AUTOFORMALIZING EUCLID

MICHAEL BEESON

Abstract. We describe the use of Claude Opus 4.7 to assist in the construction of formal proofs of the propositions in Euclid Book III.

Euclid's Book III deals with circles, which occur only twice in Book I. Building upon earlier (joint) work formalizing and computer-checking Book I, the formalization of Book III was in progress, but very slow progress, when Claude arrived to help. This abstract describes how I worked with Claude to produce formal proofs of Euclid's propositions.

First, any formalization needs a target theory: ours is the set of axioms and definitions we used for Euclid Book I, plus one more axiom needed for Book III, to express Euclid's definition that "circles are *planar* figures." See [3, 2, 1]. We call that target theory $\mathbb{E}$.

Second, to formalize a theorem, we need a formal *statement* of the theorem. In Euclid, that is often difficult, since the statement may involve notions that Euclid himself did not define. For example, two points being on the *same side* or *opposite sides* of a line. In our work on Book I, we used Tarski's definition of that notion. For a new example from Book III, Prop. III.8 involves a circle, and lines "falling on the convex circumference" from a point D outside the circle, and the notion of two such lines being on the same side of the diameter, and of one such line being "nearer the center" than the other.

The process of formalization (with or without help) is this:

- Get a formal statement of the proposition.
- Write a sketch of the proof in the target theory $\mathbb{E}$.
- Iteratively, write a few steps of formal proof and run them through the proof-checker; debug until they check, then repeat.

The AIs available in 2025 were useless at these tasks. But the point of this report is that Claude Opus 4.7 is rather good at the *write a few steps of formal proof* part. I will refer to that model just as "Claude" here. Here is how we work together, once the conversation is in progress:

- I write the formal proposition.
- I write a sketch of the proof in informal mathematical text.
- I give that text to Claude asking for a formal proof.
- Probably Claude fails. There ensues a conversation about what is needed.
- Between us we may decide that we need a lemma formulated separately. The exact formulation of that lemma may undergo interactive changes.
- Sometimes Claude can prove the lemma in one go.

Date: May 12, 2026.

- Other times Claude needs a tighter proof sketch. For example, Claude usually cannot apply axioms known as `innerPasch` and `outerPasch` without having a very detailed description of the matching of variables to be used.
- But after considerable discussion, Claude will present a formal proof for me to check. (As opposed to having Claude Code check it.) These proofs can be up to 200 lines long; it costs me many hours to produce them by hand.
- If they do not check, Claude can sometimes help identify what is missing. Sometimes the informal proof turns out to have gaps. But once the informal proof is correct, Claude succeeds to find a formal version, usually after just a few more iterations to supply missing intermediate formulas. The formal proof then comes out all at once, often more than a hundred good lines of formalized reasoning, with an effect like a fireworks display.

This process can take several hours per proposition. We are still far from the stage where Claude could read Euclid and produce formal proofs. But still this is *autoformalization* in the sense that it is Claude who produces the formal proof, after a long interactive discussion in English and informal mathematics, starting with proof steps described by the human, often in quite some detail, but still nowhere near *formal* detail.

The proof-checker I use is a custom checker, which was used for the earlier work on Book I. Since it is not a famous checker, and I am probably the only person who trusts it, the Book I proofs were translated in Coq and HOL Light by my co-authors. Eventually I will have Claude do the same for the Book III proofs, and translate everything to Lean as well. Claude learned the syntax required for these proofs in a few seconds, by reading the source code of the proof checker and about a dozen sample proof files.

The above description tells how Claude and I work together on a proof. Here is how we begin a session. We use one conversation for one proposition. To begin a new conversation, I upload `Axioms.php`, checkproofs.php, a dozen or so sample `.prf` files, and a PDF of a short TeX document with a proof sketch of the proposition to be proved in this conversation. Claude reads these immediately and is ready to go, despite having no memory at all of what we did yesterday or last week. You can continue a conversation to the next proposition, but eventually Claude runs out of "context memory", and it turns out to work fine to just have one conversation per proposition. The resulting formal proof is archived and available for further proofs.

This is work in progress, not completed work.

References

[1] Michael Beeson. Euclid after computer proof-checking. *The American Mathematical Monthly*, 129(7):623–646, 2022.

[2] Michael Beeson. On the notion of equal figures in Euclid. *Beiträge zur Algebra und Geometrie*, 2022.

[3] Michael Beeson, Julien Narboux, and Freek Wiedijk. Proof-checking Euclid. *Annals of Mathematics and Artificial Intelligence*, 85(2):213–257, January 2019.

Michael Beeson, San José State University (Professor Emeritus), University of California at Santa Cruz (research associate) `profbeeson@gmail.com`