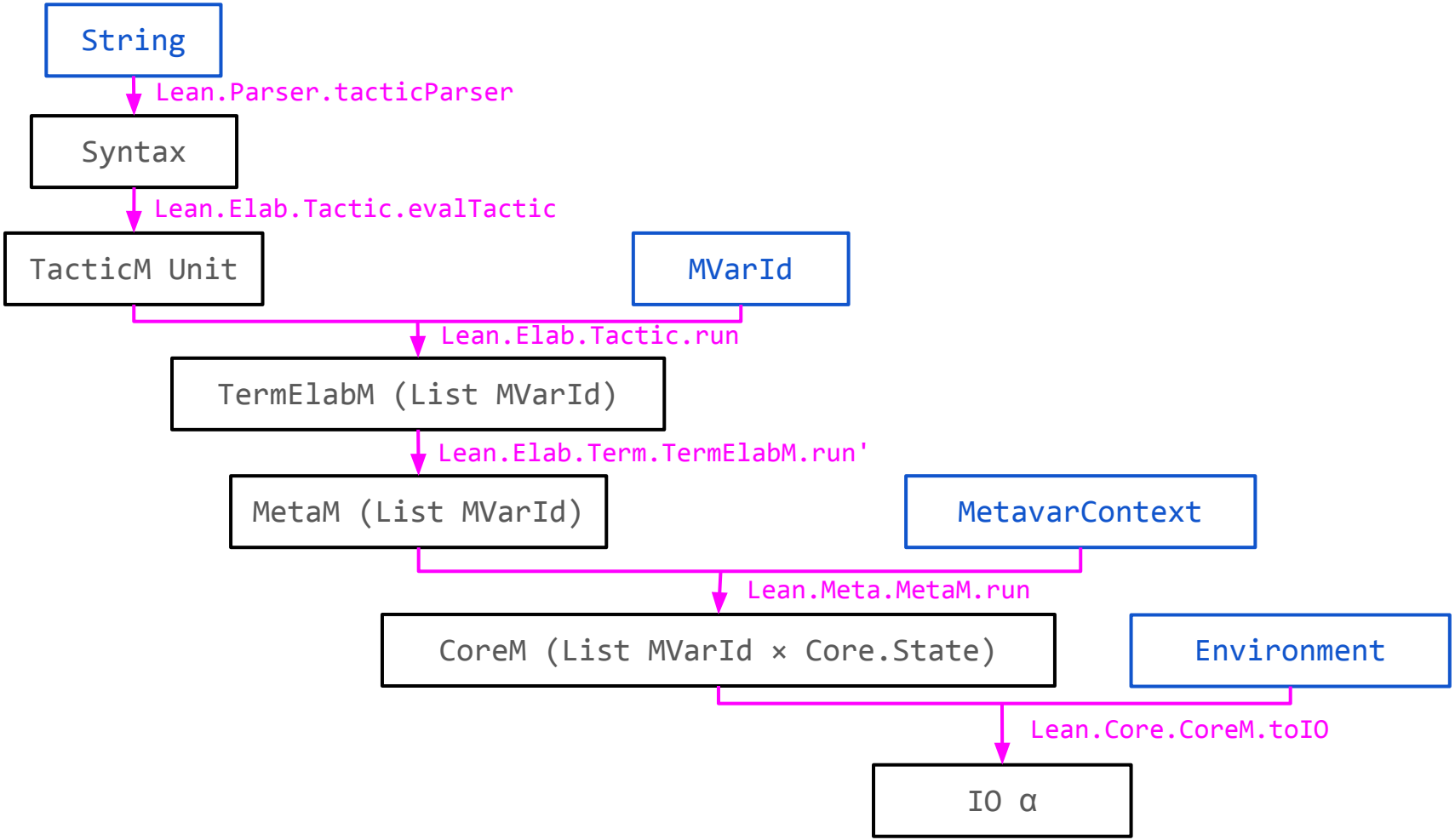
This repository has been archived by the owner on Oct 14, 2023. It is now read-only.

leanprover-community / **mathlib3**

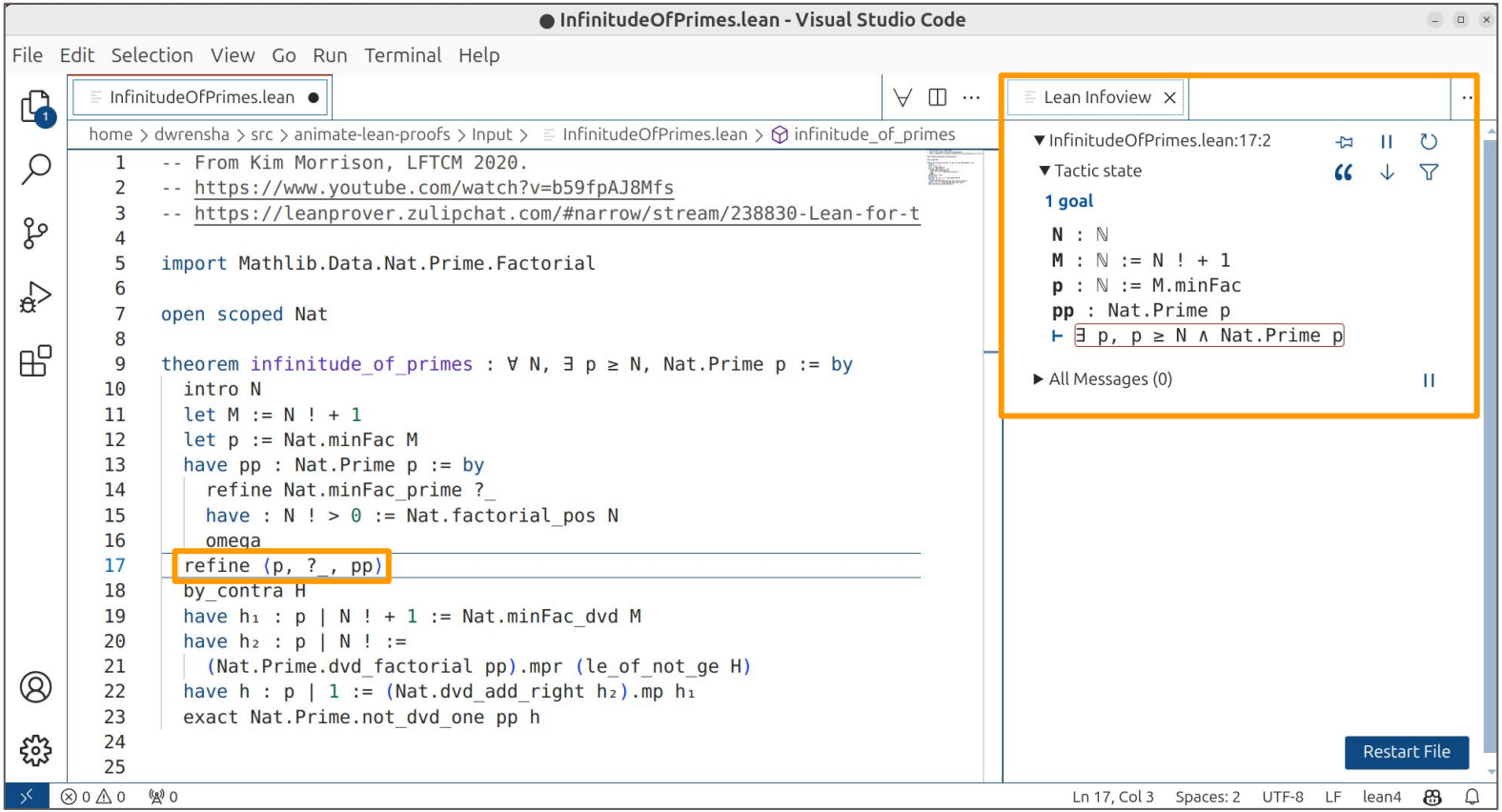
This repository has been archived by the owner on Jul 24, 2024. It is now read-only.

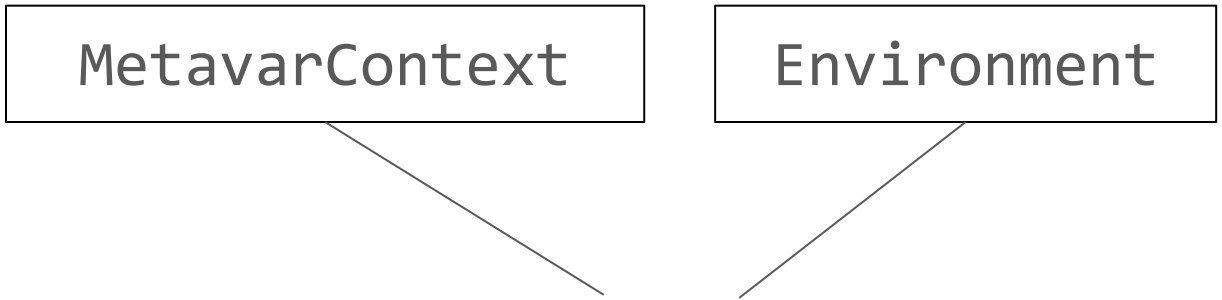
Metaprogramming in Lean 4

Arthur Paulino, Damiano Testa, Edward Ayers, Evgenia Karunus, Henrik Böving, Jannis Limperg, Siddhartha Gadgil, Siddharth Bhat


String
Lean.Parser.tacticParser
Syntax
Lean.Elab.Tactic.evalTactic
TacticM Unit
MVarId
Lean.Elab.Tactic.run
TermElabM (List MVarId)
Lean.Elab.Term.TermElabM.run'
MetaM (List MVarId)
MetavarContext
Lean.Meta.MetaM.run
CoreM (List MVarId × Core.State)
Environment
Lean.Core.CoreM.toIO
IO α

```
partial def elabCommand (stx : Syntax) : CommandElabM Unit := do
  withLogging <| withRef stx <| withIncRecDepth <| withFreshMacroScope do
    match stx with
    | Syntax.node _ k args =>
      if k == nullKind then
        -- list of commands => elaborate in order
        -- The parser will only ever return a single command at a time, but syntax quotations can return multiple ones
        -- Incrementality is currently limited to the common case where the sequence is the direct
        -- output of a macro, see below.
        withoutCommandIncrementality true do
          args.forM elabCommand
      else withTraceNode `Elab.command (fun _ => return stx) (tag :=
        -- special case: show actual declaration kind for `declaration` commands
        (if stx.isOfKind ``Parser.Command.declaration then stx[1] else stx).getKind.toString) do
        let s ← get
        match (← liftMacroM <| expandMacroImpl? s.env stx) with
        | some (decl, stxNew?) =>
          withInfoTreeContext (mkInfoTree := mkInfoTree decl stx) do
            let stxNew ← liftMacroM <| liftExcept stxNew?
            withMacroExpansion stx stxNew do
              -- Support incrementality; see also Note [Incremental Macros]
              if let some snap := (←read).snap? then
                -- Unpack nested commands; see `MacroExpandedSnapshot.next`
                let cmds := if stxNew.isOfKind nullKind then stxNew.getArgs else #[stxNew]
                let nextMacroScope := (← get).nextMacroScope
                let hasTraces := (← getTraceState).traces.size > 0
                let oldSnap? := do
                  let oldSnap ← snap.old?
                  let oldSnap ← oldSnap.val.get.toTyped? MacroExpandedSnapshot
                  guard <| oldSnap.macroDecl == decl && oldSnap.newNextMacroScope == nextMacroScope
                  -- check absence of traces; see Note [Incremental Macros]
                  guard <| !oldSnap.hasTraces && !hasTraces
                  return oldSnap
                let oldCmds? := oldSnap?.map fun old =>
                  if old.newStx.isOfKind nullKind then old.newStx.getArgs else #[old.newStx]
                Language.withAlwaysResolvedPromises cmds.size fun cmdPromises => do
                  snap.new.resolve <| .ofTyped {
                    diagnostics := .empty
                    macroDecl := decl
                    newStx := stxNew
                    newNextMacroScope := nextMacroScope
                    hasTraces
                    next := cmdPromises.zipWith cmds fun cmdPromise cmd =>
                      { range? := cmd.getRange?, task := cmdPromise.result }
```


● InfinitudeOfPrimes.lean - Visual Studio Code
File Edit Selection View Go Run Terminal Help
InfinitudeOfPrimes.lean ●
home > dwrensha > src > animate-lean-proofs > Input > InfinitudeOfPrimes.lean > infinitude_of_primes
1 -- From Kim Morrison, LFTCM 2020.
2 -- https://www.youtube.com/watch?v=b59fpAJ8Mfs
3 -- https://leanprover.zulipchat.com/#narrow/stream/238830-Lean-for-t
4
5 import Mathlib.Data.Nat.Prime.Factorial
6
7 open scoped Nat
8
9 theorem infinitude_of_primes : ∀ N, ∃ p ≥ N, Nat.Prime p := by
10 intro N
11 let M := N ! + 1
12 let p := Nat.minFac M
13 have pp : Nat.Prime p := by
14 refine Nat.minFac_prime ?_
15 have : N ! > 0 := Nat.factorial_pos N
16 omega
17 refine ⟨p, ?_, pp⟩
18 by_contra H
19 have h₁ : p ∣ N ! + 1 := Nat.minFac_dvd M
20 have h₂ : p ∣ N ! :=
21 (Nat.Prime.dvd_factorial pp).mpr (le_of_not_ge H)
22 have h : p ∣ 1 := (Nat.dvd_add_right h₂).mp h₁
23 exact Nat.Prime.not_dvd_one pp h
24
25
Lean Infoview
InfinitudeOfPrimes.lean:17:2
Tactic state
1 goal
N : ℕ
M : ℕ := N ! + 1
p : ℕ := M.minFac
pp : Nat.Prime p
⊢ ∃ p, p ≥ N ∧ Nat.Prime p
All Messages (0)
Restart File
Ln 17, Col 3 Spaces: 2 UTF-8 LF lean4



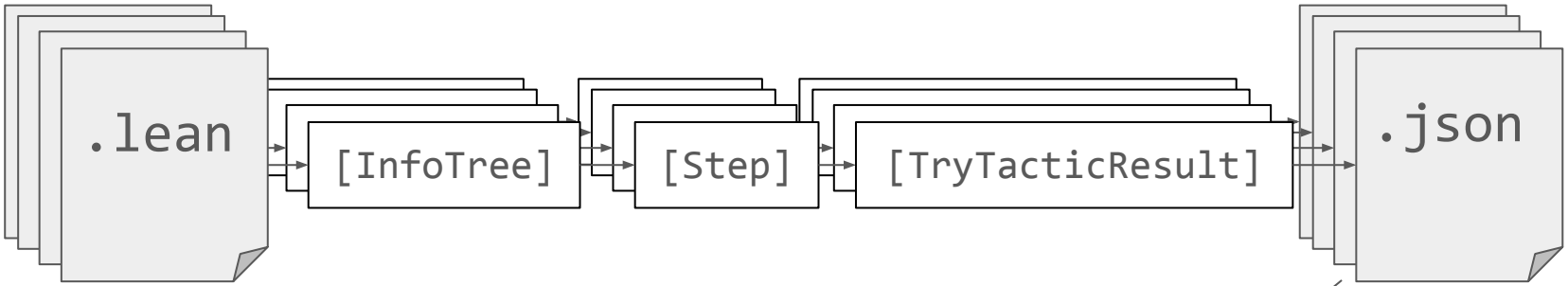

```
Meta.ppGoal : MVarId → MetaM Std.Format
```

InfoTree

```
/-- The InfoTree is a structure that is generated during elaboration and used
    by the language server to look up information about objects at particular points
    in the Lean document.
-/
inductive InfoTree where
  /-- The context object is created at appropriate points during elaboration -/
  | context (i : PartialContextInfo) (t : InfoTree)

  /-- The children contain information for nested term elaboration and tactic evaluation -/
  | node (i : Info) (children : PersistentArray InfoTree)

  /-- The elaborator creates holes (aka metavariables) for tactics and postponed terms -/
  | hole (mvarId : MVarId)
  deriving Inhabited
```

```
structure ElabInfo where
  elaborator : Name
  stx : Syntax
  deriving Inhabited

structure TacticInfo extends ElabInfo where
  mctxBefore  : MetavarContext
  goalsBefore : List MVarId
  mctxAfter   : MetavarContext
  goalsAfter  : List MVarId
  deriving Inhabited

structure CommandContextInfo where
  env           : Environment
  fileMap       : FileMap
  mctx          : MetavarContext := {}
  options       : Options        := {}
  currNamespace : Name           := Name.anonymous
  openDecls     : List OpenDecl  := []
  ngen          : NameGenerator
```

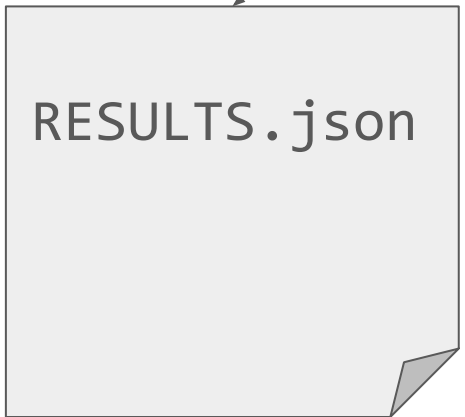
tryAtEachInSnapshotDirectory



Lean.Elab.Frontend

```
structure
  ci: Cont
  ti: Tact
structure
  env: Env
  stx: Syn
  seqStx
  focused
```

```
structure TryTacticResult where
  filepath : String
  startLine : Nat
  startCol : Nat
  oldText : String
  newText : String
  oldToEndOfBranch : String := ""
  parentName : String
  goalIsProp : Bool
  oldProof : String
  newProof : String
  fewerSteps: Bool
  message : Option String
deriving Lean.ToJson
```



`Batteries.Tactic.Lint.Linter ("environment linter")`

- `simpNF`
- `unusedHavesSuffices`
- `docBlame`

`Lean.Elab.Command.Linter ("syntax linter")`

- `docPrime`
- `multiGoal`
- `oldObtain`

1. origin story
2. big picture
3. implementation
4. **results**
5. future work

```
$ time lake exe tryAtEachStepInDirectory "with_reducible exact?" \
                .lake/packages/mathlib/Mathlib -j 31
```

```
real    296m33.206s
user    8945m59.740s
sys     87m3.059s
```

2811 results!

mathlib4#13334

14 Mathlib/MeasureTheory/Function/AEEqOfIntegral.lean

Viewed

```
@@ -265,19 +265,7 @@ theorem ae_nonneg_of_forall_setIntegral_nonneg_of_stronglyMeasurable (hfm : Stro
    intro b hb_neg
    let s := {x | f x ≤ b}
    have hs : MeasurableSet s := hfm.measurableSet_le stronglyMeasurable_const
-   have mus : μ s < ∞ := by
-     let c : ℝ≥0 := ⟨|b|, abs_nonneg _⟩
-     have c_pos : (c : ℝ≥0∞) ≠ 0 := by simpa [c, ← NNReal.coe_eq_zero] using hb_neg.ne
-     calc
-       μ s ≤ μ {x | (c : ℝ≥0∞) ≤ ‖f x‖₊} := by
-         apply measure_mono
-         intro x hx
-         simp only [s, Set.mem_setOf_eq] at hx
-         simpa only [c, nnnorm, abs_of_neg hb_neg, abs_of_neg (hx.trans_lt hb_neg), Real.norm_eq_abs,
-           Subtype.mk_le_mk, neg_le_neg_iff, Set.mem_setOf_eq, ENNReal.coe_le_coe, NNReal] using hx
-       _ ≤ (∫⁻ x, ‖f x‖₊ ∂μ) / c :=
-         (meas_ge_le_lintegral_div hfm.aemeasurable.ennnorm c_pos ENNReal.coe_ne_top)
-       _ < ∞ := ENNReal.div_lt_top (ne_of_lt hf.2) c_pos
+   have mus : μ s < ∞ := Integrable.measure_le_lt_top hf hb_neg
    have h_int_gt : (∫ x in s, f x ∂μ) ≤ b * (μ s).toReal := by
      have h_const_le : (∫ x in s, f x ∂μ) ≤ ∫ _ in s, b ∂μ := by
        refine
```

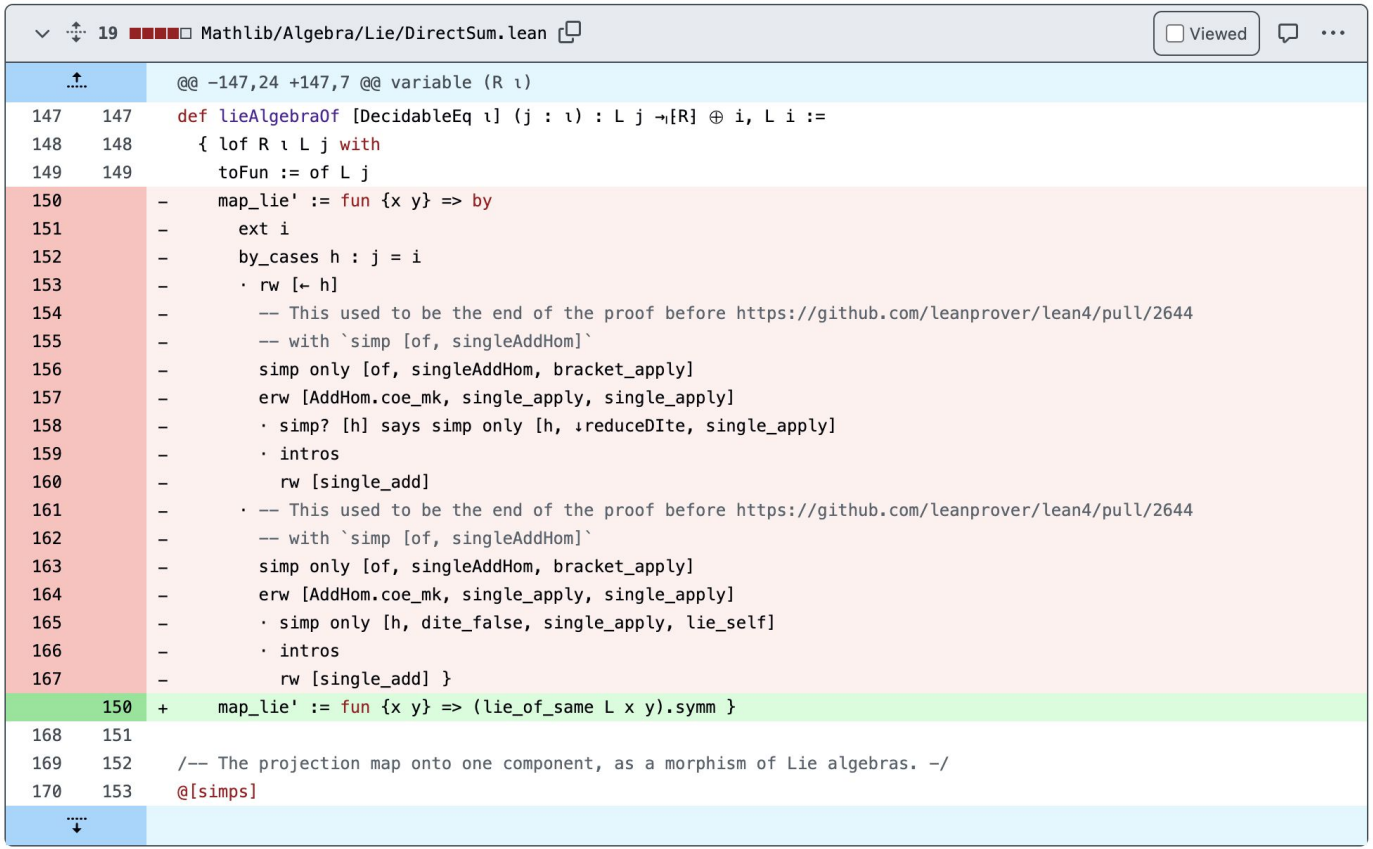
mathlib4#19899

13 Mathlib/Topology/PartitionOfUnity.lean

Viewed

```
@@ -665,15 +665,4 @@ theorem exists_continuous_sum_one_of_isOpen_isCompact [T2Space X] [LocallyCompac
665 665             exact hj)]
666 666       rfl
667 667     intro i x
668     -   refine ⟨f.nonneg i x, ?_⟩
669     -   by_cases h0 : f i x = 0
670     -   · rw [h0]
671     -     exact zero_le_one
672     -   rw [← Finset.sum_singleton (f ·  x) i]
673     -   apply le_trans _ (f.sum_le_one' x)
674     -   rw [finsum_eq_sum (f.toFun ·  x) (by exact toFinite (support (f.toFun · x)))]
675     -   simp only [Finite.toFinset_setOf, ne_eq]
676     -   gcongr with z hz
677     -   · exact fun j _ _ => f.nonneg j x
678     -   simp only [Finset.singleton_subset_iff, Finset.mem_filter, Finset.mem_univ, true_and]
679     -   exact h0
    668 +   exact ⟨f.nonneg i x, PartitionOfUnity.le_one f i x⟩
```

mathlib4#20592



lean4#6352

dwrensha commented on Dec 9, 2024 — Contributor

When `exact?` succeeds, I expect it to give me something that closes the current goal.

However, in the following example, the `exact?` tactic suggests `Try this: refine fun a a_1 => ?_`.

```
def Subgroup' (G : Type) [Mul G] : Type := sorry

instance (G : Type) [Mul G] : Membership G (Subgroup' G) := sorry

def iSup' {G : Type} [Mul G] {ι : Type} (S : ι → Subgroup' G) : Subgroup' G := sorry

theorem mem_iSup_of_mem' {G : Type} [Mul G] {ι : Type} {S : ι → Subgroup' G} (i : ι) :
  ∀ {x : G}, x ∈ S i → x ∈ iSup' S := sorry

theorem foo {G : Type}
    [Mul G] {ι : Type} (S : ι → Subgroup' G)
    {C : (x : G) → x ∈ iSup' S → Prop}
    (hp : ∀ (i : ι) (x : G) (hx : x ∈ S i), C x (mem_iSup_of_mem' i hx))
    (x y : G) :
    (fun x ↦ ∃ (h : x ∈ iSup' S), C x h) x →
      (fun x ↦ ∃ (h : x ∈ iSup' S ), C x h) y →
        (fun x ↦ ∃ (h : x ∈ iSup' S), C x h) (x * y) := by
  exact?
-- Try this: refine fun a a_1 => ?_
-- (kernel) declaration has metavariables 'foo'
```

This occurs on current nightly and on `leanprover/lean4:v4.15.0-rc1`

```
$ time lake exe tryAtEachStepInDirectory omega \
          .lake/packages/mathlib/Mathlib -j 31 \
          --filter-by-fewer-steps false
```

```
real    21m39.891s
user    591m5.542s
sys     63m48.041s
```

[Merged by Bors] - refactor: optimize proofs with omega #11093

Closed **dwrensha** wants to merge 94 commits into `master` from `tryAtEachStep-omega`

Conversation 10 | Commits 94 | Checks 27 | Files changed 93

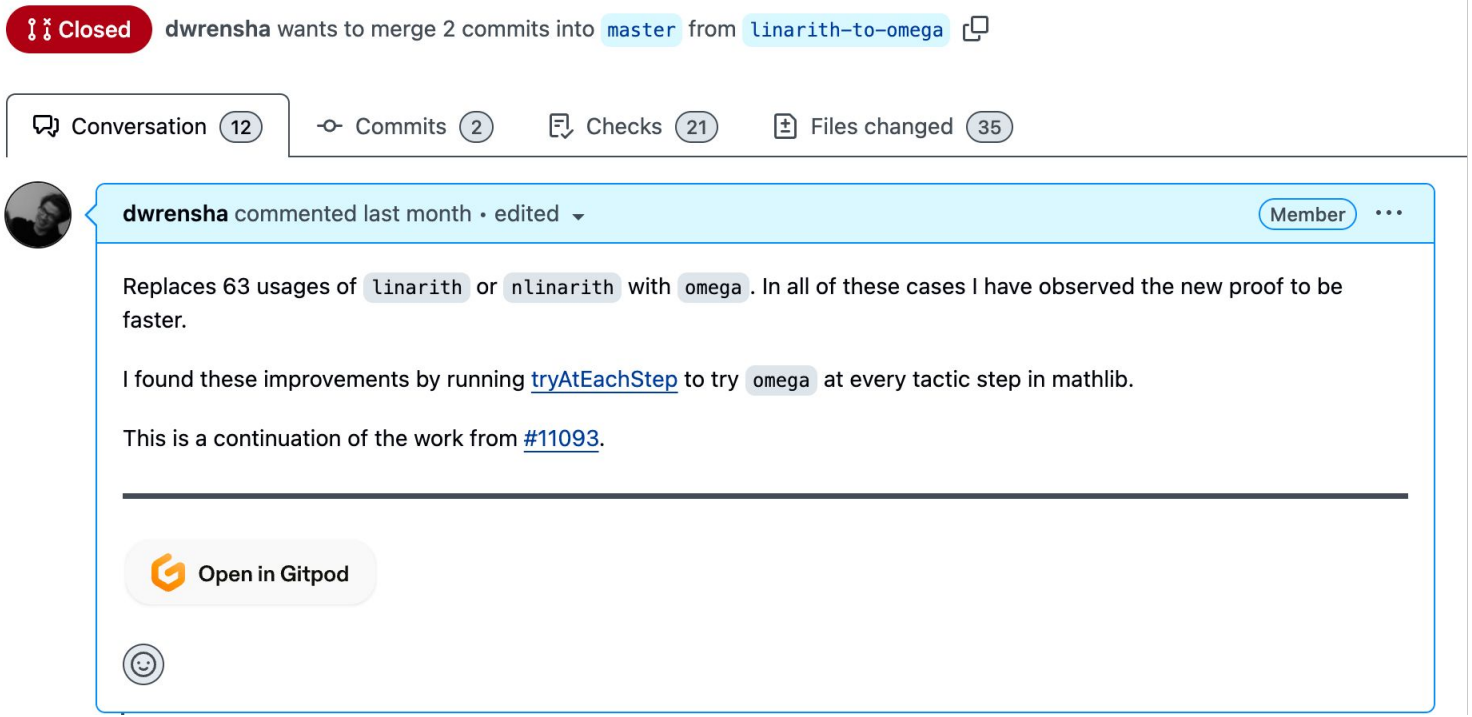
dwrensha commented on Mar 1, 2024 • edited — Member

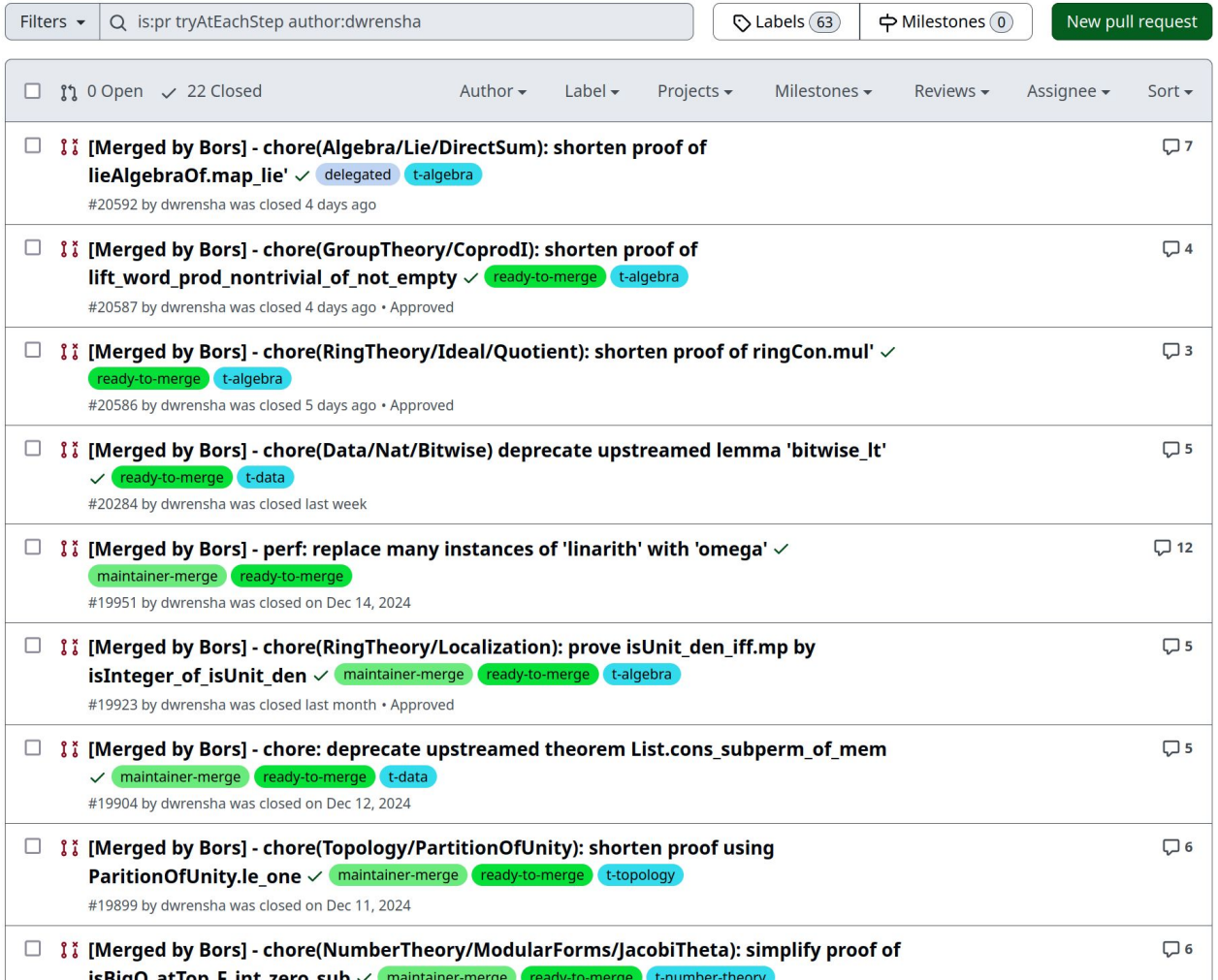
I ran tryAtEachStep on all files under `Mathlib` to find all locations where `omega` succeeds. For each that was a `linarith` without an `only`, I tried replacing it with `omega`, and I verified that elaboration time got smaller. (In almost all cases, there was a noticeable speedup.) I also replaced some slow `aesop`s along the way.

Open in Gitpod

3

[Merged by Bors] - perf: replace many instances of 'linarith' with 'omega' #19951




Filters
is:pr tryAtEachStep author:dwrensha
Labels 63
Milestones 0
New pull request
0 Open
22 Closed
Author
Label
Projects
Milestones
Reviews
Assignee
Sort
[Merged by Bors] - chore(Algebra/Lie/DirectSum): shorten proof of lieAlgebraOf.map_lie'
delegated
t-algebra
#20592 by dwrensha was closed 4 days ago
7
[Merged by Bors] - chore(GroupTheory/CoprodI): shorten proof of lift_word_prod_nontrivial_of_not_empty
ready-to-merge
t-algebra
#20587 by dwrensha was closed 4 days ago • Approved
4
[Merged by Bors] - chore(RingTheory/Ideal/Quotient): shorten proof of ringCon.mul'
ready-to-merge
t-algebra
#20586 by dwrensha was closed 5 days ago • Approved
3
[Merged by Bors] - chore(Data/Nat/Bitwise) deprecate upstreamed lemma 'bitwise_lt'
ready-to-merge
t-data
#20284 by dwrensha was closed last week
5
[Merged by Bors] - perf: replace many instances of 'linarith' with 'omega'
maintainer-merge
ready-to-merge
#19951 by dwrensha was closed on Dec 14, 2024
12
[Merged by Bors] - chore(RingTheory/Localization): prove isUnit_den_iff.mp by isInteger_of_isUnit_den
maintainer-merge
ready-to-merge
t-algebra
#19923 by dwrensha was closed last month • Approved
5
[Merged by Bors] - chore: deprecate upstreamed theorem List.cons_subperm_of_mem
maintainer-merge
ready-to-merge
t-data
#19904 by dwrensha was closed on Dec 12, 2024
5
[Merged by Bors] - chore(Topology/PartitionOfUnity): shorten proof using ParitionOfUnity.le_one
maintainer-merge
ready-to-merge
t-topology
#19899 by dwrensha was closed on Dec 11, 2024
6
[Merged by Bors] - chore(NumberTheory/ModularForms/JacobiTheta): simplify proof of
6

1. origin story
2. big picture
3. implementation
4. results
5. **future work**

```
{"startLine": 739,
 "startCol": 2,
 "proofTermLengthReduction": 786,
 "parentName": "MeasureTheory.Lp.simpleFunc.instAddLeftMono",
 "oldText": "refine (fun f g₁ g₂ hg₁₂ => ?_)",
 "newText": "with_reducible exact?",
 "message":
 "Try this: exact addLeftMono_of_addLeftStrictMono ↥(simpleFunc G p μ)",
 "lengthReduction": 383,
 "goalIsProp": true,
 "filepath":
 "/home/dwrensha/src/compfiles/.lake/packages/mathlib/Mathlib/MeasureTheory/Functio
 "fewerSteps": true},
{"startLine": 895,
 "startCol": 8,
 "proofTermLengthReduction": 1178,
 "parentName": "Monoid.CoprodI.lift_word_prod_nontrivial_of_not_empty",
 "oldText": "obtain (h, hn1, -) := Cardinal.three_le hcard 1 1",
 "newText": "with_reducible exact?",
 "message":
 "Try this: exact lift_word_prod_nontrivial_of_other_i f X hXnonempty hXdisj hpp w
 "lengthReduction": 378,
 "goalIsProp": true,
 "filepath":
 "/home/dwrensha/src/compfiles/.lake/packages/mathlib/Mathlib/GroupTheory/CoprodI.l
 "fewerSteps": true},
{"startLine": 209,
 "startCol": 4,
 "proofTermLengthReduction": 0,
 "parentName":
 "CategoryTheory.Limits.preservesFiniteLimits_of_preservesEqualizers_and_finiteProd
 "oldText": "intro J sJ fJ",
 "newText": "with_reducible exact?",
 "message":
 "Try this: exact fun J [SmallCategory J] [FinCategory J] => preservesLimit_of_pres
 "lengthReduction": 370,
 "goalIsProp": true,
 "filepath":
 "/home/dwrensha/src/compfiles/.lake/packages/mathlib/Mathlib/CategoryTheory/Limits
 "fewerSteps": true},
{"startLine": 41,
 "startCol": 2,
 "proofTermLengthReduction": 282,
 "parentName":
 "CategoryTheory.Projective.projective_iff_preservesEpimorphisms_preadditiveCoyoned
 "oldText": "rw [projective_iff_preservesEpimorphisms_coyoneda_obj]",
 "newText": "with_reducible exact?",
 "message":
 "Try this: exact projective_iff_preservesEpimorphisms_preadditiveCoyoneda_obj P",
```

```
{"startLine": 337,
 "startCol": 4,
 "proofTermLengthReduction": 3666,
 "parentName": "List.getLast_filter'",
 "oldText": "rw [List.getLast_cons_cons] at h' ⊢",
 "newText": "with_reducible exact?",
 "message":
 "Try this: exact getLast_filter_of_pos (fun x => h (Eq.symm x ▸ rfl)) h'",
 "lengthReduction": 348,
 "goalIsProp": true,
 "filepath":
 "/home/dwrensha/src/compfiles/.lake/packages/mathlib/Mathlib/Data/List/Basic.lean"
 "fewerSteps": true},
{"startLine": 58,
 "startCol": 2,
 "proofTermLengthReduction": 643,
 "parentName": "MeasureTheory.Measure.withDensity_rnDeriv_eq",
 "oldText": "suffices μ.singularPart ν Set.univ = 0 by simpa using this",
 "newText": "with_reducible exact?",
 "message": "Try this: exact (singularPart_eq_zero μ ν).mpr h",
 "lengthReduction": 331,
 "goalIsProp": true,
 "filepath":
 "/home/dwrensha/src/compfiles/.lake/packages/mathlib/Mathlib/MeasureTheory/Decompo
 "fewerSteps": true},
{"startLine": 112,
 "startCol": 2,
 "proofTermLengthReduction": 986,
 "parentName": "prime_pow_succ_dvd_mul",
 "oldText": "intro hy",
 "newText": "with_reducible exact?",
 "message":
 "Try this: exact fun a => Prime.pow_dvd_of_dvd_mul_right h (i + 1) a hxy",
 "lengthReduction": 328,
 "goalIsProp": true,
 "filepath":
 "/home/dwrensha/src/compfiles/.lake/packages/mathlib/Mathlib/Algebra/Prime/Lemmas.
 "fewerSteps": true},
{"startLine": 77,
 "startCol": 2,
 "proofTermLengthReduction": 615,
 "parentName": "Finite.to_isCoatomic",
 "oldText": "refine IsCoatomic.mk fun b => or_iff_not_imp_left.2 fun ht => ?_",
 "newText": "with_reducible exact?",
 "message": "Try this: exact IsStronglyCoatomic.toIsCoatomic α",
 "lengthReduction": 325,
 "goalIsProp": true,
 "filepath":
 "/home/dwrensha/src/compfiles/.lake/packages/mathlib/Mathlib/Order/Atoms/Finite.le
```

```
set_option trace.profiler true
```

```
$ lake exe tryAtEachStepInDirectory "search_proof" \
          --imports LeanCopilot Compfiles -j 1
```

```
.libc++abi: terminating due to uncaught exception of type lean::exception: Could not find
native implementation of external declaration 'LeanCopilot.FFI.isGeneratorInitialized'
(symbols 'l_LeanCopilot_FFI_isGeneratorInitialized___boxed' or
'l_LeanCopilot_FFI_isGeneratorInitialized').
```

Compfiles: Catalog Of Math Problems Formalized In Lean

About | **IMO** | USAMO | All

Since 1959, the International Mathematical Olympiad has included a total of **392** problems.

140 problems have been formalized (35.71%).

79 problems have complete formalized solutions (20.15%).

2024	P1	P2	P3	P4	P5	P6
2023	P1	P2	P3	P4	P5	P6
2022	P1	P2	P3	P4	P5	P6
2021	P1	P2	P3	P4	P5	P6
2020	P1	P2	P3	P4	P5	P6
2019	P1	P2	P3	P4	P5	P6
2018	P1	P2	P3	P4	P5	P6
2017	P1	P2	P3	P4	P5	P6
2016	P1	P2	P3	P4	P5	P6
2015	P1	P2	P3	P4	P5	P6
2014	P1	P2	P3	P4	P5	P6
2013	P1	P2	P3	P4	P5	P6
2012	P1	P2	P3	P4	P5	P6
2011	P1	P2	P3	P4	P5	P6
2010	P1	P2	P3	P4	P5	P6
2009	P1	P2	P3	P4	P5	P6
2008	P1	P2	P3	P4	P5	P6

Thank you!