

Recursive Asterisms

Explore in your own AI chat

A text companion for a professor, researcher or curious reader. Local exhibit companion, October 2026. Intended public website: <https://recursiveasterisms.com/> (not deployed or verified by this task).

Copy the instructions below into your preferred AI chat, then attach or paste evidence-cards.md and research-journey.md from this content packet. No automatic execution or installation is involved.

Copyable starter

Help me explore Recursive Asterisms using the evidence cards and research journey I provide. Begin by asking which result I want to understand and whether I prefer plain language, mathematical detail or formal-evidence detail. Keep historical source identifiers intact.

Distinguish formal statements, reported local replay results, finite experiments, analogies, conjectures and proposed future work. Cite the supplied source identifier for each substantive research claim. Never turn a finite result into a universal theorem or a size bound into a runtime bound.

For a result, state its definitions, quantifiers, assumptions, conclusion, known counterexamples and limits. Explain what would fail if an assumption were removed. Preserve UNKNOWN, REJECTED and adverse results. If evidence is missing, say what is missing rather than inventing it.

Treat explanations as interpretations. Do not claim to have run Lean, checked source files, opened private Notion pages or reproduced experiments unless you actually have the tools, inputs and a successful result. Ask me for an approved source packet if exact proof inspection is needed. Do not request secrets or private engine packages.

Start with: What does the factor-nine theorem say, what exactly is $\kappa(F)$, and why does it not settle P versus NP?

Selected evidence at a glance

1. TRS-LEAN-HAB9-001 / producer_factor9: $|B| \leq 9\kappa(F)$ for the specified arity-at-most-three whole-instance Horn/affine/bijunctive model. Selected-set size, not runtime.
2. B003 / ProtectedBCE: checked blocked-clause deletion preserves protected model projections; reverse repair reconstructs a supplied residual witness. Full model counts may change.
3. B025 / HornCore + HornCNFBridge: correct Horn core with a guarded encoded-CNF bridge. Admitted non-Horn inputs may remain UNKNOWN; malformed inputs are rejected.
4. B022 / SuffixFrame.run_suffix_commutates: adding output suffixes commutes with a fixed number of steps under no-read/no-pop and closed-label support. No termination claim.
5. B039: three chain residuals $49/121/241 \rightarrow 1$ after 18/40/70 elimination steps. Same combined 278-case coverage; one slower measured run. Written and finite computational evidence, not new Lean proof.

The parent replay receipt records 17 positive runs passed and eight deliberate false controls rejected across four frozen families using existing pinned dependencies. Website preparation hash-matched the 25 source/log pairs; it did not replay Lean. This is not external certification.

Questions to take further

- Which quantifier makes this a whole-instance backdoor condition?
- Can you give a small explanatory example of protected projection while clearly separating it from a replayed research result?
- Why can a recurrent search cycle fail to prove UNSAT? What did B027 report?
- What did B039 improve, and why did the full bank's answer coverage stay unchanged?
- Which additional evidence would support a runtime claim?
- What review is needed beyond a formal proof before accepting a shared contribution?
- How does balancing $2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$ become a linear algebra problem? Keep this future-domain illustration separate from the verified Boolean results.
- When I say “multiverse,” do I mean alternative formal contexts, branches of inquiry or a physical cosmological idea? Clarify before proceeding.

The connected programme

The owner-authorized Lean Establishment Programme (TRS-LEAN-ALL-001) records 52 baseline sources, an initial 615 claim/disposition rows and 18 initial formalization groups. These counts are not independent theorem counts or completed groups. It separates proof, intended meaning, runtime and implementation status, and records actual applied proof reuse as distinct from proposed dependencies. The full programme remains open. The website guide includes the B002 factor-nine dependency path and B004/B005 protected-input cost correction, attributed to the programme; these additions were not replayed by the website task.

Boundaries

This companion briefs an AI. It cannot install Lean, access private Notion, run private research, change model weights, confer new capabilities or certify generated replies. It contains no private engine, credentials, executable research payload or proof source package. Review all AI explanations against approved sources.

The network, multilingual explanations and contribution service are proposed directions. The exhibit is not a production research platform, a secure classroom MVP or customer validation. The course's later Lovable MVP requirement and approval gate remain in force.

Source records (visitor access depends on the current Notion sharing settings):

- Research exchange: <https://app.notion.com/p/3ecfa095894c8147ae79e270070cef70>
- Lean programme: <https://app.notion.com/p/3ecfa095894c810ea625ed08e17671a2>

Public methodological reading:

- Lean proof validation: <https://lean-lang.org/doc/reference/latest/ValidatingProofs/>
- Mathlib review guidance: <https://leanprover-community.github.io/contribute/pr-review.html>

- OpenStax balancing equations:

<https://openstax.org/books/chemistry-2e/pages/4-1-writing-and-balancing-chemical-equations>