Ripple: AI-Formalized Framework for CRN Computation

Ho-Lin Chen[1] Xiang Huang[2]

[1]National Taiwan University [2]University of Illinois Springfield

Supported by NSTC (Taiwan) grant 113-2221-E-002-204-MY3 and US DOE Office of Science ASCR grant DE-SC0024278.

Overview

AI agents can now prove theorems autonomously — settling open conjectures, building formal proofs from scratch.

If we want to use AI to prove theorems in our field reliably, what do we need?

Mathlib gives us general-purpose foundations. But to formalize a specific field — like molecular programming — we also need **domain-specific** infrastructure: definitions, compilers, and theorems that Mathlib does not carry.

Ripple is that missing layer for chemical reaction networks: an open Lean 4 library of CRN dynamics, population protocols, and stochastic foundations, built so that AI agents — and humans — can prove molecular-programming theorems on solid ground.

What Ripple provides:

1. The **GPAC (General Purpose Analog Computer) / CRN continuum** and the CRN-computable reals
2. **Stochastic population protocols** and a CTMC layer bridged to the deterministic limit by Kurtz's mean-field theorem
3. **Turing completeness**: continuous ODE and stochastic CRN with bounded error
4. **Population-protocol majority** formalizations

Scale: over a million lines of Lean 4.

No `sorry`. No `axiom`. Core constructions depend on exactly 3 Mathlib foundational axioms.

How We Did It: The AI + Lean Workflow

The loop:

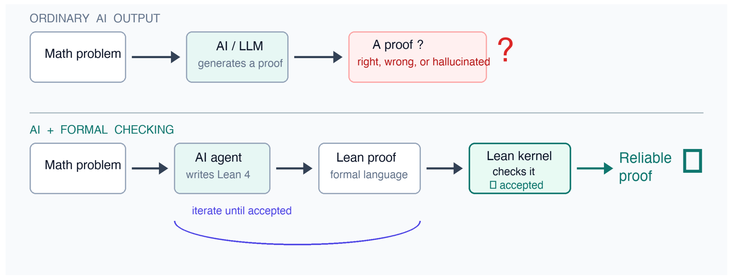
1. **Human** selects targets, designs acceptance criteria, reviews output. Agents read the papers and do the math.
2. **AI agent** writes Lean 4 definitions and proofs from the source paper.
3. `lake build` — the kernel accepts or rejects. No ambiguity.
4. **Iterate** until clean. `#print axioms` — no `sorry`, no hidden axioms.

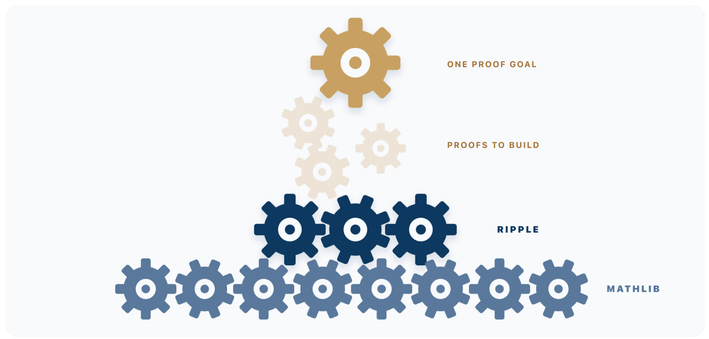
A curated **Lean playbook**, distilled from 1M+ lines of formalization:

- What counts as an **acceptable proof** (no `sorry`, no `native_decide`, no vacuous hypotheses)
- How to **orchestrate agents** in parallel on independent chapters
- How to **audit** output and what to do when **stuck**

Agents: Claude Code (Opus 4.6–4.8, Fable 5), Codex (GPT-5.4–5.6).
Cost: 4 months, under $2,000 total.

What Formalization Buys





1. **Reliable.** The kernel catches every gap; domain expertise is not a prerequisite for trust.
2. **Exposes gaps** in published work.
3. **Extensible** to other molecular-programming models (tile assembly, etc.).
4. **Enables new results** — even someone unfamiliar with the field can dependably prove new theorems.

Future Work

- **Human review** of AI-produced proofs for faithfulness.
- **Refactor** to consolidate duplicated infrastructure.
- **Curate**: distill essential structures into a clean, minimal library.
- **Mathlib tracking**: continuous integration against upstream updates.
- **Upstream**: contribute CTMC, Kurtz, Frobenius to Mathlib.

Formalized Papers

CRN-computable numbers

Huang, Klinge, Lathrop, Li, Lutz (UCNC 2017)	CRN-computable reals
Huang, Klinge, Lathrop (DNA 2019)	CRN = analog computer
Huang & Huls (DNA 2022)	Large-population protocols
Huang & Migunov (CF 2024)	Analog-to-protocol compiler
Haisler, Huang, Migunov et al. (UCNC 2025)	Selective dual-railing
Chen & Huang (DNA 2026)	Bounded analog complexity
QBee compilation pipeline	Automated quadraticization
Zero-init no-collapse (new)	Bounded CRN non-degeneracy
Albenque & Gerin (2012)	Urn model / algebraic numbers
Cassels (1960)	Catalan equation descent
$\zeta(3)$, π, e, $\ln 2$, γ, Catalan, Dottie	Computable constants

Population-protocol majority

Angluin, Aspnes, Eisenstat (2008)	Approximate majority
Doty et al. (FOCS 2021)	Exact majority
Burman, Chen, Doty et al. (PODC 2021)	Self-stabilizing leader election
Kanaya et al. (SSS 2025)	Self-stabilizing majority

Turing universality

Bournez, Graça, Pouly (JACM 2017)	ODE Turing completeness
Soloveichik, Cook, Winfree, Bruck (2008)	Stochastic CRN universality

Kurtz's mean-field theorem

Kurtz (1970)	Weak convergence
Kurtz (1978)	Strong approximation
CLT-scale	$\mathbb{E}[\sup \|\bar{X}^N - x\|^2] \le C/N$

Supporting mathematics

Frobenius regular-singular theory	ODE analysis
Modular forms / Chudnovsky / Ramanujan	Number theory
CTMC / Doob / Bennett–Freedman	Probability

This is not because any of it was easy. For example, the Bournez–Graça–Pouly (JACM) and Doty et al. (FOCS) each took 8–9 weeks of continuous agent work. It is because agents run in parallel, around the clock, on independent chapters. A single human team would need years to formalize all of the above.

GitHub Repo **arXiv Paper**

LaTeX TikZposter