15

The Philosophical and Pedagogical Implications of a Computerized Diagrammatic System for Euclidean Geometry

Nathaniel Miller
University of Northern Colorado

1 Introduction

This chapter briefly describes the computer proof system CDEG, version 2.0, a computerized formal system for giving diagrammatic proofs in Euclidean Geometry. The existence of such a computer proof system shows that geometric arguments that rely on diagrams can be made rigorous, and this fact has important philosophical implications for how we understand and teach geometry. The full details of the computer system and its underlying logic are complicated and are discussed elsewhere; this chapter instead gives a brief example of how the computer system can be used to prove a theorem of Euclidean geometry, and then discusses some of the philosophical implications of such a system.

The chapter starts with a brief history of the use of diagrams in geometry, and explains why we might be interested in such a formal system. It then gives an introduction to what constitutes a geometric diagram in this context, and how such diagrams are manipulated by this computer system. As an example, it shows how Euclid's first proposition from Book I of the *Elements* [1] can be derived in this system. CDEG is meant to formalize the proof methods found in Euclid's *Elements*, and the following sections of this chapter discuss the correspondence between these two proof systems, the question of why we would want to develop a computer system that mimics Euclid's proof practice, and other related philosophical issues. Finally, the last section of this chapter discusses some ways these philosophical issues might bear on the way we teach geometry courses.

CDEG stands for "Computerized Diagrammatic Euclidean Geometry." This computer proof system implements a diagrammatic formal system for giving diagram-based proofs of theorems of Euclidean geometry that are similar to the informal proofs found in Euclid's *Elements*. It is based on the diagrammatic formal system FG, which is described in detail in my book, *Euclid and his Twentieth Century Rivals: Diagrams in the Logic of Euclidean Geometry* [7]. The book also describes an earlier version of CDEG; however, CDEG has evolved significantly since its publication, and is now publicly available in a beta version.

Euclid's *Elements* was written around 300 B.C. and was considered the gold standard of careful reasoning and mathematical rigor for roughly the next 2000 years. However, this changed with the development of the idea of a formal proof system in the late 1800s. A formal proof system is one in which all the allowable rules are set out carefully in advance in a way that can be followed mechanically, so as to leave no room for human error. Most formal proof systems are *sentential*, meaning that they manipulate sentences that are strings of characters in some formal language. The notion of a formal proof system is, in a way, the logical culmination of Euclid's proof method, in which a set of definitions, postulates, and common notions are set out in advance as assumptions, and proofs rely on them and on previously proven propositions. However, it has been commonly assumed that Euclid's *Elements* could not form the basis for a formal proof system, because his proofs make crucial use of diagrams as part of their arguments. The rules for using diagrams in proofs have not been well understood, and this made it seemingly impossible to incorporate these kinds of proofs into a traditional sentential formal system. Thus, in the twentieth century, the most commonly held view became that Euclid's proofs, as well as all other proofs using diagrams, were inherently informal and could not be made rigorous. The comments made by Henry Forder in *The Foundations of Euclidean Geometry* in 1927 are typical of this view: "Theoretically, figures are unnecessary; actually they are needed as a prop to human infirmity. Their sole function is to help the reader to follow the reasoning; in the reasoning itself they must play no part." [2, p. 42].

During the twentieth century, two approaches were used to give a rigorous foundation for Euclidean geometry. The first was to create sentential axiom systems for geometry that contained many more axioms than Euclid's. Several people proposed sets of axioms, the most famous of which is Hilbert's [3]. This approach gives rise to what is commonly referred to as "synthetic geometry." Another approach is to encode geometry within the theory of the real numbers via the Cartesian plane. This was the approach taken by Tarski [13], who showed that this provides a decision procedure for the parts of geometry that can be encoded this way—that is, a mechanical procedure that can determine if a statement is true or false (although it make take an impractical amount of time to actually come to a decision). This approach is generally referred to as "analytic geometry." Both approaches yield completely formal ways to prove theorems of Euclidean geometry. (See [7, Section 1.1] for a more extensive history of geometry and the use of formal systems in geometry.)

However, the proofs produced by the approaches generally look nothing like Euclid's proofs. So, while they may formalize Euclid's theorems, they do not formalize his proof methods. They therefore shed no light on the question of whether or not Euclid's proof methods can be made formal. In [7, Section 1.2], I propose the *formality hypothesis*:

Formality Hypothesis An informal proof method is sound if and only if it is possible to give a formal system with the property that informal proofs using the informal methods can always be translated into equivalent correct proofs in the formal system.

CDEG is designed as a diagrammatic computer proof system that can mimic Euclid's proofs in a completely rigorous way, and can show, via the formality hypothesis, that his proof methods that employ diagrams are valid modes of informal reasoning.

When we say that CDEG is a diagrammatic computer proof system, it means that it allows its user to give geometric proofs using diagrams. It is based on a precisely defined syntax and semantics of Euclidean diagrams. To say that it has a precisely defined syntax means that all the rules of what constitutes a diagram and how we can move from one diagram to another have been completely specified. The fact that the rules are completely specified is perhaps obvious if you are using the formal system on a computer, since computers can only operate with precisely defined rules. However, it was commonly thought for many years that it was not possible to give Euclidean diagrams a precise syntax, and that the rules governing their use were inherently informal.

To say that the system has a precisely defined semantics means that the meaning of each diagram has also been precisely specified. In general, one diagram drawn by CDEG can actually represent many different possible collections of lines and circles in the plane. What they all share, and share with the diagram that represents them, is that they all have the same topology. This means that any one can be moved and stretched into any other, staying in the plane. So, for example, a diagram containing a single line segment represents all possible single line segments in the plane, since any such line segment can be stretched into any other. See [7] for more details concerning the syntax and semantics of Euclidean diagrams.

2 CDEG Diagrams

A sample CDEG diagram is shown in Figure 1. It occurs in the proof of Euclid's first proposition and represents a line segment along with a circle drawn with that line segment as its radius.

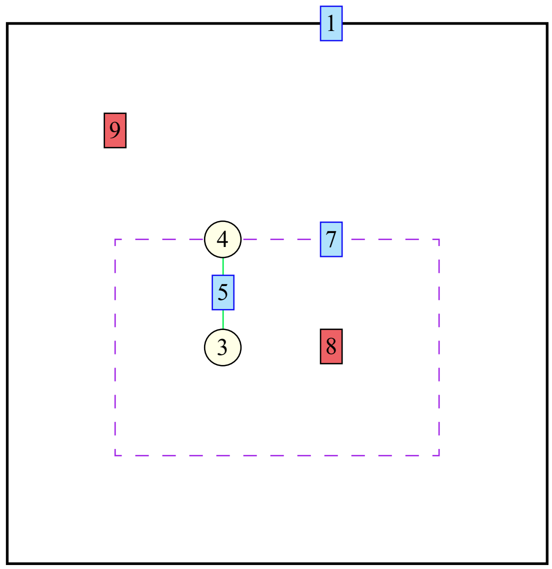

Figure 1. A CDEG diagram showing the second step in the proof of Euclid's first proposition.

CDEG diagrams contain two kinds of objects: *dots* and *segments*. The dots, which are shown as light yellow circles, represent points in the plane, while the segments represent pieces of lines and circles. There are actually two kinds of segments that can occur in a diagram: *solid segments*, which represent pieces of lines, and *dotted segments*, which represent pieces of circles. The dots and segments of a diagram are enclosed by a bold line called the *frame*. The dots and segments of a diagram, along with the frame, break the diagram into a collection of *regions*. Every dot, segment, region, and piece of the frame in a CDEG diagram is labeled with a number, so that we can refer to it. Dots are labeled with a number inside the yellow circle; segments are labeled with a number in a blue box along the segment, and regions are labeled with a number in a red box somewhere in the midst of the region.

The segments in a diagram are part of diagrammatic lines and circles. Each line and circle is assigned a different color, so that all the segments that make up a line or circle will be the same color. Lines may continue to the frame, in which case we consider them to be infinite in that direction. Thus, infinite lines intersect the frame in two places, rays intersect it in one place, and line segments don't intersect it at all.

Each CDEG diagram represents all the possible equivalent collections of points, lines, and circles with the same topology as the diagram. However, it is often the case that when we perform some geometric operation on a diagram, such as connecting two points with a line, there may be multiple possible outcomes depending on which of the equivalent collections you start with. CDEG allows for this possibility through the use of *diagram arrays*. A diagram array is a collection of several possible diagrams showing different possible outcomes.

CDEG allows you to manipulate diagrams through the use of construction and inference rules. These rules are meant to be *sound*. This means that if you start with a diagram D that represents a collection $\mathcal{C}$ of points, lines, and circles in the plane, and you apply a rule to D, then at least one of the diagrams in the resulting diagram array should still represent $\mathcal{C}$ (or, in the case of a construction rule that added a line or circle to the diagram, $\mathcal{C}$ with the appropriate line or circle added). The definition of soundness is the diagrammatic analogue of the normal logical definition of soundness, which is that a formal system is sound if the outputs of its rules only produce statements that are logical consequences of its inputs, so that the system can only produce true statements. For further details, see [7].

3 A Sample CDEG Session

As a brief example, this section shows how to reproduce the proof of Euclid's Proposition 1 from Book I of the *Elements*. A more detailed version is given as a tutorial in the *CDEG User's Manual* [5]. Euclid's Proposition 1 shows that an equilateral triangle can be constructed on any given base.

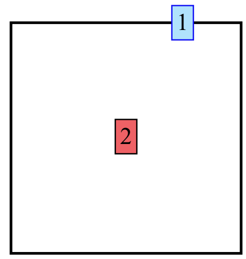

Figure 2. The empty primitive diagram as drawn by CDEG.

First we start CDEG and ask it what commands are available:

```
Welcome to CDEG!
(Type h for help.)
CDEG(1/1)% h
Options are:
<a>dd dot to segment, draw <c>ircle,
<d>elete objects, <e>rase diagram, get <h>elp,
apply <m>arker inference rules, con<n>ect dots,
extend segment in <o>ne direction,
<p>rint diagram as text, <q>uit,
add dot to <r>egion, <s>ave/load diagrams,
se<t> pd, or e<x>tend segment.
CDEG(1/1)%
```

The prompt here (`CDEG(1/1)%`) tells us that we are currently working with the first diagram in a diagram array that contains one diagram. Since we have just started the program, this is the empty diagram. This is the initial diagram that is displayed. It is shown in Figure 2. It contains a single region bounded by the frame; CDEG has assigned this region the number 2. CDEG assigns each object in a diagram a unique number by which it can be identified. Next, we use the "r" command ("`add dot to <r>egion`") to add two new dots to this region:

```
CDEG(1/1)% r
Enter region number: 2
CDEG(1/1)% r
Enter region number: 2
```

CDEG now displays the resulting diagram, which adds two more dots, numbered 3 and 4. We can connect them using the `con<n>ect dots` command.

```
CDEG(1/1)% n
Enter first dot's number: 3
Enter second dot's number: 4
```

The resulting diagram is shown in Figure 3; this is the starting diagram for our proof of Euclid's Proposition 1. Next, we will `draw a <c>ircle` centered at dot 3 and going through dot 4.

```
CDEG(1/1)% c
Enter center dot's number: 3
Enter radius dot's number: 4
```

The resulting diagram is the one that was shown in Figure 1 in Section 2. The diagrammatic circle in the diagram looks rectangular rather than circular, but all we care about here is the topology of the diagram.

Next, we want to draw another circle centered at dot 4 and going through dot 3. We will then form a triangle by connecting the endpoints of the segment to one of the points, dot number 12, on the intersection of the two circles.

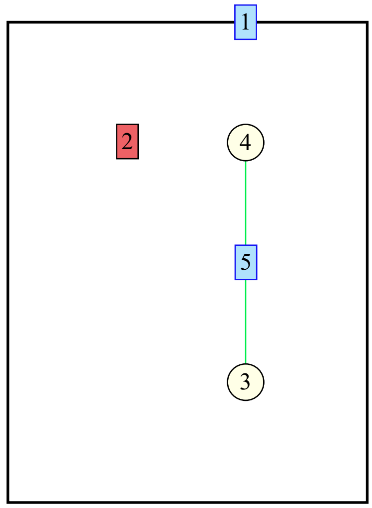

Figure 3. A CDEG diagram showing a single line segment.

```
CDEG(1/1)% c
Enter center dot's number: 4
Enter radius dot's number: 3
CDEG(1/1)% n
Enter first dot's number: 3
Enter second dot's number: 12
CDEG(1/1)% n
Enter first dot's number: 4
Enter second dot's number: 12
```

The resulting diagram is shown in Figure 4.

Next, we can mark segment 5 congruent to segment 25, because they are both radii of the purple circle in Figure 4, and, likewise, we can mark segment 5 congruent to segment 29, because they are both radii of the red circle. We can then mark all three of the segments congruent to one another using the `<c>ombine markers` command that is available as part of the `apply <m>arker inference rules` submenu. For more details of how this works, see the extended version of this example given in the user manual [5]. The markings of the diagram are actually given as text accompanying the diagram. Representations that combine text with diagrams are sometimes referred to as heterogeneous representations; thus, this is an example of a heterogeneous representation.

Finally, we may want to clean up the diagram by erasing the superfluous pieces added in the course of our construction. We can do this using the `<d>elete objects` command, leaving us with the diagram shown in Figure 5, which shows an equilateral triangle on our original base.

Thus, we have shown how to construct an equilateral triangle on a given base, duplicating Euclid's Proposition 1.

4 CDEG vs. Euclid's *Elements*

CDEG is designed so that there is a direct correspondence between the construction and inference rules that it uses and the postulates, common notions, and definitions that Euclid uses in the *Elements*. For example, the CDEG proof of Euclid's Proposition 1 that was given in Section 3 directly follows Euclid's proof of the proposition. For an extended discussion of all of CDEG's commands, explanations of how each is used, and how they correspond to Euclid's postulates, common notions, and definitions, see the *CDEG User's Manual* [5].

CDEG does not contain many inference rules other than those set out in Euclid's postulates, common notions, and definitions. It is commonly asserted that Euclid's *Elements* contains many subtle gaps that can only be repaired by adding further postulates in the manner, for example, of Hilbert's *Foundations of Geometry* [3]. The added postulates typically have to do with issues of betweenness, continuity, and the intersections of geometric objects. For example,

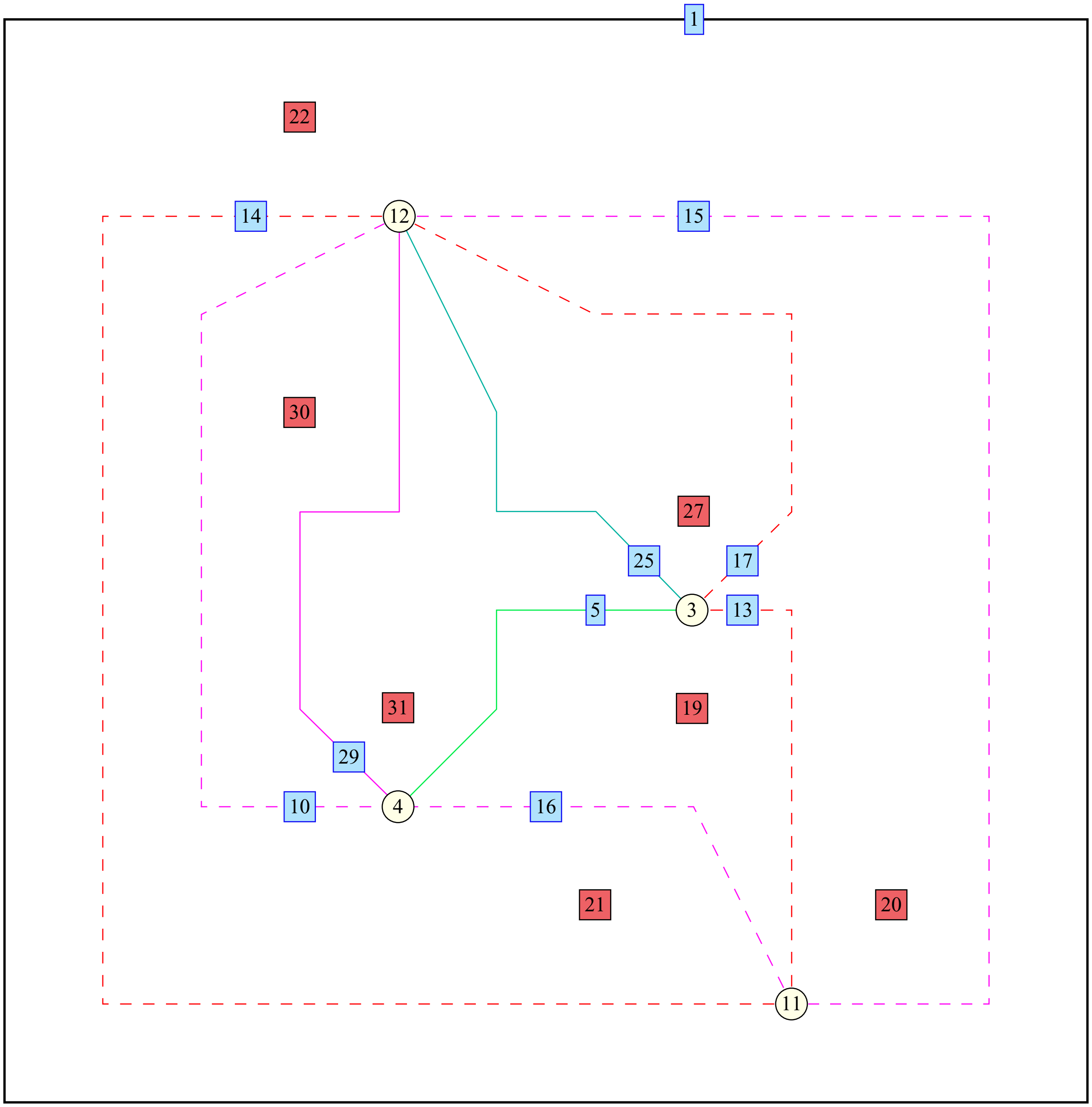

Figure 4. A CDEG diagram showing the fourth step in the proof of Euclid's first proposition

in the proof of Euclid's first postulate, Euclid assumes without apparent justification that the two circles constructed in the course of the proof must intersect. However, in CDEG this issue is solved by the underlying diagrammatic machinery. After we add the second circle, the single diagram shown in Figure 4 is produced, and in it, the two circles do, indeed, intersect, giving us two intersection points (dots 11 and 12) that we can use in the rest of the derivation. Similarly, most of the other gaps in Euclid's reasoning are taken care of by the diagrammatic machinery. Thus, we can view them as being part of an unarticulated diagrammatic process rather than as flaws in Euclid's arguments.

One particular way that CDEG differs from Euclid's *Elements* is in its adoption of the side-angle-side and side-side-side triangle congruence criteria as primitive rules rather than as propositions to be derived. Euclid derives the rules using transformations and the principle of superposition. While it is possible to formally mimic the derivations using transformation rules, as is done in FG [7, Section 3.3], for the computer system it was simpler to just adopt the triangle congruence rules directly.

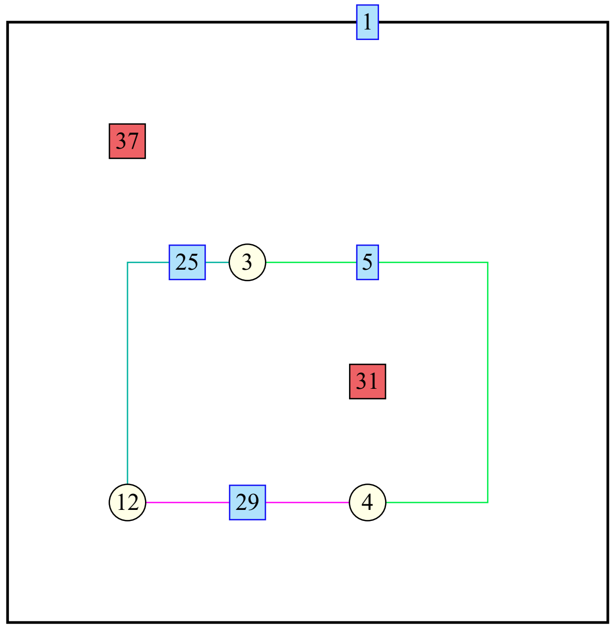

Figure 5. A CDEG diagram showing the triangle obtained in the final step of the proof of Euclid's first proposition.

Because it duplicates all of Euclid's proof methods, CDEG should be theoretically able to prove versions of all of Euclid's propositions from the first four books of the *Elements*, which is the part that deals purely with planar geometry.

However, anyone who tries to actually use CDEG to duplicate the books will quickly realize that in practice it will be difficult to use CDEG to duplicate all of Euclid's proofs. One issue is that as the diagrams become more complicated, the amount of computer time required for one step in the proof can grow exponentially. In particular, the commands that draw circles and lines can take an amount of time that is exponential in the number of objects in a diagram. Thus, some computations may take impractically long. Furthermore, the result of applying a construction rule to a single primitive diagram is a diagram array containing all of the topologically distinct possible diagrams that could occur when the newly constructed object is added, and the number of new diagrams in such an array can also be exponential in the number of objects in the original diagram. So the number of cases that need to be considered can also grow very quickly. See [7, Section 4.1] for more discussion of this phenomenon.

A related issue is that of unsatisfiable diagrams. Ideally, we would want CDEG 's construction rules to return as few diagrams as possible, in order to minimize the number of cases that need to be considered. Unfortunately, the rules sometimes produce diagrams that represent arrangements of circles and lines that cannot be physically realized with actual straight lines and circles. It turns out that this is unavoidable in practice. In [6], it is shown that the question of determining which diagrams that result from applying a construction rule are physically realizable is at least NP-hard, which means that it is not practically computable in a reasonable amount of time.

Another issue that exacerbates the problem of an exploding number of cases is the lack of lemma incorporation in CDEG. Lemma incorporation refers to the use of previously derived propositions and lemmas in proving new theorems. Most proof systems include the ability to do this, but CDEG does not, because it is technically harder to implement lemma incorporation in a diagrammatic setting. Of course, previously derived propositions and lemmas can always be rederived in the course of a proof. However, the additional objects in the diagram in the course of a later proof normally necessitate considering even more cases. Thus, the lack of lemma incorporation can lead to a huge blowup in the length of a given proof. For a more extensive discussion of lemma incorporation, see [7, Section 4.1]. Lemma incorporation will hopefully be included in some future version of CDEG.

In the meantime, one way to use CDEG is to try to duplicate one of Euclid's proofs, but to only complete the proof for one branch of the many possible cases that arise. This allows the user to avoid tediously looking at many different cases that are all essentially similar, while still seeing the essence of the proof. This might be one way to try to use CDEG with students. In this case, we haven't actually proven the theorem in general, but have rather given an illustration of how we would prove it. Interestingly, this actually mirrors Euclid's normal practice. In the *Elements*, he normally only gives a proof for one single case, the hardest one, leaving the other cases for the reader to fill in. For discussion of this practice, see Heath's commentary on Proposition 2 in his edition of the *Elements* [1]. This is one of the aspects of Euclid's practice that probably contributed to the twentieth-century impression that his proofs were inherently informal, since without a formal system like the one provided by CDEG, it wasn't clear that it was possible to give a principled account of how many cases really had to be considered.

5 Why a Computer System?

CDEG is essentially a computer implementation of the formal system FG described in [7]. Actually implementing the formal system on a computer was a highly non-trivial matter that took several years' work. Why would we want to implement an existing formal system on a computer?

The first reason that we might want a computer implementation is to demonstrate that the system really is completely formal: that the diagrams that are being manipulated are, indeed, completely specified as formal objects, and that the rules of the system are completely specified on them. With traditional, sentential formal systems, we do this by writing our axioms in a formal language, and then carefully writing rules of inference as typographical manipulations of its sentences. However, when our formal objects are diagrams, it is difficult to achieve this level of specificity without a computer implementation. Diagrams are complicated formal objects, and we have strong informal intuitions about how they should work that may cloud our ability to judge if our rules have been completely formally specified.

Furthermore, even if our rules are completely formally specified, without a computer implementation, it would be quite difficult to play with the formal system to see what derivations are like, and to make sure that they really work the way that we think they will. This is particularly true in geometry, where constructions can lead to case branching, with a large number of cases that are virtually impossible to keep track of without using a computer. Thus, we may not be able to prove everything we think we can.

This worry is not just academic. Several other diagrammatic formal systems have been proposed by other researchers and have appeared in print but have later turned out to have rules that were ill-defined or unsound, in the technical sense that they could derive conclusions that didn't logically follow from their hypotheses. For example, Isabel Luengo's formal system DS1, described in a chapter of the book *Logical Reasoning with Diagrams* [4], is unsound, as explained in [7, Appendix C]; likewise, John Mumma's Eu, described in [11], [12], and elsewhere, is also unsound, as described in [10]. Because neither system was implemented as a computer system, the way they worked was not completely specified concretely, and so they were not fully understood. This helps to explain why neither the designers of the systems nor the many referees who looked at their work before it was published noticed their significant problems. Thus, we should approach a proposed diagrammatic formal system with a certain amount of healthy skepticism. A working computer system is one way to allay some of the skepticism.

Secondly, a computer system is the only way to make a formal system widely available. Many potential users will not be able to make sense out of a formal system that is just specified mathematically, but will be able to try out a computer implementation.

The third reason for a computer implementation is to be able explore exactly what the formal system is able to prove. Above, I claim that CDEG should be able to duplicate the first four books of Euclid's *Elements*. The only way to verify this claim is to systematically go through each of Euclid's proofs, and to see how to duplicate it within CDEG. To date, I have done this with many proofs from Euclid's Book I, but have not yet gone systematically through all of Euclid's proofs. Doing so is a possible future project that would be essentially impossible without the computer implementation.

One small example of the way that the computer implementation sheds light on the formal system has to do with the way that CDEG handles subtraction of segments. Euclid's common notion 3 states that "If equals be subtracted from equals, the remainders are equal." I originally thought that this rule should be derivable from the rule for addition of segments, and so it wasn't included in CDEG version 1.0. It was only when I actually tried to do the derivation within CDEG that I discovered that the proof I had in mind wouldn't work because it relied on Euclid's Proposition 2, and one case that arose in the proof of Proposition 2 could only be proven using common notion 3. Thus, segment subtraction is now included in CDEG as a primitive rule. I don't think I would have found this mistake without using the computer system itself, and it was a significant omission, since without this change, even Euclid's Proposition 2 would not have been derivable! I thought that I had found something that Euclid missed, but it turned out that, as usual, he was one step ahead of me.

I hope that readers of this chapter will be interested in trying CDEG for themselves. It can be downloaded from www.unco.edu/NHS/mathsci/facstaff/Miller/personal/CDEG/. However, the version of CDEG that is now available is a beta version and most likely still contains bugs. If you try out CDEG and discover any bugs, please let me know by sending me an email. I can be reached at nat@alumni.princeton.edu.

6 Philosophical Implications for Teaching

I have not used CDEG with students in any geometry class that I have taught. However, I think that the philosophical issues it raises are directly relevant to how we teach geometry, and that they arise naturally in many geometry courses.

Geometry classes tend to be a venue in which ideas about proof and proving come up in our curricula. So they are a good place to discuss philosophical questions about proof: What does it mean to prove something? Why do we prove things? What constitutes a convincing argument in mathematics, and how are they different from convincing arguments in other disciplines? Where do definitions come from, and what is their role in proof? How do we pick what assumptions we are going to make, and then how do we use them? The geometry classes that I teach are generally taught in an inquiry-based format, and these sorts of questions arise naturally all the time. (For detailed descriptions of some of my inquiry-based geometry classes, see [8] and [9].) They are questions whose answers mathematicians tend to take for granted, but students who are asked to write definitions and proofs without a specific template for the first time invariably struggle with them.

As, indeed, they should. The answers to the questions are by no means as simple as they might seem, and their difficulties are reflected in many geometry curricula. For two thousand years, Euclid's *Elements* was considered the gold standard of logical reasoning, and was itself the basis for most geometry curricula. With the rise of formalism in the philosophy of mathematics, however, the new view was that Euclid's reasoning was not rigorous, and that a lot of the lack of rigor came from Euclid's use of diagrams. This gave rise to formal sentential axiomatizations of geometry, like that found in [3]. However, unlike Euclid's *Elements*, the formalizations of geometry don't reflect informal practice in giving proofs in geometry very well, and they generally don't work well as a starting point for students studying geometry for the first time. So we get geometry textbooks—high school geometry textbooks, especially—that don't really want to follow either kind of approach, and end up muddled somewhere in the middle, trying to follow an axiomatic system more complicated than Euclid's, adding new axioms whenever a proof would be too hard, but still don't make clear precisely what role diagrams play in their proofs.

CDEG and its underlying formal system are intended to show geometry teachers that diagrammatic proofs in the style of Euclid can be made perfectly rigorous, so that teachers don't need to shy away from these kinds of proofs. For the most part, they probably won't show CDEG to their students, any more than they would show them a fully formal sentential presentation of geometry. Instead, I would hope that teachers would share with their students informal geometric proofs using diagrams in the style of Euclid, would ask their students to be able to produce their own proofs in this style, and that, along the way, they would facilitate wide-ranging class discussions of the basic philosophical questions of what makes these proofs convincing or unconvincing.

Bibliography

[1] Euclid, *The Elements*, translated with introduction and commentary by Thomas L. Heath, 2nd ed., Dover, New York, 1956.

[2] Forder, Henry George, *The Foundations of Euclidean Geometry*, Cambridge University Press, Cambridge, 1927.

[3] Hilbert, David, *Foundations of Geometry*, translated by Leo Unger, 4th ed., Open Court Publishing Co., La Salle, 1971.

[4] Luengo, Isabel, A diagrammatic subsystem of Hilbert's geometry, *Logical Reasoning with Diagrams*, G. Allwein and J. Barwise, eds., Oxford University Press, New York, 1996.

[5] Miller, Nathaniel, *CDEG User's Manual*, available from www.unco.edu/NHS/mathsci/facstaff/Miller/personal/CDEG/.

[6] Miller, Nathaniel, Computational complexity of diagram satisfaction in Euclidean geometry, *Journal of Complexity*, 22(2) (2006), 250–274.

[7] Miller, Nathaniel, *Euclid and his Twentieth Century Rivals*, CSLI Press, Stanford, 2007.

[8] Miller, Nathaniel, Modern geometry I, *Journal of Inquiry-Based Learning in Mathematics* 17 (2010), 1–104.

[9] Miller, Nathaniel, Modern geometry II, *Journal of Inquiry-Based Learning in Mathematics* 19 (2010), 1–46.

[10] Miller, Nathaniel, On the inconsistency of Mumma's Eu, *Notre Dame Journal of Formal Logic* 53 (1) (2012), 27–52.

[11] Mumma, John, Ensuring generality in Euclid's diagrammatic arguments, pp. 222–235 in G. Stapleton, J. Howse, and J. Lee, eds., *Diagrammatic Representation and Inference: 5th International Conference, Diagrams 2008*, Springer Lecture Notes in Computer Science 5223, 2008.

[12] Mumma, John, Proofs, pictures, and Euclid, *Synthese* 175 (2) (2010), 255–287.

[13] Tarski, Alfred, *A Decision Method for Elementary Algebra and Geometry*, University of California Press, Berkeley, 1951.