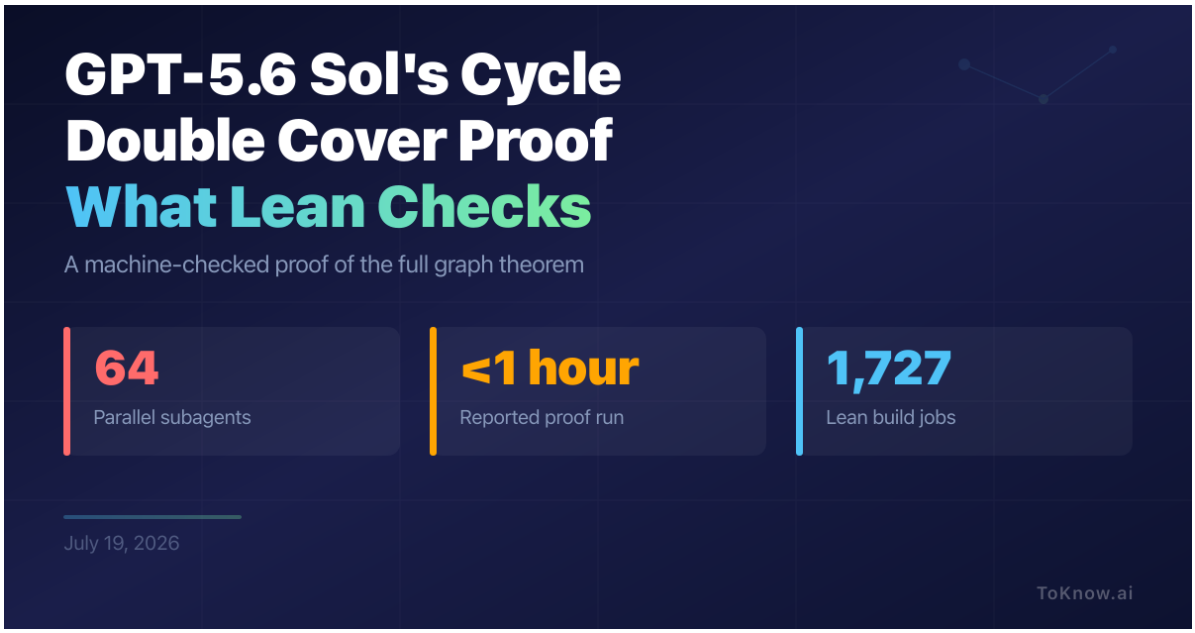
GPT-5.6 Sol's Cycle Double Cover Proof: What Lean Actually Checks

Kabui, Charles

2026-07-19

Read at ToKnow.ai



OpenAI says GPT-5.6 Sol Ultra used **64 subagents in just under one hour** to produce a three-page proof of the Cycle Double Cover Conjecture. The conjecture asks whether every finite graph without a bridge, an edge whose removal disconnects the graph, has cycles that cover each edge exactly twice. The proof first reduces the problem to cubic graphs, where every vertex has three edges. It then uses the Jaeger-Kilpatrick eight-flow theorem to label

edges with nonzero values from a group of eight elements. A short linear algebra argument turns those labels into the required cycles. Graph theorist Thomas Bloom called it "a very nice proof".

The stronger correctness check is a separate 7,224-line Lean formalization. Lean's small trusted kernel checks every proof step against the stated definitions. OpenAI's verification record reports **1,727 build jobs**, no warnings, no unfinished `sorry` or `admit` placeholders, and no project-specific axioms. Its final theorem covers finite loopless bridgeless multigraphs, including parallel edges. That verifies the formal statement encoded in Lean. It does not show that the formalization was created during the one-hour run, and mathematicians must still inspect whether its definitions match the conventional conjecture. This is why machine-checked source matters: readers can audit more than polished prose.

Read More: OpenAI's Reasoning Model Disproves a 1946 Erdős Conjecture in Geometry

Sources:

- OpenAI's proof
- OpenAI's full prompt
- Lean verification record
- Final Lean theorem
- Thomas Bloom's proof commentary
