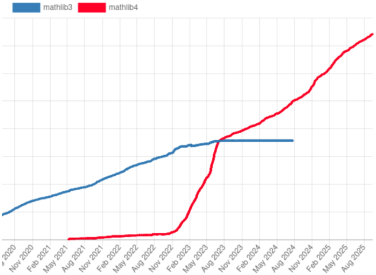
Growing mathlib: review and triage tooling for a large formalised mathematics library

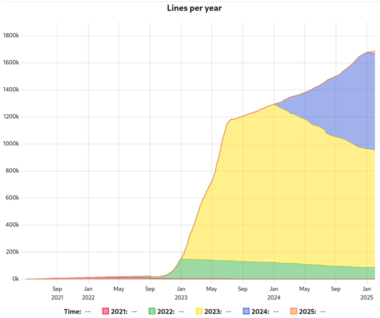
Michael B. Rothgang (he/him)

Formalised mathematics group
Universität Bonn

EuroProofNet Workshop on Proof Libraries
September 15, 2025

What is this all about?





Left: Number of files in mathlib and mathlib4 over time

Right: Lines in mathlib4 per year (without `#aligns`)

Up-to-date graphs: see

`https://leanprover-community.github.io/mathlib_stats.html` resp.

`https://plugh.de/tmp/mathlib4_years_adjusted.html`

What does mathlib's growth *really* mean?

Challenges for users: lots of churn
(e.g. renamed lemmas, import changes, breaking changes to proofs)

What does mathlib's growth *really* mean?

Challenges for users: lots of churn
(e.g. renamed lemmas, import changes, breaking changes to proofs)

Challenges for maintainers (more later):

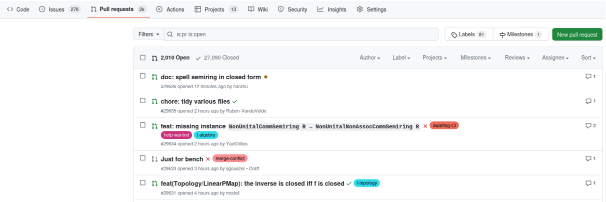
- churn also applies
- keeping technical debt and performance in check
- catch systemic issues: linters
- reviewing contributions

What does mathlib's growth *really* mean?

Challenges for users: lots of churn
(e.g. renamed lemmas, import changes, breaking changes to proofs)

Challenges for maintainers (more later):

- churn also applies
- keeping technical debt and performance in check
- catch systemic issues: linters
- reviewing contributions



is:pr is:open

☐ 2,010 Open ✓ 27,090 Closed

☐ doc: spell semiring in closed form ●
#29636 opened 12 minutes ago by harahu — 1

☐ chore: tidy various files ✓
#29635 opened 2 hours ago by Ruben-VandeVelde — 1

☐ feat: missing instance `NonUnitalCommSemiring R → NonUnitalNonAssocCommSemiring R` ✗ awaiting-CI help-wanted t-algebra
#29634 opened 2 hours ago by YaelDillies — 2

☐ Just for bench ✗ merge-conflict
#29633 opened 3 hours ago by sgouezel • Draft — 1

☐ feat(Topology/LinearPMap): the inverse is closed iff f is closed ✓ t-topology
#29631 opened 4 hours ago by mcdoll — 1

Today's topic: what can mathlib tooling do for you?

- as a user: dealing with churn
- as a contributor: linting to catch common mistakes
- as a reviewer
- as a maintainer

Before we begin

This is joint work with many people, including

Johan Commelin, Damiano Testa and Bryan Gin-ge Chen (left to right)

Tools for mathlib users: dealing with churn

- deprecation warnings for deleted/renamed lemmas
- files moved or split: `deprecated_module` warns about deleted/renamed files
- imports changing: add `import Mathlib` and use `#min_imports`
- Can be automated: prototypes (e.g. `lake exe update_deprecations` by Damiano Testa)

Tools for mathlib users: finding lemmas

- strict naming convention: `https://leanprover-community.github.io/contribute/naming.html`
- lemma-finding tactics: `exact?`, `apply? [using h]`, `rw??`
- loogle (Joachim Breitner): structured theorem search
- moogle, leansearch: natural language search
- generated documentation: rendered in the browser, clickable

Tools for Lean users: proof automation

- dependent types make some automation harder
- for example: `aesop`, `omega`, `fun_prop`, `bv_decide`, `grind` (very new)

Tooling for mathlib contributors: overlooked aspects

- open development, live collaboration (github/zulip)
- welcoming community
- newcomer-friendly: great docs, focus on good tools
- code review: quality control and **teaching venue**
- continuous integration; not rocket science rule (Graydon Hoare)

Scaling mathlib: tools for contributors

- mathlib cache (future: also for projects depending on mathlib)
- `add_deprecations` script (Damiano Testa):
  deprecate renamed lemmas automatically
- `shake` (Mario Carneiro): find superfluous imports
- `lean4checker` (Kim Morrison):
  ensure no meta-programming “tampers” with the environment

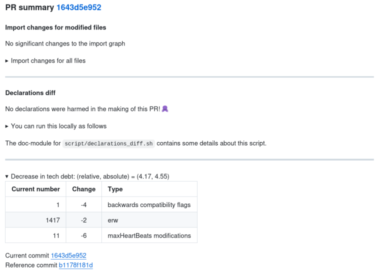
Scaling mathlib: linters

- code style checking, formatting
- file naming conventions
- function naming conventions
- robustness and consistency (e.g. non-terminal simps; simp lemmas in normal form)
- deprecations

Scaling mathlib: some technical challenges

- fast core system (see FRO talk)
- keeping up with core changes
- speeding up mathlib, for example
  - local profiling (easy)
  - benchmarking and performance tracking
  - refactor `FunLike` hierarchy (Anne Baanen, $\approx 20\%$ speed-up)
  - unbundling typeclasses (Yuyang Zhao, in progress)
- keeping technical debt in check
- import refactoring and reduction
  parallelism; less recompilation; easier minimisation

Tooling for reviewers



PR summary 1643d5e952

Import changes for modified files

No significant changes to the import graph

▸ Import changes for all files

Declarations diff

No declarations were harmed in the making of this PR!

▸ You can run this locally as follows

The doc-module for `script/declarations_diff.sh` contains some details about this script.

▾ Decrease in tech debt: (relative, absolute) = (4.17, 4.55)

Current number	Change	Type
1	-4	backwards compatibility flags
1417	-2	erw
11	-6	maxHeartBeats modifications

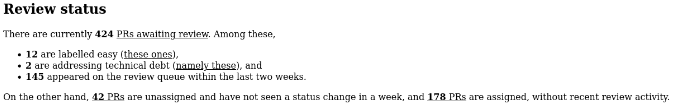
Current commit 1643d5e952
Reference commit b1178f181d

- sticky summary comment (Damiano Testa):
  show renamings, import changes, technical debt change
- emoji-bot: signal on zulip if a PR has already been reviewed/merged
- finding good reviewers: auto-labelling and automatic assignment

Automatic reviewer assignment

- collect each reviewer's areas of interest/expertise
- assign candidate reviewers with the best expertise (subject to opt-out and capacity)
- random assignment, taking capacity into account
- can be overriden; "stealing review" is welcome!

Automatic reviewer assignment: reflections



Review status

There are currently **424** PRs awaiting review. Among these,

- **12** are labelled easy (these ones),
- **2** are addressing technical debt (namely these), and
- **145** appeared on the review queue within the last two weeks.

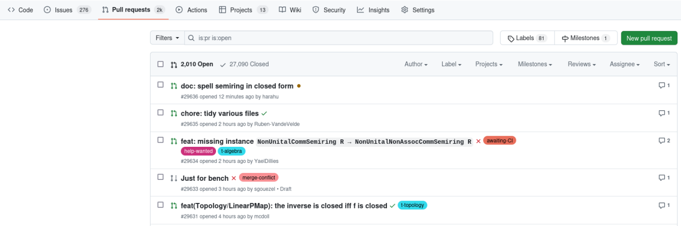
On the other hand, **42** PRs are unassigned and have not seen a status change in a week, and **178** PRs are assigned, without recent review activity.

- Assignment avoids diffusion of responsibility
- Uncovers areas with unclear labelling or missing reviewers
- Stale assigned PRs: need manual triage

Scaling reviews of PRs

- everybody can review, don't be shy!
  guidelines: `https://leanprover-community.github.io/contribute/pr-review.html`
- *maintainers* have final merge rights: currently 26
- *reviewers* group: experienced members (currently 54)
  PR approval notifies maintainers directly
- helps share the load: 1/3 PRs gets reviewer-approved first
  4986 merged PRs $>$ 15000; 1659 got maintainer-merged, 28 twice or more
- make it enjoyable: regular review/triage meeting; reviewing retreats

Editorial tooling for mathlib (Commelin-R.)



□ 2,010 Open ✓ 27,090 Closed

□ doc: spell semiring in closed form ● 1
#29636 opened 12 minutes ago by harahu

□ chore: tidy various files ✓ 1
#29635 opened 2 hours ago by Ruben-VandeVelde

□ feat: missing instance `NonUnitalCommSemiring R → NonUnitalNonAssocCommSemiring R` ✕ awaiting-CI help-wanted t-algebra 2
#29634 opened 2 hours ago by YaelDillies

□ Just for bench ✕ merge-conflict 1
#29633 opened 3 hours ago by sgouezel • Draft

□ feat(Topology/LinearPMap): the inverse is closed iff f is closed ✓ t-topology 1
#29631 opened 4 hours ago by mcdoll

Mathlib has a review bottleneck; need

- more reviewer bandwith
- discoverability: are there PRs I can review?
- assignment of responsibility — one designated reviewer per PR
- triage and tracking: make sure no PR is left behind

Mathlib needs editorial tooling

Editorial tooling for mathlib (Commelin-R.)

Overall goal

Keep track of all open pull requests, and ensure each PR gets a timely response.

Editorial tooling for mathlib (Commelin-R.)

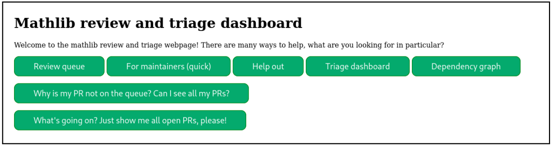
Overall goal

Keep track of all open pull requests, and ensure each PR gets a timely response.

Specific aims

- track the review queue, filtered e.g. by subject area
- “leave no PR behind”
- automatically assign a suitable reviewer
- better “last updated” information; total time in review
- different target groups need different information

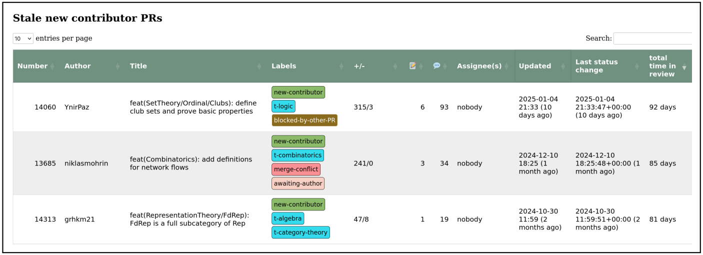
A review and triage dashboard for mathlib



Mathlib review and triage dashboard

Welcome to the mathlib review and triage webpage! There are many ways to help, what are you looking for in particular?

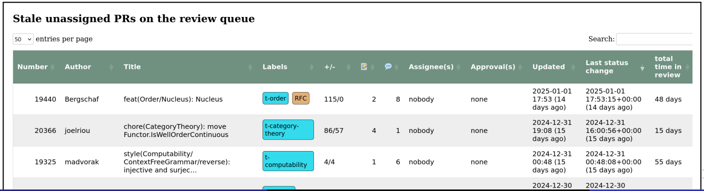
Review queue
For maintainers (quick)
Help out
Triage dashboard
Dependency graph
Why is my PR not on the queue? Can I see all my PRs?
What's going on? Just show me all open PRs, please!



Stale new contributor PRs

10 entries per page Search:

Number	Author	Title	Labels	+/-	📝	💬	Assignee(s)	Updated	Last status change	total time in review
14060	YnirPaz	feat(SetTheory/Ordinal/Clubs): define club sets and prove basic properties	new-contributor t-logic blocked-by-other-PR	315/3	6	93	nobody	2025-01-04 21:33 (10 days ago)	2025-01-04 21:33:47+00:00 (10 days ago)	92 days
13685	niklasmohrin	feat(Combinatorics): add definitions for network flows	new-contributor t-combinatorics merge-conflict awaiting-author	241/0	3	34	nobody	2024-12-10 18:25 (1 month ago)	2024-12-10 18:25:48+00:00 (1 month ago)	85 days
14313	grhkm21	feat(RepresentationTheory/FdRep): FdRep is a full subcategory of Rep	new-contributor t-algebra t-category-theory	47/8	1	19	nobody	2024-10-30 11:59 (2 months ago)	2024-10-30 11:59:51+00:00 (2 months ago)	81 days



Stale unassigned PRs on the review queue

50 entries per page Search:

Number	Author	Title	Labels	+/-	📝	💬	Assignee(s)	Approval(s)	Updated	Last status change	total time in review
19440	Bergschaf	feat(Order/Nucleus): Nucleus	t-order RFC	115/0	2	8	nobody	none	2025-01-01 17:53 (14 days ago)	2025-01-01 17:53:15+00:00 (14 days ago)	48 days
20366	joelriou	chore(CategoryTheory): move Functor.IsWellOrderContinuous	t-category-theory	86/57	4	1	nobody	none	2024-12-31 19:08 (15 days ago)	2024-12-31 16:00:56+00:00 (15 days ago)	15 days
19325	madvorak	style(Computability/ContextFreeGrammar/reverse): injective and surjec...	t-computability	4/4	1	6	nobody	none	2024-12-31 00:48 (15 days ago)	2024-12-31 00:48:08+00:00 (15 days ago)	55 days
									2024-12-30	2024-12-30	

Some technical details

- statically generated page: Python writes HTML (with some CSS/JS)
- backend: download metadata for all github PRs, near-real time
- about 4500 lines of Python code
  and 1000 lines of shell scripts, `.graphql` schemas etc.
- some surprises/lessons learned
  - local testing is good
  - beware of different time-zones
  - github actions: run at most every 5min, and no guarantees
    deal with concurrent runs of same workflow
  - prepare for outliers, e.g. PRs with *huge* metadata
  - pagination
  - did it really update? Github's "last update" is incomplete
  - expect errors: network, intermittent, push races, . . .
  - github's search results are sometimes outdated

Next steps for review and triage

- faster dashboard updates: ideally, latency < 1 minute
- closer integration with zulip
  (e.g. weekly automatic post of stale maintainer merged PR)
- triage team for manual follow-up
- reminder: all tooling requires regular maintenance
- need more reviewers!

Summary

1. It takes a village to create a big library.
2. Empower and mentor new contributors.
3. Build tooling to automate as much as you can.
4. Custom tooling takes effort, but is often be worth it!
5. Maintaining software sustainably needs funding.

Ask not what mathlib tooling can do for you —
ask what you can do for mathlib tooling!

Summary

1. It takes a village to create a big library.
2. Empower and mentor new contributors.
3. Build tooling to automate as much as you can.
4. Custom tooling takes effort, but is often be worth it!
5. Maintaining software sustainably needs funding.

Ask not what mathlib tooling can do for you —
ask what you can do for mathlib tooling!

Thanks for listening! Any questions?

Further reading

Anne Baanen, Matthew Robert Ballard, Bryan Gin-ge Chen, Johan Commelin, Michael Rothgang and Damiano Testa.
Growing Mathlib: maintenance of a large scale mathematical library.
To appear, CICM '25. arXiv: `https://arxiv.org/abs/2508.21593`

Théo Zimmermann. *Challenges in the collaborative evolution of a proof language and its ecosystem*. PhD thesis, Université Paris Cité, 2019.
About Rocq and its ecosystem.

Fabian Huch. *Big Math in Interactive Theorem Provers: Scaling the Isabelle Archive of Formal Proofs*. PhD thesis draft, TU München, 2025.
Focus on the Isabelle ecosystem and AFP, but reviews the others also.