A formal bridge from Anthropic's refutation of the Exponential Time Hypothesis to a standard multitape Turing machine*

Carlos Toledo[a,*]

[a]*Independent researcher, the cert-machine project, Brazil*

Abstract

In October 2026 Anthropic published a Lean 4 proof that the Exponential Time Hypothesis (ETH) is false. Its statement, `ETHIsFalse`, is the negation of $s_3 > 0$ with s_3 taken over Mathlib's multi-stack Turing machines (`FinTM2`) reading a custom encoding of duplicate-free formulas over the variables $x_0, \ldots, x_{n-1}$; the step from that machine and encoding to a standard model is left to prose. We prove in Lean that `ETHIsFalse` implies ETH is false as stated in the cslib-community library `complexitylib`: for every $\delta > 0$ a deterministic multitape Turing machine decides 3-SAT on the plain binary encoding of every formula within $c \cdot 2^{\delta n} \cdot (L+1)^e + c$ steps. We then port Anthropic's proof (809 files, 762 unchanged, 47 edited for the newer toolchain) into the same Lean, so that the proof, the statement and the bridge are checked in one environment and no step crosses a version boundary. Every step was checked from pinned bytes by Lean's kernel, by nanoda over 77,276 declarations, by `leanchecker --fresh`, and by Lean's judge `lake comparator --paranoid` with six checkers; 9 planted wrong variants were refused. Whether the formal statement is the textbook ETH remains a reading of a definition. The proofs were written by AI agents directed by the author, who also wrote the checker; this is not an independent review.

Keywords: Exponential Time Hypothesis, 3-SAT, formal proof, Lean, machine models, independent checking

*v0.1 of 2026-11-05, built from repository commit 289590ff66bc; archived in release v2026.11.05, doi:10.5281/zenodo.23966840. Every measured number is a macro read at build time from the records named in the Data availability statement; a record that no longer supports a sentence stops the build.

*Corresponding author.

Email address: `carlos@carlostoledo.co` (Carlos Toledo)

1. What is claimed, and what is not

Two AI laboratories published Lean proofs in October 2026 that the Exponential Time Hypothesis of Impagliazzo and Paturi (2001) is false (Anthropic, 2026; OpenAI, 2026). Both proofs are accepted by Lean's kernel, which we confirmed independently for both, against their own statements, with a second kernel (Toledo, 2026). The question that readers raised is not whether the proofs check but what they state: each statement is about a machine and an input encoding of its laboratory's making, and the step from there to a standard machine is argued in prose.

This note closes that step for Anthropic's statement. *Claimed:* a Lean theorem, checked as described in Section 4, that Anthropic's `ETHIsFalse` implies the negation of ETH as defined over cslib's deterministic multitape Turing machine with a binary encoding of arbitrary formulas; and a second Lean theorem in which Anthropic's own proof of `ETHIsFalse`, ported, supplies the hypothesis in the same environment. *Not claimed:* that ETH is false; the correctness or novelty of either laboratory's algorithm outside its Lean development; that a formal definition is the textbook one, which no machine decides (Section 5).

2. The two statements

Anthropic's. `MainResults.lean` at commit 16a5ce41 (Lean v4.33.1, sha256 `47f78a9550d95b37…`) defines `InTime 3` δ: some `FinTM2` machine with finite alphabets and a constant c decide, for every n and every 3-CNF φ on $x_0, \ldots, x_{n-1}$ without repeated clauses, the satisfiability of φ within $\lfloor c\, 2^{\delta n} \rfloor$ steps, reading `encClauses 3` φ. Then $s_3 = \inf\{\delta : \texttt{InTime 3}\ \delta\}$, `ETH` is $0 < s_3$, and `ETHIsFalse` is its negation.

complexitylib's. `ETH` is $\exists\, \delta > 0, \neg$ `DecidableInExpTime (KSAT 3) encCNF numVars` δ, where `DecidableInExpTime` asks for one deterministic multitape Turing machine (cslib's `Turing.MultiTapeTM`) and an exponent e such that the machine decides membership in `KSAT 3` within $O(2^{\delta n}\,(L+1)^e)$ steps on the binary encoding `encCNF` of length L, for every formula: variable labels are arbitrary natural numbers, and literals and clauses may repeat. This is the form of the hypothesis on a standard machine model with a reasonable binary encoding; its distance from the textbook definition is discussed in Section 5.

3. The bridge

```
EthBridge.Anthropic.not_eth_of_ethIsFalse : MainResults.ETHIsFalse
→ ¬ComplexityAbstracts.SubexponentialSATAbstract.ETH
```

Given $\delta > 0$, take $m = \lceil 2/\delta \rceil + 1$ and a `FinTM2` decider at a rate below $1/m$. The proof has four parts. (i) *One step of any program in degree one.* complexitylib already simulates a `TM2` program on the multitape machine in polynomial time, but the degree of that polynomial depends on the program, and here the program is chosen after δ, so the bound would be circular. We prove that a single step of any `TM2` program is computable in degree one, uniformly in the program (`DegreeStep.stepTotalD`). (ii) *The run on a unary clock.* The t-step run is a degree-two fold over the pair (input, t in unary) (`DegreeRun.runBitD`), tied to Mathlib's `TM2OutputsInTime` (`runBit_eq`). (iii) *The reduction.* A polynomial-time map from `encCNF` to `encClauses 3` relabels the variables densely, removes repeated clauses, and passes n along, so that the decider can write $2^{n/m+c_0}$ in unary and run that many steps (`Words.prep_polyTime`, `Prep.prep_spec`: satisfiability and duplicate-freeness preserved). (iv) *The rate.* The composed decider runs within $2^{2n/m} \cdot \mathrm{poly}(L)$ steps, and $2/m \leq \delta$ (`Assemble.decidableInExpTime`).

One environment. The bridge elaborates Anthropic's `MainResults.lean`, byte for byte, against complexitylib's Lean (v4.36.0-rc2) rather than Anthropic's (v4.33.1). To remove that version boundary from the chain, Anthropic's proof (the import closure of its `Solution.lean`, 809 files) was ported to the same toolchain: 762 files are byte-identical; 47 were edited for library renames, missing imports and tactic repairs, each edit listed with both digests in the port's ledger. No edit touches `MainResults.lean`, and every statement changed only by a forced rename is listed. Then

```
EthBridge.AnthropicChain.not_eth : ¬ETH :=
not_eth_of_ethIsFalse MainResults.ethIsFalse
```

is checked in one Lean. The ported proof is a modified copy; since Lean checks whatever proof is supplied against the unchanged statement, the edits affect no decided fact, but the proof checked is not Anthropic's bytes.

4. How it was checked

Each trusted statement was written by hand and pinned by digest before any proof was decided (bridge `c84eae828cc11769…`, chain `229e7202c5a00303…`). On a rented Ubuntu machine, from the published commits (claimbox 235fbfa and c1442b1, on complexitylib 7c66eb71), each run (a) checked the statement block byte for byte against the pin and `MainResults.lean` against Anthropic's published digest; (b) built from source; (c) printed the axioms from our own file: `propext`, `Classical.choice` and `Quot.sound` only, for the theorem and its helpers; (d) exported the theorem with `lean4export` and replayed it in nanoda,

a kernel written independently of Lean's, which accepted 23,430 declarations under the bridge and 77,276 under the chain; (e) ran `leanchecker --fresh`; (f) ran Lean's judge `lake comparator --paranoid` against a challenge written on the machine from the pin, where Lean's kernel, `leanchecker-paranoid`, `lean4lean`, nanoda, `con-leche` and `con-ron` all accepted (the last two over 22,058 and 74,801 declarations). Three planted variants per theorem had to be refused, and were: the statement closed by `sorry` (shown by the axioms file and refused by nanoda), the statement through a new axiom (likewise), and the statement silently changed (refused by the statement check and rejected by the judge for a mismatch). The judge's sandbox could not start on the rented containers; the judge ran without it, and the records say so.

Beside the chain. Two further results use the same checks. The library's own proof that its `ETH` is false (commit 1b7d62a8; nanoda over 49,025 declarations) is a re-proof on the standard machine, not a bridge from either laboratory's statement. And ETH is false on the word RAM with $O(\log L)$-bit words (`EthBridge.WordRAM.not_eth_ram`, nanoda over 49,735 declarations, six checkers): complexitylib's simulator bounds simulated addresses by the running time, which rules out logarithmic words for super-polynomial runs; our variant bounds them by the work space, and complexitylib's model counter runs in polynomial space. On OpenAI's side, a third party's Lean bridge (Hawkley, 2026) proves that OpenAI's theorem implies `ETHIsFalse` inside OpenAI's Lean (v4.34.1); we replayed it with two kernels (Toledo, 2026).

5. What stays open

(1) *The textbook match is a reading.* That complexitylib's `ETH` is the hypothesis of Impagliazzo and Paturi (2001) is a reading of a definition: the machine, the binary encoding, the parameter n and the polynomial factor in L are standard choices, and any further distance is argued, not proved. (2) *OpenAI's side crosses one version boundary.* OpenAI's statement reaches `ETHIsFalse` in OpenAI's Lean; that the copy there and the one here are the same proposition is read, not decided; a port of OpenAI's proof, as done here for Anthropic's, would close it. (3) *The ported proof is a modified copy* (Section 3). (4) *The judge ran without its sandbox*, which protects the judge from a hostile build; the build was the author's. (5) *One author.* The statements were written by the author; the proofs and the port were written by AI agents (Claude, Anthropic) under the author's direction; the checks were run by the author's tooling. Roles are separated, but no second party has replayed the records; doing so is the next step, and the Data availability statement gives what it needs.

Data availability

The Lean sources are in `https://github.com/carlostoledo1891/claimbox/tree/c1442b1/lean/eth-bridge` (the port's ledger is `AnthropicEth/PORT.md`). The records, the pinned statements, the machine scripts and every fact of every run are in `https://github.com/carlostoledo1891/cert-machine`: `certs/eth-chain.json`, `eth-c2.json`, `eth-c1.json`, `eth-comparator.json`, `eth-cslib-d1.json`; `corpus/eth-refutation/`; `tools/runpod-eth-chain.sh`. A summary with the figure is at `https://carlostoledo.co/reports/eth-anthropic-chain.html`.

Declaration of generative AI use

The Lean proofs, the port and parts of the tooling were written by AI agents (Claude, Anthropic) directed by the author, who wrote the statements, chose the checks and takes responsibility for the claims. One of the two laboratories whose claims are discussed is Anthropic.

References

Anthropic, 2026. formal-math: `eth/`, a Lean formalization that the Exponential Time Hypothesis is false. `https://github.com/anthropics/formal-math`, commit 16a5ce41.

Hawkley, D., 2026. OpenAI–Anthropic ETH cross-formalization (RPI-ETH-010). `https://formalize.now/378/`.

Impagliazzo, R., Paturi, R., 2001. On the complexity of k-SAT. J. Comput. Syst. Sci. 62, 367–375.

OpenAI, 2026. The Exponential Time Hypothesis is false; and a Lean formalization in `https://github.com/openai/math`.

Toledo, C., 2026. Two claimed refutations of ETH, checked. `https://carlostoledo.co/eth/`.