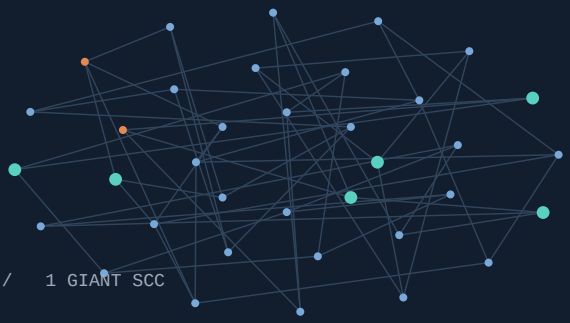
SYSTEMS INTELLIGENCE / TECHNICAL BRIEF 01

The Hidden Architecture of Mathematics

A dependency-graph reading of Mathlib4: how formalized mathematical areas import, reuse, and connect to one another.

DEPENDENCY LANDSCAPE



8,422 FILES / 27 AREAS / 1 GIANT SCC

8,422	27	25 / 27
FILES	**AREAS**	**GIANT SCC**
Repository snapshot count	Subject-level aggregation	Reported area-level strongly connected component

EXECUTIVE ASSESSMENT

The infographic depicts Mathlib as a highly interconnected software-and-mathematics ecosystem. Its strongest structural claim is that 25 of 27 aggregated areas belong to one strongly connected component. This is a statement about transitive file-import reachability—not proof that every field conceptually depends on every other field, nor that mathematical proofs are circular.

Analytical posture. This brief distinguishes (1) what the source infographic explicitly reports, (2) what follows from graph theory, and (3) plausible architectural interpretations. Ambiguous annotations are flagged rather than silently reconciled.

01 / EXECUTIVE SUMMARY

What the graphic says

The source visual summarizes file-level imports in a Mathlib4 repository snapshot identified as commit `3cb72cf`, dated 2026-09-27. It reports 8,422 files classified into 27 mathematical areas and displays, for each ordered pair of areas, the share of files in the row area that transitively import the column area.

25	2	33
AREAS IN SCC	**OUTSIDE SCC**	**ORANGE MARKS**
25 of 27; approximately 92.6%	Condensed; Information Theory	Upward relationships meeting the legend threshold

Three findings deserve attention

01 — A highly integrated core. The source reports that 25 of 27 areas form one strongly connected component at the area level. In graph terms, every area in that component can reach every other through directed paths in the aggregated dependency graph.

02 — Two areas outside the giant SCC. The caption identifies Condensed and Information Theory as the two outside areas and states that they are imported by no other area. Under the importer → imported convention, this means they have no incoming edges from other areas at the area level; it does not mean they have no dependencies of their own.

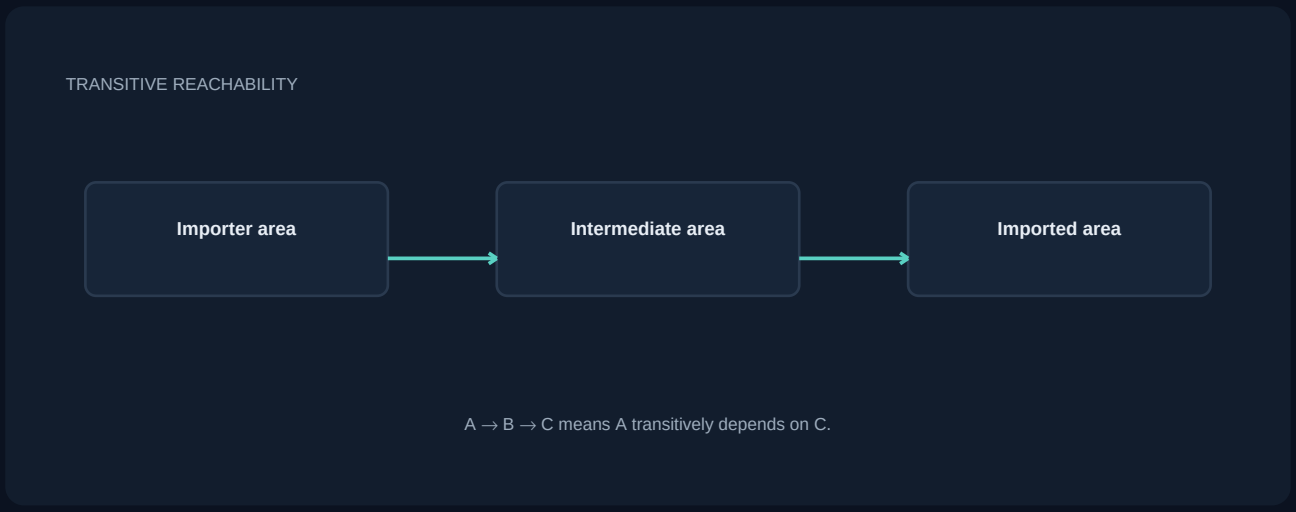
03 — Sparse conceptual-looking links can coexist with close mathematical relationships. The callouts highlight 3% for Category Theory / Linear Algebra and 1% for Analysis / Category Theory. These are file-level import prevalence figures, not measures of conceptual distance or intellectual influence.

> **BOTTOM LINE**
> Treat the matrix as an architectural map of formalized source files. It is useful for identifying coupling, reachability, and candidate dependency hubs. It is not a direct map of mathematical foundations, theorem usage, or the best order in which humans should learn mathematics.

02 / READING THE MATRIX

A directional, weighted dependency matrix

There are 27 row labels and the same 27 column labels. A cell at row X, column Y represents the share of files classified in X that transitively import at least one file classified in Y. The matrix is directional: the X→Y percentage need not equal the Y→X percentage.



Formal definition

Let F_X be the set of files classified in area X, and let D(Y) be the set of files classified in area Y. Let f ■ g mean that a nonempty directed import path exists from file f to file g. Then the displayed proportion can be represented as:

$$M[X,Y] = |\{ f \in F(X) : \exists\, g \in F(Y),\ f \blacksquare g \}| \,/\, |F(X)|$$

Multiply by 100 to express it as a percentage. This formulation captures prevalence across importing files. A file that reaches one Y-file and a file that reaches many Y-files both count once in the numerator.

How to read a cell

MATRIX ELEMENT	INTERPRETATION
Row X	The importing area: files whose dependencies are being counted.
Column Y	The imported area: the destination category being tested.
Bubble size	Share of row-area files that transitively reach at least one file in the column area.
Orange bubble	A highlighted upward relationship under the chart's stated threshold.
Diagonal square	Self-area position; visually distinguished from cross-area comparisons.

The bubble legend gives reference sizes at 5%, 25%, 50%, and 100%. A small dot is still a real measured relationship; it simply means the dependency is present in a smaller share of files in that row area.

> **IMPORTANT**
> This is not a symmetric correlation matrix. It is a directed dependency-prevalence matrix. Do not describe a cell as a 'correlation' or assume the mirrored cell has the same value.

03 / TRANSITIVE IMPORTS

Why indirect dependencies count

A source file can import another file, which imports a third. The first file then transitively depends on the third even if it never names that third file in its own import list.

A → B → C means A reaches C through B. The transitive closure of a directed graph records all such reachable pairs, not just immediate edges.

Direct versus transitive

DIRECT IMPORT GRAPH	TRANSITIVE REACHABILITY
A → B B → C C → D	A reaches B, C, and D. B reaches C and D. C reaches D.

This matters because advanced mathematical files may rely on foundational infrastructure several imports away. A direct-import-only view would miss those inherited dependencies. Conversely, transitive closure can make dependencies look widespread even when the importing file uses only a narrow part of the imported library.

> **INTERPRETIVE LIMIT**
> A transitive import establishes that the imported module is available along the dependency path. It does not establish that every declaration in that module—or even a particular declaration—is used by the importing file.

Graph model

Represent the file system as a directed graph G=(V,E): each file is a vertex; an edge f→g means f imports g. Reachability f■g means there is a directed path from f to g. The infographic aggregates file-level reachability into area-level percentages.

The resulting matrix is a weighted directed graph on 27 area-nodes. Its entries are prevalence weights, not counts of import statements and not counts of distinct theorems used.

04 / THE 27 AREA LABELS

The taxonomy being compared

The source chart lists 27 areas in a fixed order. The following descriptions provide short orientation to the labels. They are general subject descriptions, not a reconstruction of the precise file-classification rules used by the author of the infographic.

AREA	ORIENTATION
Control	A source classification label; its exact scope is not defined in the infographic.
Logic	Formal propositions, inference, quantification, and proof-related infrastructure.
Data	A source classification label; its exact scope is not defined in the infographic.
Order	Order relations, partial orders, lattices, and related structures.
Basic	A source classification label for basic infrastructure; exact scope is not supplied.
Computability	Algorithmic computability, decidability, and formal models of computation.
Algebra	Abstract operations and structures such as groups, rings, fields, and modules.
Set Theory	Sets, membership, subsets, and set-based constructions.
Group Theory	Groups and their structure-preserving maps and actions.
Combinatorics	Counting, finite configurations, discrete structures, and graph-like objects.
Model Theory	Formal languages, structures, interpretations, and logical definability.
Category Theory	Objects, morphisms, composition, and universal constructions.
Linear Algebra	Vector spaces, linear maps, matrices, and related constructions.
Ring Theory	Rings, ideals, modules, and polynomial-ring structures.
Algebraic Topology	Algebraic invariants associated with topological spaces.
Topology	Continuity, open sets, compactness, connectedness, and topological spaces.
Representation Theory	Algebraic structures represented through linear or other transformations.
Information Theory	Information, entropy, coding, and communication-related mathematics.
Field Theory	Fields, field extensions, and related algebraic constructions.
Condensed	The condensed mathematics area.
Dynamics	Evolution of systems, iterations, flows, orbits, and fixed-point concepts.
Analysis	Limits, continuity, differentiation, integration, and function spaces.
Algebraic Geometry	Geometric objects and structures defined or studied algebraically.
Number Theory	Integers, primes, divisibility, congruences, and arithmetic structures.
Geometry	Shapes, spaces, distances, curvature, and geometric structures.
Measure Theory	Measures, measurable spaces, and integration foundations.
Probability	Random variables, distributions, expectation, and probabilistic structures.

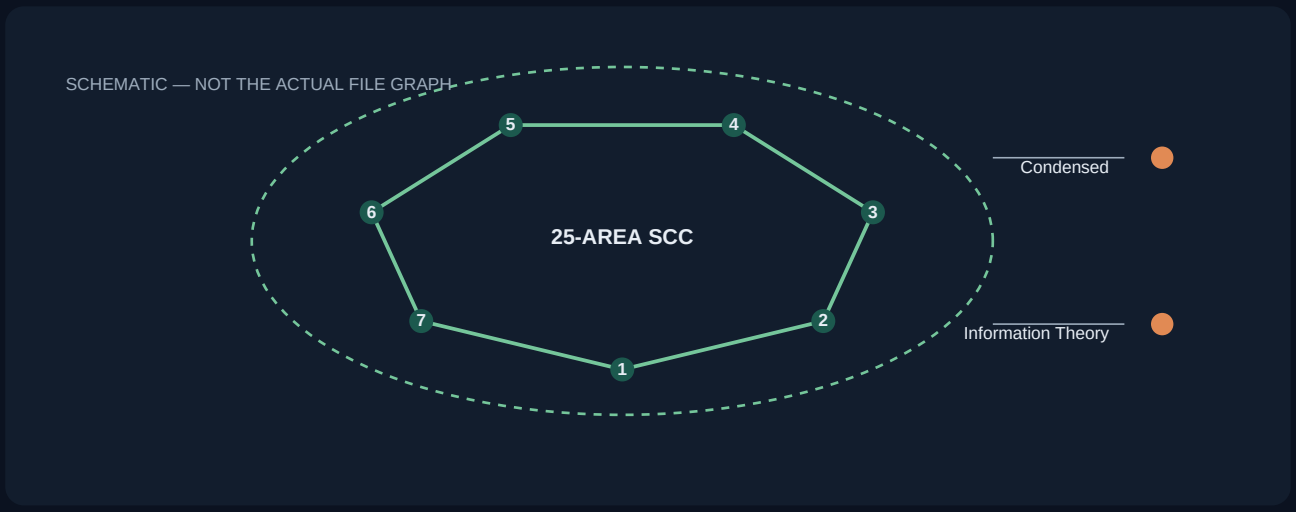
CLASSIFICATION CAUTION

Several areas overlap conceptually—e.g., Algebraic Topology, Algebraic Geometry, and Representation Theory. The chart's categories are a practical aggregation scheme, not mutually exclusive philosophical divisions of mathematics.

05 / THE GIANT STRONGLY CONNECTED COMPONENT

25 of 27 areas: a mutually reachable core

The source caption reports that 25 of the 27 areas form one strongly connected component at the area level. A strongly connected component (SCC) is a maximal set of vertices in which every vertex can reach every other vertex by a directed path.



Formal criterion

For all areas X and Y in the component, X⇝Y and Y⇝X. The paths may pass through other areas; direct edges between every pair are not required.

```
25 / 27 = 0.9259… ≈ 92.6% of the listed areas are in the reported giant SCC.
```

What the two exceptions mean

The infographic names Condensed and Information Theory as the two areas outside the giant SCC and says they are “imported by no other area.” With edges oriented importer → imported dependency, this indicates no incoming area-level edges from other areas. These areas may still import other areas; the caption does not say they are dependency-free or isolated.

> **DO NOT OVERREAD THE SCC**
> A giant SCC is a property of the aggregated area graph. It does not imply that every file imports every other file, that every pair of areas has a direct dependency, or that Lean proofs are logically circular.

Why it matters architecturally

A large SCC indicates that the aggregated dependency structure is not a simple one-way hierarchy. Multiple areas are connected through dependency paths that eventually loop back. This can reflect extensive reuse, shared abstractions, and the historical evolution of a mature library. Whether any particular cycle is desirable or problematic requires inspection of the underlying modules and dependency edges.

06 / THREE CALLOUTS UNDER THE LENS

What the highlighted percentages do—and do not—say

The infographic annotates three relationships. These are source-reported values and should be interpreted as file-level dependency shares, subject to the chart's row/column convention.

Set Theory / Dynamics — 91%

The callout reads “SetTheory → Dynamics: 91%” and mentions `Dynamics.FixedPoints.Defs`, saying it “sits under 50% of mathlib.” However, the chart's general legend says rows are importing areas and columns are imported areas. The arrow shorthand in the callout is therefore potentially ambiguous relative to the matrix convention. The safest source-faithful report is that the annotation names Set Theory and Dynamics with a 91% figure; the direction should be verified against the corresponding cell or underlying data before making a directional claim.

The note about `Dynamics.FixedPoints.Defs` appears to refer to its position in the library's import-depth distribution. The infographic does not define the calculation behind “under 50%,” so it should not be read as “half of Mathlib imports this file.”

Category Theory / Linear Algebra — 3%

The callout describes these areas as “a world apart.” The measured 3% is a low file-level dependency share for the indicated direction, not a measure of conceptual distance. Category Theory can express linear algebra categorically, while a library implementation may encode the relevant mathematics without importing a specialized Category Theory module.

Analysis / Category Theory — 1%

The 1% figure similarly indicates sparse file-level dependency prevalence in the indicated direction. It does not imply that Analysis and Category Theory are mathematically unrelated. Analytic constructions can be formalized using other abstractions and modules without requiring imports from a dedicated Category Theory area.

> **INTERPRETATION RULE**
> Conceptual relationship ≠ source-file import. Historical influence ≠ import. Theorem usage ≠ transitive import. Keep these four kinds of relationship separate.

07 / UPWARD DEPENDENCIES & ARCHITECTURE

The orange marks

The legend identifies 33 highlighted upward relationships, with a threshold of at least 20% and a column 15-or-more levels deeper. The source image does not fully specify how the level ordering is computed, so the exact ranking algorithm cannot be reconstructed from the infographic alone.

The visual intent is to call attention to substantial dependencies that run against the chart's expected ordering. In software architecture, such a pattern can be worth investigating because a foundational layer depending on a much deeper or more specialized layer may create coupling or constrain refactoring.

> **UPWARD DOES NOT AUTOMATICALLY MEAN BAD**
> A highlighted dependency can be a legitimate consequence of the mathematics or of the library's chosen abstractions. It is a diagnostic signal, not a defect label.

A layered mental model versus an ecosystem

A textbook curriculum often presents a progression from logic and basic structures toward algebra, topology, analysis, geometry, and specialized theories. A mature library may be less linear: general abstractions are reused across domains, later developments feed back into shared infrastructure, and source-file boundaries reflect implementation decisions as well as mathematical organization.

This is why the 25-area SCC is significant. It challenges a purely one-way picture of mathematical software architecture, but it does not overturn the role of foundational logic or the formal kernel.

A critical distinction: software cycles are not proof cycles

The import graph is a graph of modules. Lean's kernel checks proof terms according to its formal rules. A cycle in an area-level aggregation does not establish that a proposition is justified by itself. The aggregation may also hide file-level structure; it is not a theorem-dependency graph.

08 / MEASUREMENT & LIMITATIONS

A precise reading of the evidence

The denominator is the number of files in the importing area. The numerator is the number of those files that transitively reach at least one file in the imported area. This makes the percentage normalized within the row area, but it does not weight by the number of imported files or by the number of declarations used.

MEASURED BY THE GRAPH	NOT ESTABLISHED BY THE GRAPH
Area-level file-import reachability	Conceptual dependence between fields
Share of files with a transitive dependency	Share of theorems that use another field's results
Directed asymmetry between area pairs	Historical or intellectual influence
Area-level strongly connected component membership	Logical circularity or foundational priority
Highlighted upward relationships under the legend	Proof that an architecture is defective

Four methodological caveats

1. File-level granularity. A file can import a module while using only a small subset of its declarations. Import reachability can overstate actual semantic use.

2. Aggregation. Each area may contain heterogeneous subfields. An area-level percentage can conceal very different patterns among its constituent modules.

3. Classification. The image does not provide the rules assigning files to the 27 areas. Overlapping mathematical subjects make classification choices consequential.

4. Snapshot dependence. The results describe a particular repository commit. Refactoring, new modules, or changed classifications can alter the matrix and SCC structure.

> **REPRODUCIBILITY REQUIREMENT**
> To independently validate the exact percentages, orange-mark count, SCC membership, and depth annotation, one would need the commit, file-to-area mapping, direct import graph, and the algorithm used to compute transitive closure and ordering levels.

09 / WHAT TO INVESTIGATE NEXT

From descriptive map to research program

The infographic is a useful first-order diagnostic. A deeper study could test whether the observed network reflects healthy reuse, unnecessary coupling, or a mixture of both.

Priority investigations

INVESTIGATION	METHOD	WHAT IT WOULD REVEAL
Direct vs transitive matrix	Compute both matrices from the same file graph.	Whether broad dependencies arise from direct imports or long dependency chains.
Declaration-level usage	Inspect references to imported declarations.	Which available imports are actually used by proofs and definitions.
Area centrality	Compute in/out degree, betweenness, and reachability.	Potential infrastructure hubs and bridge areas.
SCC decomposition	Compute SCCs at file level and area level.	Whether the giant area SCC is supported by many paths or a few bridges.
Temporal evolution	Repeat on selected commits or releases.	How coupling and modularity change over time.
Change-impact analysis	Track dependents of high-centrality modules.	Potential refactoring blast radius and maintenance risk.

Suggested analytical metrics

In-degree (importer → imported orientation). Number of other area-nodes with direct edges into a dependency area; a candidate measure of how many areas rely on it.

Out-degree. Number of direct dependency areas an area imports; a rough measure of its immediate dependency breadth.

Betweenness centrality. How often an area lies on shortest paths between other areas; useful for identifying bridges, with the caveat that dependency graphs may have many alternative paths.

Reachability prevalence. Share of files or areas that can reach a given area; a broader measure of transitive architectural footprint.

Temporal delta. Compare matrices M(t) across commits to detect growing coupling, modularization, or newly emerging cross-area relationships.

10 / ANALYST CONCLUSION

The map beneath the disciplines

The source infographic offers a compact view of the hidden software architecture beneath a substantial body of formalized mathematics. It shows that subject labels do not translate into a clean one-way hierarchy of independent modules. Instead, at the level of transitive file imports, the reported Mathlib4 snapshot contains a large mutually reachable core spanning 25 of 27 areas.

The two areas outside that component—Condensed and Information Theory—are notable not because the image proves they are mathematically isolated, but because the caption reports that no other area imports them. The highlighted low percentages involving Category Theory similarly warn against confusing formal module dependency with conceptual proximity.

The most productive next step is not to rank mathematical subjects by importance. It is to interrogate the dependency graph at finer granularity: identify direct edges, actual declaration usage, file-level SCCs, bridging modules, and changes over time. That would turn a striking descriptive graphic into an actionable study of formalization architecture.

FINAL JUDGMENT

Mathlib's area-level import network, as summarized in the supplied graphic, is best understood as an interconnected formalization ecosystem. The 25/27 SCC is the central structural finding; the rest of the analysis should be read through the distinction between source-file reachability and mathematical meaning.

SOURCE & SCOPE

Primary source: user-supplied infographic titled “How much of math depends on other math?” The source reports file-level imports at Mathlib4 commit `3cb72cf` (2026-09-27), across 8,422 files and 27 areas. This brief interprets the graphic and its annotations; it does not independently recompute the dataset or verify the repository snapshot.

All illustrative diagrams in this brief are explanatory schematics, not reconstructions of the underlying 8,422-file graph. The three callout percentages and the 33 upward-relationship count are transcribed from the source image. The direction of the Set Theory / Dynamics shorthand is flagged as ambiguous against the chart's stated row-column convention.

APPENDIX / SOURCE VISUAL

Original infographic

The original image is reproduced below as the primary visual source for the analysis. The report's diagrams elsewhere are explanatory schematics.

