Munkres' General Topology Autoformalized in Isabelle/HOL

Dustin Bryant[1], Jonathan Julián Huerta y Munive[2], Cezary Kaliszyk[3], and Josef Urban[4]

[1]Independent [2]Aalborg University [3]University of Melbourne [4]AI4REASON and University of Gothenburg / Chalmers University

Goal and result. We report on a large-scale LLM-assisted autoformalization[1] of the general topology chapters of Munkres' *Topology* in Isabelle/HOL. The experiment produced 85,472 lines of Isabelle/HOL code across four theory files, covering all 39 sections of Chapters 2–8 (Sections 12–50), from topological spaces and continuous maps through dimension theory. The current development contains 199 definitions and 806 lemmas, theorems, and corollaries, all fully proved: the final `sorry` count is zero. Landmark results include the Tychonoff theorem, the Stone–Čech compactification, the Nagata–Smirnov and Smirnov metrization theorems, the Baire category theorem, both classical and general versions of Ascoli's theorem, the space-filling curve, the existence of a nowhere-differentiable function, and the final dimension-theory imbedding theorems. The estimated cost (subscription for 24 active days) is about $160.

Suitability as an autoformalization benchmark. The project is a controlled follow-up to a recent Megalodon formalization of the same source material [13]. Munkres is long, standard, proof-rich, and sufficiently diverse to stress many aspects of formalization: elementary set-topological reasoning, products and quotient maps, compactness, normality, metrization, paracompactness, function spaces, Baire-category analysis, and dimension theory. Reusing the same textbook across proof assistants makes it possible to compare not merely isolated theorem proving but the whole human–agent–prover loop. In contrast with the Megalodon experiment, Isabelle/HOL supplies a mature library, Isar's declarative proof language, and strong automation via `sledgehammer`, `blast`, `simp`, `meson`, `metis`, and arithmetic procedures. This richer environment shortened many direct proofs, but also introduced a new failure mode: uncontrolled tactics can trigger large searches in the imported `Complex_Main` context.

Method: sorry first, then hammer in bulk. The central methodological lesson is the "sorry-first" discipline. The agent first writes only the definitions, theorem statements, and a structured Isar skeleton whose proof leaves are `sorry`. The skeleton is compiled, each open leaf is annotated with a bounded `sledgehammer` call, Isabelle/HOL is run once to collect proof suggestions, and the fastest reconstructed proofs are substituted back. Failed leaves are decomposed into smaller `have` blocks and the cycle repeats. This separates neural proof structuring from ATP proof search. The final instruction file eventually forbade the agent from inserting new direct tactics in fresh code–not `auto`, not `simp`, not `blast`, and not even an apparently harmless one-line proof–because premature tactics were the main source of timeouts. Build-time profiling and a split session cache were also essential: caching Chapters 2–4 reduced incremental builds from roughly a minute to about 12–13 seconds.

Agents and scale. Two LLM coding agents were used sequentially. ChatGPT 5.2, through the Codex CLI, produced the initial bulk formalization during March 2–13, 2026, including all definitions and theorem statements and many deferred proofs. Claude Opus 4.6, through Claude Code, then performed systematic proof development during March 21–April 3, reducing 51 remaining `sorry`'s to zero. Both were driven by an automated `tmux`-based loop that reissued the standing instruction when the agent finished or stalled. The Claude session log contains 92,524 JSONL records, including 17,925 shell commands, 5,186 Isabelle/HOL builds, 3,266 bulk theory-processing runs, 7,135 file edits, and 4,962 file reads. Of 764 non-system user messages, 635 were automatic prompts; the remaining human interventions mostly corrected tactic explosion, inefficient tool use, cherry-picking of easy goals, and resource-management mistakes.

Formalization design. The development intentionally mirrors Munkres rather than Isabelle's native type-class topology. A topology is represented by an explicit carrier set and a family of open sets, for example `is_topology_on X T`. This supports multiple topologies on the same

[1] https://github.com/JUrban/isa_top_autoform1

type and keeps statements close to the textbook, but it does not interoperate directly with Isabelle's existing topological-space hierarchy. The choice produced robust, readable source traceability: definitions and theorems are named by Munkres number and annotated with source locations in the original LaTeX. It also created a parallel topology library, with known rough edges. In particular, some definitions are weaker than the reusable mathematical ideal–for example, the topology predicate should defensively require $T \subseteq \mathcal{P}(X)$. The proved theorems remain sound because their assumptions supply the needed carrier constraints, but a future cleanup pass should strengthen definitions and add a bridge to Isabelle's library.

Evaluation and proof engineering observations. The development contains 8,879 uses of `blast`, 4,608 of `simp`, 576 of `auto`, 100 of `metis`, and 59 of `meson`; it also contains 8,160 explicit `unfolding` occurrences, because the agents avoided registering simplification rules. This is non-idiomatic but stable: it prevents accidental simpset explosions. Median direct proof length is about 25 lines, but several landmarks required substantial infrastructure. The Urysohn metrization theorem has a 1,196-line direct proof inherited from the early, less factored phase; the Nagata–Smirnov theorem uses 38 helpers; the space-filling curve required a Hilbert-curve-style recursive construction; and the nowhere-differentiable-function theorem combines the Baire category theorem with uniform approximation in $C([0,1],\mathbb{R})$. Compared with Megalodon, the Isabelle direct proofs of the Urysohn lemma and Tietze extension are an order of magnitude shorter, while section totals show that much work has merely moved into helper lemmas.

Limitations and negative results. The experiment also exposed several current limitations of agentic autoformalization. The agents are weak at defensive definition design, often proving exactly what is needed under strong local assumptions instead of designing robust reusable interfaces. They rarely perform a global compression pass after success; consequently the result contains many wrapper lemmas, repeated unfoldings, and proof scripts that are longer than a human-maintained library would tolerate. They also lack a reliable cost model for Isabelle tactics. A tactic suggested by training data as idiomatic can be disastrous in a large context, while a slightly less fashionable proof method such as `meson` can be much faster for equality-light topology goals. These are not failures of soundness–Isabelle checks every proof–but they are failures of taste, maintainability, and autonomous resource management.

Reusable lessons. We believe the main lessons transfer beyond topology. First, textbook autoformalization in Isabelle is a systems problem: build latency, session caching, proof-search batching, and log analysis matter as much as model quality. Second, declarative proof assistants with proof reconstruction offer a particularly good division of labor: the LLM proposes the sequence of mathematical reductions, and external provers solve the small leaves. Third, rough but complete formalizations can be valuable intermediate artifacts. Once all statements are proved, expert effort can be spent on semantic cleanup and library integration rather than on the open-ended search for missing proofs. This suggests a two-stage future workflow: use agents to produce complete but verbose corpora, then use a combination of humans, proof minimizers, and additional agents to turn them into idiomatic libraries.

Conclusion. The experiment suggests that textbook-scale autoformalization in a mature proof assistant is now practical, cheap relative to manual formalization, and scientifically measurable. The result is not yet an idiomatic reusable topology library: definitions should be strengthened, redundant assumptions removed, and proofs compressed. But it is already a complete, independently checkable formalization of a major topology textbook segment, created under an explicit no-human-code policy. We see the main contribution as the demonstrated workflow: LLMs are effective at proposing declarative proof structure, while hammers and proof reconstruction discharge the leaves. The next step is to turn the observed failure modes into stronger agent rules and to use this corpus as a benchmark for cross-system autoformalization.

References

[1] G. Bancerek et al. The role of the Mizar Mathematical Library for interactive proof development in Mizar. *J. Autom. Reason.* (2017).

[2] J. C. Blanchette, C. Kaliszyk, L. C. Paulson, and J. Urban. Hammering towards QED. *J. Formaliz. Reason.* 9(1), 101–148 (2016).

[3] C. E. Brown, C. Kaliszyk, M. Suda, and J. Urban. Hammering higher order set theory. In *CICM 2025*, LNCS 16136, pp. 3–20. Springer (2025).

[4] A. Hahn and A. De Lon. Understandable autoformalization with Felix. Submitted to ITP 2026.

[5] J. Hölzl, F. Immler, and B. Huffman. Type classes and filters for mathematical analysis in Isabelle/HOL. In *ITP 2013*, LNCS, pp. 279–294. Springer (2013).

[6] A. Q. Jiang et al. Draft, sketch, and prove: guiding formal theorem provers with informal proofs. In *ICLR 2023*. OpenReview.net (2023).

[7] C. Kaliszyk, J. Urban, J. Vyskocil, and H. Geuvers. Developing corpus-based translation methods between informal and formal mathematics. In *CICM 2014*, LNCS 8543, pp. 435–439. Springer (2014).

[8] D. Mulligan and L. C. Paulson. Isabelle/Copilot: an AI-driven assistant for Isabelle/HOL. Isabelle users mailing list (2026).

[9] J. R. Munkres. *Topology*, 2nd edn. Prentice Hall (2000).

[10] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL—A Proof Assistant for Higher-Order Logic*. LNCS 2283. Springer (2002).

[11] L. C. Paulson. Isabelle: the next seven hundred theorem provers. In *CADE 1988*, LNCS, pp. 772–773. Springer (1988).

[12] L. C. Paulson. Porting the HOL Light analysis library: some lessons. In *CPP 2017*, p. 1. ACM (2017).

[13] J. Urban. 130k lines of formal topology in two weeks: simple and cheap autoformalization for everyone? arXiv:2601.03298 (2026).

[14] Q. Wang, C. Kaliszyk, and J. Urban. First experiments with neural translation of informal to formal mathematics. In *CICM 2018*, LNCS 11006, pp. 255–270. Springer (2018).

[15] Y. Wu et al. Autoformalization with large language models. In *NeurIPS 2022* (2022).

A Formalization content by chapter

File	Lines	Defs	Results	Sorry	Status
Ch2 (§§12–22)	20,751	78	189	0	complete
Ch3 (§§23–29)	13,458	28	107	0	complete
Ch4 (§§30–36)	18,908	18	123	0	complete
Ch5–8 (§§37–50)	32,355	75	387	0	complete
Total	85,472	199	806	0	complete

B Landmark theorem proof sizes

Direct counts measure only the theorem's proof body; section counts include local definitions and helper lemmas. This distinction matters because the final workflow deliberately moved proof burden into small helper lemmas.

Theorem	Direct	Section	Helpers	Megalodon direct
Urysohn lemma (33.1)	287	3,126	20	2,964
Urysohn metrization (34.1)	1,196	3,373	19	2,174
Tietze extension (35.1)	324	3,597	18	10,369
Tychonoff (37.3)	198	1,394	6	–
Nagata–Smirnov (40.3)	968	2,455	38	–
Space-filling curve (44.1)	325	1,919	44	–
Ascoli general (47.1)	372	1,912	29	–
Baire category (48.2)	62	2,907	21	–
Nowhere differentiable (49.1)	53	1,296	24	–
Dimension imbedding (50.5)	1,103	6,271	96	–

C Representative theorem statements

Nagata–Smirnov metrization.

```
theorem Theorem_40_3:
shows "top1_metrizable_on X TX <->
  (is_topology_on X TX & top1_regular_on X TX &
   (EX B. top1_sigma_locally_finite_family_on X TX B &
          basis_for X B TX))"
```

Space-filling curve.

```
theorem Theorem_44_1:
shows "EX f::real => (real * real).
  top1_continuous_map_on
    (top1_closed_interval 0 1)
    (top1_closed_interval_topology 0 1)
    ((top1_closed_interval 0 1) * (top1_closed_interval 0 1))
    (product_topology_on
      (top1_closed_interval_topology 0 1)
      (top1_closed_interval_topology 0 1)) f
  & f ‘ (top1_closed_interval 0 1)
    = (top1_closed_interval 0 1) * (top1_closed_interval 0 1)"
```

D Session-log summary

Quantity	Value
Compressed log records	92,524 JSONL records
Assistant messages	50,119
Shell commands	17,925
Isabelle/HOL builds	5,186
Bulk theory-processing runs	3,266
File edits	7,135
File reads	4,962
Non-system user messages	764
Automatic standing prompts	635 (83.1%)
Manual interventions	129

E Known cleanup tasks

The current development is complete but intentionally not polished. The most important cleanup tasks are: strengthen carrier-set conditions in definitions such as `is_topology_on`; add

a reconciliation layer with Isabelle's type-class topology; compress verbose proof scripts after completion; remove redundant assumptions; relocate helper lemmas to their natural sections; and replace repeated explicit unfoldings by carefully scoped automation where this does not reintroduce tactic explosions.

E.1 The Tychonoff Theorem: A Case Study

The Tychonoff theorem (Theorem 37.3: arbitrary products of compact spaces are compact) was the first major result proved in Phase 2 (March 21, 07:52). The proof follows Munkres' approach via the finite intersection property (FIP) and Zorn's lemma, decomposed into six helper lemmas:

```
theorem Theorem_37_3:
assumes hIne: "I ≠ {}"
assumes hComp: "∀i∈I. top1_compact_on (X i) (TX i)"
shows "top1_compact_on (top1_PiE I X)
(top1_product_topology_on I X TX)"
```

Key helpers include `Lemma_37_1` (existence of maximal FIP collections via Zorn's lemma), `Lemma_37_2` (properties of maximal FIP collections), as well as the lemma `tychonoff_subbasis_in_maxFIP` (subbasis elements belong to the maximal FIP extension). The proof involves delicate combinatorics of product topologies, subbases, and Hilbert's ε operator (`SOME`)—the latter being a recurring source of difficulty in Isabelle formalizations, as the chosen witness must be shown to satisfy the required properties across multiple contexts.

E.2 The Nowhere-Differentiable Function: From Baire to Analysis

Theorem 49.1—the existence of a continuous function on $[0, 1]$ that is nowhere differentiable—was one of the most technically demanding results, fully completed on March 25 in a 14-hour, 40-commit session. The proof applies the Baire category theorem to the complete metric space $\mathcal{C}([0, 1], \mathbb{R})$ with the sup metric. The key steps: (i) prove $\mathcal{C}([0, 1])$ is complete (`C01_complete`), hence Baire; (ii) define the difference-quotient sets U_n and prove they are open (`top1_U49_open`) and dense (`U49_dense_approx`, requiring piecewise-linear approximation via "triangle functions"); (iii) assemble via the Baire property. The density proof required uniform continuity (Heine–Cantor), Archimedean arguments for the mesh size, and careful partition-of-unity reasoning for the piecewise approximant—a substantial piece of formal analysis sitting atop the topology.

F Qualitative Analysis of the Definitions and Theorems

Beyond the quantitative metrics reported above, we conducted a manual review of the formalization's definitions, theorem statements, and proofs, checking faithfulness to Munkres, mathematical correctness, and adherence to Isabelle best practices. The review was based on commit `06dd7b2` (March 31), since the human reviewers needed a fixed snapshot to work from while the formalization continued to evolve. The review covered Chapters 2–4 in detail (approximately 53,000 lines) and sampled the remaining chapters. We summarize the findings below, organized by theme.

F.1 Definitional Soundness

The most consequential class of issues concerns definitions that are *logically weaker* than the intended mathematical concept, typically because a carrier-set constraint is missing from the right-hand side.

The topology predicate itself. The formalization defines `is_topology_on X T` without requiring $T \subseteq \mathcal{P}(X)$. This omission is not merely cosmetic: one can formally derive, for instance, that the collection $T := \{\emptyset, \{0\}, \{1\}, \{0,1\}\}$ satisfies `is_topology_on {0} T`, despite containing $\{1\} \not\subseteq \{0\}$. The root cause is Isabelle's convention that $\bigcap \emptyset = \mathit{UNIV}$, which forces the clause $F \neq \{\}$ in the finite-intersection axiom; this in turn means the definition does not prevent T from containing sets that extend beyond X. A corrected definition would add the premise $T \subseteq \mathit{Pow}\ X$, restoring the standard requirement that every open set is a subset of the carrier.

Propagation to derived predicates. The same weakness propagates to several downstream definitions. The predicate `openin_on X T U` is defined as $U \in T$ without requiring $U \subseteq X$, allowing formally "$\{1\}$ is open in the space $\{0\}$." Similarly, `neighborhood_of` does not constrain its argument to lie within X, and the finer-than relation `finer_than T T'` neglects to require that both T and T' are topologies on the same underlying set—making it meaningful to compare arbitrary set families. The product-basis and box-basis definitions carry redundant conditions (e.g., requiring both $U_i \in T_i$ and $U_i \subseteq X_i$, when the latter follows from the former under the assumption that each T_i is a topology on X_i); while not incorrect, this bloat obscures the mathematical content.

Assessment. Crucially, these definitional weaknesses do *not* invalidate the proved theorems: every theorem in the formalization carries explicit `is_topology_on` assumptions that supply the missing constraints at each use site. The proofs are therefore sound, but the definitions are less reusable than they could be—a user inheriting these definitions without the accompanying assumptions could derive spurious results. This pattern is characteristic of LLM-generated code: the model reliably produces definitions that are *sufficient* for the proofs at hand but does not anticipate downstream reuse or defensive design.

F.2 Faithfulness to Munkres

A second category of issues concerns theorem statements that deviate from the textbook in content, not merely in formulation.

Incomplete statements. Several theorems omit mathematically significant clauses present in Munkres. For example, Theorem 19.1 in the formalization lacks the characterization that in the product topology, all but finitely many factors equal the full space X_α—the defining feature that distinguishes the product topology from the box topology. Theorem 26.7 proves compactness of a binary product $X \times Y$ but does not state the finite-product generalization that Munkres gives (and which follows by a straightforward induction). Theorem 15.1 is reduced to the assertion that the product topology is a topology, omitting the stronger statement that products of basis elements form a basis for it.

Mislabeled results. In a few cases, a lemma is labeled with a Munkres number but proves a different (often weaker or tangentially related) statement. Most notably, `Lemma_23_1` in the formalization merely unfolds the definition of connectedness, whereas Munkres' Lemma 23.1 characterizes connectedness of a subspace Y in terms of sets whose closures do not meet the complementary part of Y—a genuinely different result. Similarly, Lemma 23.2 omits the disjointness clause ($Y \cap D = \emptyset$ or $Y \cap C = \emptyset$) that gives the result its geometric content.

Missing content. The formalization omits the alternate characterization of connectedness via clopen sets (a standard equivalence stated informally on the same page as Munkres' definition), sequential compactness and Theorem 28.2, the section on nets, several counterexamples demonstrating that limit-point compactness does not imply compactness, and the notion of "distance from a point to a set." These gaps are understandable given the project's pace but affect the formalization's value as a standalone reference.

F.3 Proof Quality

The proofs exhibit a characteristic uniformity imposed by the sorry-first methodology. While this discipline is effective at scale, it introduces several systematic shortcomings.

Missed simplifications. Many lemmas that serve as "wrappers" around previously proved results retain unnecessarily long proofs when a one-line combination would suffice. For example, `Lemma_21_2`—which simply conjoins its two constituent halves `Lemma_21_2_sequence` and `Lemma_21_2_sequence_converse`—could be discharged by two `metis` calls rather than a structured Isar proof. Several proofs throughout can be replaced by short `by (simp add: ...)` or `by (smt ...)` invocations. The sorry-first workflow, which delegates every step to `sledgehammer` in isolation, does not attempt global proof compression after the fact.

Tactic-level inefficiencies. Isolated instances of the deprecated `apply` / `using` idiom persist (e.g., `apply (rule bexI[...])` followed by `using ... by blast`), where a single combined `by (rule ...)` call would be both shorter and more idiomatic. Duplicate rewrite-rule warnings (`simp add:` supplying a rule already in the simpset) appear throughout, suggesting that `sledgehammer`'s suggestions were accepted without filtering. These are minor blemishes individually but accumulate across 85,000 lines.

No proof optimization pass. Because the methodology prioritizes *completing* all proofs over *polishing* them, the formalization lacks the compression pass that a human formalizer would perform: merging small helper lemmas back into their parent proofs, shortening verbose Isar scripts, and eliminating redundant intermediate steps. A number of long proofs (e.g., Theorem 30.3) could be substantially shortened with manual attention.

F.4 Structural and Organizational Issues

Ordering. Munkres introduces separations before defining connectedness; the formalization reverses this order. Several lemmas appear far from their natural location—for example, the basis-monotonicity lemma `topology_generated_by_basis_mono_via_basis_elems` appears in §21 rather than §13 where bases are introduced. These placement issues are mostly cosmetic but make the formalization harder to navigate.

Redundant definitions. The predicate `countable` is defined three times: once locally in Chapter 3, once as `top1_countable` in Chapter 4, and it is also available as the identical definition in `HOL-Library.Countable_Set`. A bridging lemma `top1_countable_iff_countable` confirms the equivalence, underscoring the redundancy.

Unnecessary assumptions. A recurring pattern is the inclusion of assumptions that are logically redundant given the other hypotheses. For example, the assumption `is_topology_on` is redundant in lemmas about path reversal where the continuity hypothesis already implies it, and the assumption $a \leq b$ in Theorem 27.1 when the conclusion is vacuously true for $b < a$. While harmless, these weaken the stated results and again reflect the LLM's tendency to include "safe" extra hypotheses rather than proving the sharpest possible statement.

F.5 Library Integration

The formalization's relationship with Isabelle's existing topology library is complex. On one hand, importing `Complex_Main` provides powerful analytical infrastructure; on the other, the explicit-carrier-set design deliberately avoids the type-class-based topology, creating a parallel universe of definitions.

Several sections—notably §24 (connected subspaces of $\mathbb{R}$), §27 (compact subspaces of $\mathbb{R}$), and the path-related definitions—awkwardly mix the two systems, using Isabelle's built-in `topological_space` type class for reasoning about $\mathbb{R}$ while maintaining explicit carrier sets for the general theory. In at least one case (`top1_compact_Icc_linear_continuum`), a lemma is stated entirely in the language of Isabelle's existing type-class topology with no reference to the project's own definitions.

A reconciliation layer—systematically relating `is_topology_on` to Isabelle's `topological_space` class—would be valuable but was not attempted.

F.6 What Went Well

It would be misleading to focus solely on deficiencies. Several aspects of the formalization are genuinely impressive, especially given the 24-day timeline.

Coverage and correctness. All 156 non-exercise numbered results from Munkres' Chapters 2–8 have formal counterparts, and all 806 results are fully proved with zero `sorry`'s. Despite the definitional issues noted above, no proved theorem is *wrong*: the explicit assumptions compensate for the weak definitions at every point of use.

Proof structure. The extreme decomposition into helper lemmas—while verbose—produces proofs that are individually short, independently verifiable, and easy to debug. The main theorems (Tychonoff, Urysohn, Tietze, Nagata–Smirnov, Baire) read as clean assemblies of pre-verified components, making their logical structure transparent in a way that monolithic proofs do not.

Difficult results. The successful formalization of the nowhere-differentiable function (Theorem 49.1), which requires a delicate interplay of topology, real analysis, and uniform approximation, and of the Nagata–Smirnov metrization theorem (Theorem 40.3), with its 38 helper lemmas and intricate metric construction, demonstrates that LLM-assisted formalization can handle genuinely deep mathematics, not merely routine textbook exercises. The final push (March 30 – April 3) proved several further challenging results: the space-filling curve (Theorem 44.1), requiring a Hilbert-curve-style recursive construction with snake-order cell traversal and continuity of the limit function via uniform convergence; the dimension-theory imbedding theorems (Theorems 50.5 and 50.6), involving general-position arguments, partitions of unity, and weighted-sum constructions into $\mathbb{R}^{2m+1}$; and Corollary 45.5, characterizing compact subsets of complete metric spaces as closed and bounded—the very last sorry to be eliminated.

Source traceability. Every definition and theorem is annotated with its Munkres section and line number in the source LaTeX (`(** from §`n `... [top1.tex:`k`] **)`), providing a complete correspondence between the informal and formal texts.

F.7 Summary

The qualitative review reveals a formalization that is *correct but rough*: the theorems are proved and the mathematics is sound, but the definitions carry unnecessary weaknesses, some theorem statements are incomplete relative to Munkres, proofs are unpolished, and the integration with Isabelle's existing infrastructure is ad hoc. These shortcomings are largely systematic—they reflect the LLM's consistent failure modes (defaulting to weak definitions, accepting verbose proofs without compression, adding redundant assumptions) rather than isolated errors. A focused cleanup pass, addressing the definitional issues and compressing proofs, would substantially increase the formalization's value as a reusable library; we estimate this would require days rather than weeks of expert effort, building on the sound foundation that the LLM-assisted workflow has established. Obviously, our analysis of the failure modes done here will also serve for further evolution of the rules given to the LLM agents in future autoformalization experiments.