A Lean 4 Formalisation of OpenAI's Proof of the Cycle Double Cover Conjecture

Vaibhav Bajpai, Johannes Carmesin, and Utku Okur
TU Freiberg

Abstract

OpenAI proved the cycle double cover conjecture. We provide a complete Lean 4 formalisation of the proof.

OpenAI [2] proved that every bridgeless graph has a cycle double cover; that is, a family $(C_i \mid i \in I)$ of cycles such that every edge belongs to exactly two of the cycles C_i. We provide a complete Lean 4 formalisation of the proof. Thus, assuming Lean 4 is correct, it remains only to verify that the following statement correctly expresses the cycle double cover conjecture:

```
/- A binary edge vector of `G`. -/
abbrev EdgeVector :=
  G.edgeSet → ZMod 2

/- An edge vector is Eulerian if the sum of its coefficients over the
edges incident with `v` is zero at every vertex. -/
def IsEulerianVector (C : G.EdgeVector) : Prop :=
  ∀ v : V, ∑ e ∈ (G.incidenceFinset v).subtype (· ∈ G.edgeSet), C e = 0

/- A finite family of edge vectors is a cycle double cover if every member
is Eulerian and every edge occurs in exactly two members. -/
def IsCycleDoubleCover {n : ℕ}
    (C : Fin n → G.EdgeVector) : Prop :=
  (∀ i, G.IsEulerianVector (C i)) ∧
    ∀ e : G.edgeSet, (Finset.univ.filter fun i => C i e ≠ 0).card = 2

/- **Cycle double cover conjecture** : every bridgeless cubic
simple graph has a cycle double cover. -/
theorem exists_cycleDoubleCover
    (hcubic : G.IsRegularOfDegree 3)
    (hbridgeless : ∀ e, ¬G.IsBridge e) :
    ∃ (n : ℕ) (C : Fin n → G.EdgeVector), G.IsCycleDoubleCover C := by
```

The full, `sorry`-free Lean 4 proof[1] is available as release `v1.4`. The formalisation actually proves the slightly stronger statement for all bridgeless multigraphs, as well as the statement for simple cubic graphs displayed above. We used assistance from Claude Code (Fable 5, Ultracode+xhigh+ workflows), Codex (ChatGPT 5.6 Sol, extra high reasoning effort), and Nelson's library [1] in producing the formalisation.

[1] Repository: `https://github.com/utkuokur/CDC_lean`. Lean version: v4.30.0-rc2; Mathlib version: `1b045b9df0`.

References

[1] Peter Nelson. Matroid: A formalization of matroid theory in Lean 4. `https://github.com/apnelson1/Matroid`, 2026.

[2] OpenAI. A proof of the cycle double cover conjecture. `https://cdn.openai.com/pdf/04d1d1e4-bc75-476a-97cf-49055cd98d31/cdc_proof.pdf`, Preprint 2026.