

Numina Fuse is here

Project Numina is excited to release the beta version of [Numina Fuse](#). Fuse is a web app that connects to your Lean projects on GitHub integrating an updated version of our [Numina-Lean-Agent](#), allowing users to leverage the agent in the browser to analyze and update their Lean projects.

Once access to a Lean repo has been granted to the Numina Fuse app, a user can see the repo in their Fuse portal and make new branches on the repo through the app.

The screenshot shows the Numina Fuse web interface for a repository named 'moser' by BoltonBailey/moser. The interface includes a header with the Numina Fuse logo, 'Guide', 'Feedback', and a GitHub icon. A '+ New Blueprint' button is in the top right. The main content is divided into two sections: 'BLUEPRINTS' and 'COMPLETED'. The 'BLUEPRINTS' section lists five items with their names, merge status, update time, and icons for refresh and delete. The 'COMPLETED' section lists three items with their names, merge status, and merge time. The third item in 'COMPLETED' has a detailed description of changes made.

moser			
BoltonBailey/moser			
BLUEPRINTS			
refactor wormreplacement		Updated today	
Computing the polygon of allowable additions	Merged	Updated 1 day ago	
generalize discretization	Merged	Updated 1 day ago	
Fixing sorries	Merged	Updated 1 day ago	
Lower Bounds for Moser's Worm Problem	Merged	Updated 5 days ago	
COMPLETED			
#5 Lower Bounds for Moser's Worm Problem Merged Merged 5 days ago			
#6 Numina/fixing sorries Merged Merged 1 day ago			
#7 Fix Polygon.lean: open Fin.NatCast, deorphan docstring, weaken iff Merged - Add 'open Fin.NatCast' so '(k : Fin n)' coerces (the instance is scoped in Lean 4.30 to avoid coercion loops). - Move 'cyclicCCWChain_implies_isCCWPolygon's docstring to attach to the actual lemma — it was stranded above the 'cyclicCCWChain_gap_ccw' helper, causing a parse error. - Replace 'ConvexPolygon.ofList_eq_none_iff' with the one-direction 'ofList_eq_none_of_length_lt_three'. The forward direction depends on 'convexHullRationalPoints_convex', which transitively relies on the 'sorry' in 'cyclicCCWChain_gap_ccw'. Merged 1 day ago			
#8 generalize discretization Merged Merged 1 day ago			
#9 Computing the polygon of allowable additions Merged Merged 1 day ago			

© 2026 Project Numina · Privacy · Terms

These branches are associated with TeX blueprints that the user can upload or write directly. A user can annotate definitions and theorems in these TeX files with LeanBlueprint-style commands like `\uses` or `\leanok` which the agent will read to get information about what tasks are in process in the repository.

 Numina Fuse / moser / refactor wormreplacement

GuideFeedback

Overview

Blueprint

Source

</> Lean

+ New chat

History

Settings

What should we do next?

Tell me what's next...

Auto

↑

refactor-wormreplacement.tex

```
1 % Blueprint: refactor wormreplacement
2 % Created by Numina
3
4 \begin{document}
5
6 \begin{definition}
7   \label{def:worm-replacement}
8   \lean{wormReplacement}
9   The worm replacement method returns a collection of
10  sets that unionHull the worm with the given set
11 \end{definition}
12
13 \end{document}
14
```

DEFINITIONFORMALIZED

def:worm-replacement

The worm replacement method
returns a collection of sets that
unionHull the worm with the given set

The user can then interact with the agent through a chat interface which lets them ask the agent to carry out tasks or make other changes to the project. For example, you can ask the agent to identify and explain the remaining **sorrys** in a particular part of the project that they are working on.

Numina Fuse

VCV-io / Merkle-tree-soundness-chainInLog

Overview

Blueprint

Source

Lean

New chat

History

Settings

Please summarize the sorries we still have in VCVio/CryptoFoundations/MerkleTree/Inductive/Extractability.lean

Read Extractability.lean

Grep sorry in MerkleTree/Inductive/Extractability.lean

Grep ^\s=sor... in MerkleTree/Inductive/Extractability.lean

Grep \bsorry... in MerkleTree/Inductive/Extractability.lean

Only one literal sorry axiom remains. Let me confirm by looking at the helper theorems mentioned in the docstrings.

Read Extractability.lean lines 95-124

Read Extractability.lean lines 1300-1349

Read Extractability.lean lines 470-549

Grep getPut... in MerkleTree/Inductive/Extractability.lean

Read Extractability.lean lines 560-609

Only one live sorry remains in the file.

extractability_game_coll_caseB_le_inv_ca

rd at

Tell me what's next...

Auto

VCVio/CryptoFoundations/MerkleTree/Inductive/Extractability.lean

1 /-

2 Copyright (c) 2024 ArkLib Contributors. All rights reserved.

3 Released under Apache 2.0 license as described in the file LICENSE.

4 Authors: Quang Dao

5 -/

6

7 import VCVio.CryptoFoundations.MerkleTree.Inductive.Collision

8 import VCVio.CryptoFoundations.MerkleTree.Inductive.Lemmas

9 import ToMathLib.Data.IndexedBinaryTree.Lemmas

10

11 /-!

12 # Inductive Merkle Tree Extractability

13

14 -/

15

16 open scoped NNReal

17

18 section ToVCV

19

20 /-

21 Convenience corollary to 'probEvent_mono' for when the implication holds e

22 not just on the support of the distribution.

23 -/

24 lemma probEvent_mono''.(u, v) {m : Type u → Type v} [Monad m] {α : Type u}

25 {mx : m α} {p q : α → Prop}

26 (h : ∀ x, p x → q x) : Pr[p | mx] ≤ Pr[q | mx] := by

27 apply probEvent_mono

28 tauto

29

30

31 /-

32 For any computation 'oa' and predicate 'p', the probability of 'p' holding

33 equals the probability of 'p * Prod.fst' holding on the output of 'oa.with

34 This follows from the fact that 'withQueryLog' only appends a log without

Files

Infoview

VCVio / CryptoFoundations / MerkleTree

Extractability ▲ 1 +403 -35

Collision

Completeness

Defs

Lemmas

If you want the agent to spend a significant amount of time working on one task, you might want to check the “Auto mode” box. This will let the agent work for extended periods of time, even if you leave the computer. In this example, we have a codebase that was pretty heavily commented from some previous work using Claude, so we can make sure the comments were focused on the main issues and then direct the agent to make an attempt.

Numina Fuse

VCV-io / Merkle-tree-soundness-chainInLog

Overview

Blueprint

Source

Lean

New chat

History

Settings

is 'sorry' 'd is the actual probability bound:

Pr[verified = true ∧ extractedTree.get_idx

= none | game.withQueryLog] ≤ 1/|α|.

The docstring (lines 1319-1336) sketches the

proof strategy: locate the level $k \in \{1, \dots,$

$\text{idx.depth}\}$ where the verifier's hash chain

leaves the committingAdv's tree, then apply a

probOutput_query-style bound on the

verifier's terminal hash query, using $4 *$

$s.\text{leafCount} + 1 \leq q_b$ to rule out adversaries

that pre-set the random oracle on inputs the

verifier later queries.

The other four matches for the word sorry in

the file are docstring text only - they describe

an earlier state of the development. The

helpers they refer to

(support_implies_chainInLog,

getPutativeRoot_support_chain at line 564,

extractability_game_noColl_caseA_eq_zero

at line 1229) are all fully proved now. Those

stale docstrings are waiting cleaning up if you

Great, can you proceed to fill the sorry,

generating additional lemmas as needed?

Auto

VCVio/CryptoFoundations/MerkleTree/Inductive/Extractability.lean

1 /-

2 Copyright (c) 2024 ArkLib Contributors. All rights reserved.

3 Released under Apache 2.0 license as described in the file LICENSE.

4 Authors: Quang Dao

5 -/

6

7 import VCVio.CryptoFoundations.MerkleTree.Inductive.Collision

8 import VCVio.CryptoFoundations.MerkleTree.Inductive.Lemmas

9 import ToMathLib.Data.IndexedBinaryTree.Lemmas

10

11 /-!

12 # Inductive Merkle Tree Extractability

13

14 -/

15

16 open scoped NNReal

17

18 section ToVCV

19

20 /-

21 Convenience corollary to 'probEvent_mono' for when the implication holds e

22 not just on the support of the distribution.

23 -/

24 lemma probEvent_mono''.(u, v) {m : Type u → Type v} [Monad m] {α : Type u}

25 {mx : m α} {p q : α → Prop}

26 (h : ∀ x, p x → q x) : Pr[p | mx] ≤ Pr[q | mx] := by

27 apply probEvent_mono

28 tauto

29

30

31 /-

32 For any computation 'oa' and predicate 'p', the probability of 'p' holding

33 equals the probability of 'p * Prod.fst' holding on the output of 'oa.with

34 This follows from the fact that 'withQueryLog' only appends a log without

Files

Infoview

VCVio / CryptoFoundations / MerkleTree

Extractability ▲ 1 +403 -35

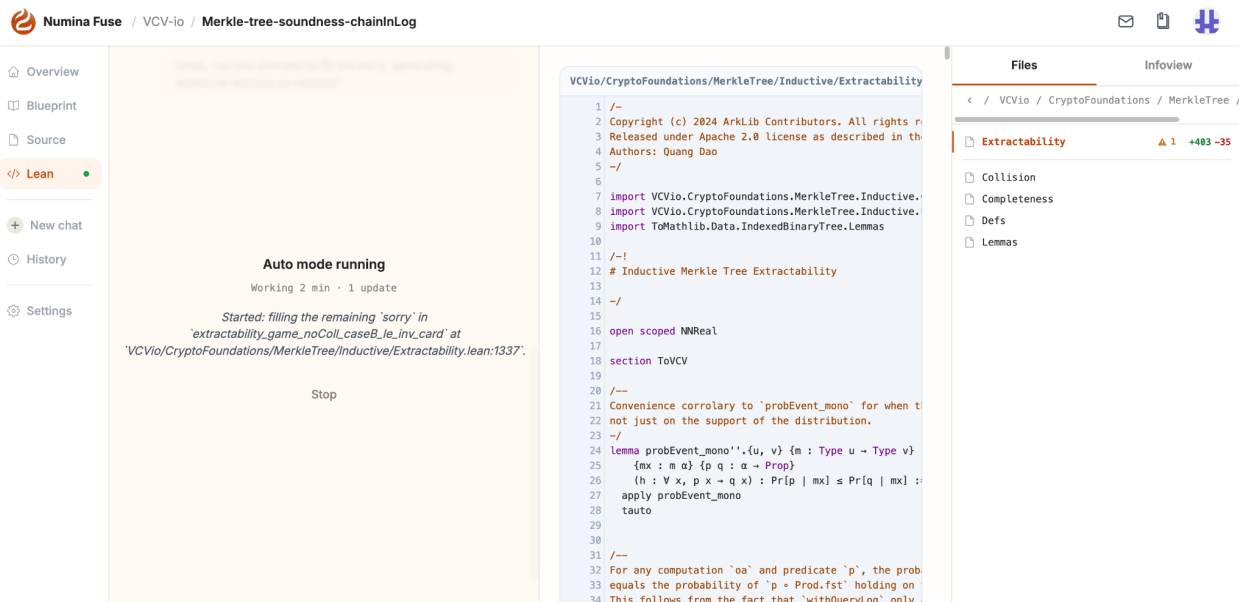
Collision

Completeness

Defs

Lemmas

Auto mode will give you occasional updates as it runs, and you can stop it at any time. This mode is especially effective if you have a plan for a **sorry** that breaks into several lemmas with multiple components that Fuse can spin up subagents for.



We have been using the app over the past few weeks to update a few projects: One example is a project on [Moser's worm problem](#), a longstanding open problem in convex geometry. Progress on this problem is one of many optimization problems that a [recently created site](#) from Terence Tao [tracks](#), and other lower bound results for it suggest it might be amenable to automation. Formalization seems like a good fit to me for this project in particular, because it involves lots of careful manipulations of representations of convex polygons and it will be valuable to at some point to rigorously check that these manipulations are correct.

We have also used the app to help with some work on the [VCV-io](#) cryptography library. That project has recently taken on some content from the downstream ArkLib project, including some work formalizing [Merkle trees](#) there. The app has been useful for porting parts of an unfinished PR upstream and fleshing them out so that they could be merged more easily. Since then, we've continued using the app to work on outstanding issues with the main PR contribution.

For tasks like these, you'd probably reach for an in-editor AI coding agent like Claude Code or Codex, but Fuse offers many benefits:

1. Parallel subagents. Spin them up for multi-part formalization tasks.
2. Long-running auto mode. Kick off work, close the laptop, come back to a diff.
3. Sandboxed cloud execution. Agents touch a server-side clone of your repo, not your machine.

We recommend using the app most in situations where you have a worked-out informal plan for your Lean development, and you want to get a baseline AI implementation of the plan without too much mental overhead.

*Fuse was led by Justin Asher, with Ruichen Qiu and Yichuan Cao as the main developers.
Thanks also to Hugues de Saxcé and Junqi Liu for their contributions along the way.*