Long, wide, and trivial: a nondeterministic time hierarchy in Isabelle (progress report)

András Z. Salamon[1] and Michael Wehar[2]

[1] School of Computer Science, University of St Andrews, Scotland, UK
Andras.Salamon@st-andrews.ac.uk
[2] Computer Science Department, Bryn Mawr College, PA, USA
mwehar@brynmawr.edu

Abstract

We report on an Isabelle formalisation of the nondeterministic time hierarchy theorem. Tool-using LLM agents in Claude Code construct the proof. The current attempt is at about 19,000 lines for the linear speedup theorem so far, after two earlier attempts ran into problems. Prior formalisations of analogous results (the AFP *Universal Turing Machine* entry, Asperti's Matita formalisation of the deterministic hierarchy theorem) omit the tedious step counting that dominates our work. We propose classifying proof shape along three axes: the *length* of the reasoning chain, the per-step *width* of the state vector, and the per-step *content difficulty*. Bounded-resource analyses of computational models occupy a *long-wide-trivial* region: long chains of mechanical steps, each over a wide machine configuration. We conjecture that this region is hard for human working memory but tractable for LLMs with long context windows. Our progress improved markedly when we switched from a 200K-token to a 1M-token context window in March 2026. Three workflow disciplines seemed essential: a detailed markdown specification of the architecture, a watchdog wrapper around `isabelle build`, and an evolving project knowledge base consulted before any non-obvious tactic. Quantifying width is a key open problem.

1 Three Axes for Proof Shape

Formal proofs have three axes: the length of the reasoning chain, the width of the state vector each step carries, and the per-step content difficulty. "Wide" and "narrow" here mean the information carried per step, analogously to proof-complexity usage. Most ATP work has focused on narrow-state proofs: SAT solvers and saturation-based first-order provers arguably fall into this region. Large mathematical formalisations like Feit–Thompson or Liquid Tensor are mostly long and nontrivial.

Bounded-resource analyses of computational models occupy a different region: *long-wide-trivial*. Each step is mechanical. But every step operates on a machine configuration that travels with the proof: tape contents, head positions, internal state, step counter, simulation invariants. Previous work has avoided this region. Asperti's Matita formalisation of the deterministic hierarchy theorem proved the diagonalisation direction in 472 lines and *axiomatised* the universal-simulation direction [1]. A recent Isabelle example in this region is Balbach's Cook–Levin formalisation [2] (56,514 lines of Isar), which tracks polynomial upper bounds on multi-tape-Turing-machine running times. The hierarchy theorem needs accounting precise enough to compare specific time bounds.

2 Three Attempts

In early 2026 we made three attempts at proving the nondeterministic time hierarchy theorem.

Attempt 1 (January–March, 6,424 lines of Isar) proved that a constructed diagonal language lies outside the smaller complexity class. Diagonalisation finished quickly, but universal simulation stalled.

Attempt 2 (March–April) had a more detailed blueprint and better tooling, but no pre-committed low-level architecture. The universal machine ended up with different layouts between its tapes, requiring case splitting in otherwise routine proofs (and hence fast-growing proof bodies as the project continued), and work to make the foundations more uniform ended up a similarly hydra-like task. When we abandoned it, attempt 2 was 42,612 lines of Isar, producing a 927-page generated PDF proof document.

Attempt 3 (April–May, ongoing) opened with a detailed markdown specification based on the AFP *Multitape_To_Singletape_TM* substrate [4]. While writing the spec we found that the crucial nondeterministic tape reduction component of the hierarchy proof relies on the foundation of the linear speedup theorem [5], so we reduced our scope to just this result. Attempt 3 stands at approximately 19,000 lines, with lemma statements in place and proof bodies in progress, capturing the basic "group k consecutive input symbols so the simulator runs k times faster" method.

The AFP *Universal_Turing_Machine* entry [6] (37,554 lines) and Asperti's Reverse Complexity programme both forgo step-counted analysis, which we cannot do.

3 Workflow and Tooling

The first essential discipline is *spec-first* design. Every architectural decision is committed in markdown (currently 5,375 lines) before any theory work. A design memo per theorem fixes simulation invariants and the proof outline. The LLM then works against a predetermined architecture. In contrast, the LLM works poorly when asked to make architectural choices mid-proof. Attempt 2 collapsed for that reason.

Some other infrastructure seems to have been essential for this project. A watchdog wrapper around `isabelle build` enforces timeouts and cleans up orphaned subprocesses (around 2,500 invocations across all attempts). A project knowledge base is consulted before any non-obvious tactic, for project-specific Isar patterns, and insights that might be useful in other projects. This replaces long tactic-swap-and-retry sequences with a quick lookup and an attempt to reuse a previously useful approach.

In March 2026 our rate of progress jumped sharply. We adopted Claude Opus's 1M-token context window option (initially version 4.6, currently 4.7), replacing the 200K-token default. The 200K window had been a real obstacle, requiring constant workarounds. This suggests that the long-wide-trivial region is limited by context-window capacity, not by reasoning capacity. Recent hammer-reconstruction work in interactive theorem proving [3] addresses a different problem: ATP-replaceable subgoals rather than wide proof state.

4 Open Questions

Three open questions remain. (i) How to measure the width axis directly. Line count conflates the three axes. Candidate metrics include the average and maximum number of live hypotheses per step, the ratio of fact-introducing to fact-consuming lines, and the per-step proof-state textual size. These need validation against LLM-tractability. (ii) Is the de Bruijn factor (the ratio of formal to informal lines of proof) for long-wide-trivial regions unusually large in Isabelle, or is it just as bad in Lean/Rocq? A translation to Lean or Rocq would shed light on this. (iii)

Whether width scales manageably as primitives are assembled into the universal machine used to simulate the core diagonal construction. We can only answer this by finishing the proof.

References

[1] Andrea Asperti. Reverse complexity. *Journal of Automated Reasoning*, 55(4):373–388, 2015. https://doi.org/10.1007/s10817-015-9349-x.

[2] Frank J. Balbach. The cook-levin theorem. *Archive of Formal Proofs*, January 2023. https://isa-afp.org/entries/Cook_Levin.html, Formal proof development.

[3] Chad E. Brown, Cezary Kaliszyk, Martin Suda, and Josef Urban. Hammering higher order set theory. In *Intelligent Computer Mathematics (CICM 2025)*. Springer, 2025. https://arxiv.org/abs/2509.08264.

[4] Christian Dalvit and René Thiemann. A verified translation of multitape Turing machines into singletape Turing machines. *Archive of Formal Proofs*, November 2022. https://isa-afp.org/entries/Multitape_To_Singletape_TM.html, Formal proof development.

[5] Juris Hartmanis and Richard E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965. https://doi.org/10.1090/S0002-9947-1965-0170805-7.

[6] Jian Xu, Xingyuan Zhang, Christian Urban, Sebastiaan J. C. Joosten, and Franz Regensburger. Universal Turing machine. *Archive of Formal Proofs*, February 2019. https://isa-afp.org/entries/Universal_Turing_Machine.html, Formal proof development.