A DIAGRAMMATIC FORMAL SYSTEM FOR EUCLIDEAN GEOMETRY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Nathaniel Gregory Miller

May 2001

A DIAGRAMMATIC FORMAL SYSTEM FOR EUCLIDEAN GEOMETRY

Nathaniel Gregory Miller, Ph.D.

Cornell University 2001

It has long been commonly assumed that geometric diagrams can only be used as aids to human intuition and cannot be used in rigorous proofs of theorems of Euclidean geometry. This work gives a formal system **FG** whose basic syntactic objects are geometric diagrams and which is strong enough to formalize most if not all of what is contained in the first several books of Euclid's *Elements*. This formal system is much more natural than other formalizations of geometry have been. Most correct informal geometric proofs using diagrams can be translated fairly easily into this system, and formal proofs in this system are not significantly harder to understand than the corresponding informal proofs. It has also been adapted into a computer system called **CDEG** (Computerized Diagrammatic Euclidean Geometry) for giving formal geometric proofs using diagrams. The formal system **FG** is used here to prove meta-mathematical and complexity theoretic results about the logical structure of Euclidean geometry and the uses of diagrams in geometry.

Biographical Sketch

Nathaniel Miller was born on September 5, 1972 in Berkeley, California, and grew up in Guilford, CT, Evanston, IL, and Newtown, CT. After graduating from Newtown High School in 1990, he attended Princeton University and graduated *cum laude* in mathematics in 1994. He then went on to study mathematics and computer science at Cornell University, receiving an M.S. in computer science in August of 1999 and a Ph.D. in mathematics in May of 2001. When not doing mathematics, he plays the cello, teaches swing dancing, and gardens.

"We do not listen with the best regard to the verses of a man who is only a poet, nor to his problems if he is only an algebraist; but if a man is at once acquainted with the geometric foundation of things and with their festal splendor, his poetry is exact and his arithmetic musical."

- Ralph Waldo Emerson, *Society and Solitude* (1876)

Acknowledgements

First and foremost, I would like to thank my advisor David W. Henderson for his unfailing help and support of a thesis topic that was slightly off of the beaten path. I'd also like to thank the other members of my committee, Richard Shore and Dexter Kozen, for their valuable help and suggestions. I am further indebted to the many other people with whom I have discussed this work and who have offered helpful comments, especially Yaron Minsky-Primus, who was always happy to discuss diagrams in the middle of raquetball games; Avery Solomon; my father, Douglas Miller; and Zenon Kulpa, who asked the questions whose answers led to Section 4.3. Finally, thanks to all of my friends and family, and especially to my parents, Douglas and Eleanor Miller: words cannot express the thanks I feel for all of the love and support that you have given me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

To begin, consider Euclid's first proposition, which says that an equilateral triangle can be constructed on any given base. While Euclid wrote his proof in Greek with a single diagram, the proof that he gave is essentially diagrammatic, and is shown in Figure 1.1. Diagrammatic proofs like this are common in informal treatments of geometry, and the diagrams in Figure 1.1 follow standard conventions: points, lines, and circles in a Euclidean plane are represented by drawings of dots and different kinds of line segments, which do not have to really be straight, and line segments and angles can be marked with different numbers of slash marks to indicate that they are congruent to one another. In this case, the dotted segments in these diagrams are supposed to represent circles, while the solid segments represent pieces of straight lines.

It has often been asserted that proofs like this, which make crucial use of diagrams, are inherently informal. The comments made by Henry Forder in *The Foundations of Euclidean Geometry* are typical: "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

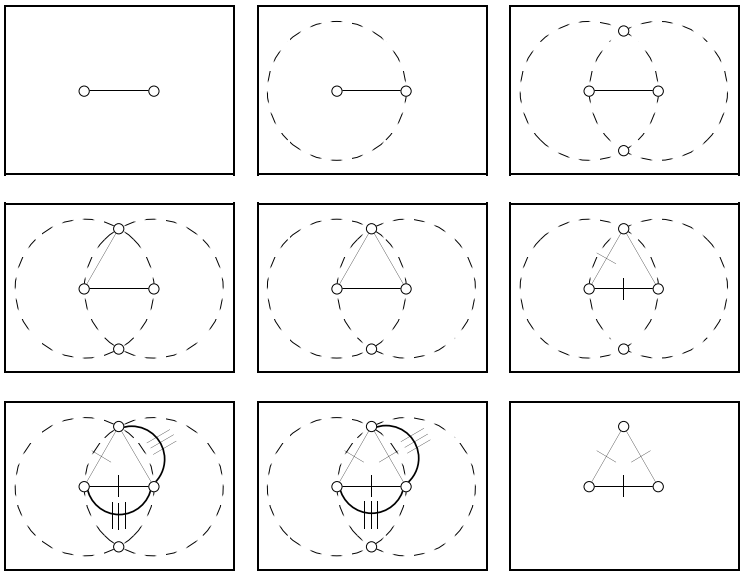


Figure 1.1: Euclid's first proposition.

no part" [8, p.42]. Traditional formal proof systems are sentential—that is, they are made up of a sequence of sentences. Usually, however, these formal sentential proofs are very different from the informal diagrammatic proofs. A natural question, then, is whether or not diagrammatic proofs like the one in Figure 1.1 can be formalized in a way that preserves their inherently diagrammatic nature.

The answer to this question is that they can. In fact, the derivation contained in Figure 1.1 is itself a formal derivation in a formal system called **FG**, which will be defined in the following sections, and which has also been implemented in the computer system **CDEG** (Computerized Diagrammatic Euclidean Geometry). These systems are based a precisely defined syntax and semantics of Euclidean diagrams. We are going to define a diagram to be a particular type of geometric object satisfying certain conditions; this is the syntax of our system. We will also give a formal definition of which arrangements of lines, points, and circles in the plane are represented by a given diagram; this is the semantics. Finally,

we will give precise rules for manipulating the diagrams—rules of construction, transformation, and inference.

In order to work with our diagrams, we will have to decide which of their features are meaningful, and which are not. A crucial idea will be that all of the meaningful information given by a diagram is contained in its topology, in the general arrangement of its points and lines in the plane. Another way of saying this is that if one diagram can be transformed into another by stretching, then the two diagrams are essentially the same. This is typical of diagrammatic reasoning in general.

1.1 A Short History of Diagrams, Logic, and Geometry

The use of diagrams in geometry has a long history.[1] In fact, geometric diagrams are found among some of the oldest preserved examples of written mathematics, such as the Babylonian clay tablets found by archeological digs of ancient Mesopotamian city mounds at the end of the nineteenth century. These tablets, most of which are believed to date from around 1700 B.C., contain some fairly

[1]The history given here is not meant to be exhaustive, and is drawn from many sources. For more information, see [1] and [20] for discussion of Babylonian mathematics; [12] for discussion of the mathematics of other ancient cultures and the development of algebra in Arabia; [13] for discussion of the history of Greece and Greek mathematics and a thumbnail sketch of the history of mathematics up to the twentieth century; [23] and [3] for biographies of many mathematicians, including Archimedes, Descartes, Fermat, Leibniz, Gauss, Lobachevsky, and Boole; [4], [14], and [5] for the history of the discovery of non-Euclidean geometry, the arithmetization of mathematics, and the formalization of logic; [9] and [22] for the history of logic diagrams; and [2] for the history of recent developments in the theory of reasoning with diagrams.

Figure 1.2: A Babylonian Tablet dating from around 1700 B.C.

sophisticated arithmetical computations, and a number of them include diagrams. For example, the old-Babylonian tablet shown in Figure 1.2 (reproduced from [1]) shows the computation of the length of the diagonal of a square with sides of length 30, using a very good approximation of the square root of two. Geometric diagrams are also found in ancient Egyptian, Chinese, and Indian mathematical works.

It is with the Greeks, though, that mathematics really came into its own, and first and foremost among the Greek mathematical texts that have come down to us is Euclid's *Elements*. In fact, Euclid's *Elements* was such a seminal work that it has almost entirely eclipsed older Greek mathematical works—even though it wasn't written until around 300 B.C., long after the crowning achievements of the Greeks

in art and literature, and thirty years after Alexander The Great had incorporated Greece into his empire centered in Alexandria, in Egypt. (In fact, Euclid himself lived and worked in Alexandria.) Thus, despite the fact that Euclid's *Elements* was part of a rich Greek mathematical tradition dating back to the beginning of the sixth century B.C., almost no earlier Greek mathematical works have come down to us in their entirety. This seems to be largely because *The Elements* succeeded in incorporating the majority of the preexisting mathematics into its logical development. *The Elements* has been a preeminent work in mathematics since the time it was written for a number of reasons, but chief among them is the fact that Euclid set down his assumptions in advance and tried to give explanations for why geometrical facts were true on the basis of his assumptions and previously shown facts. Thus, it is with Greek mathematics that we first encounter the notions of mathematical proof and the logical development of a subject. We also find a precursor of formal symbolic logic in the Greek theory of syllogistic reasoning, codified in Aristotle's *Prior Analytics*, written about fifty years before Euclid's *Elements*. Euclid's main concern in *The Elements* was Euclidean geometry, and, as we have already seen, his proofs of geometric facts rely heavily on diagrams. In fact, his first three postulates specify diagrammatic actions that can be performed in the course of a proof, although they are often translated in ways that obscure this fact: for example, his first postulate allows you "To draw a straight line from any point to any point." (See Appendix A for Sir Thomas Heath's literal translations of Euclid's Postulates.) Thus, these postulates, as originally stated, are hard to understand in any way that isn't essentially diagrammatic.

The rise of the Roman Empire around 200 B.C. more or less eclipsed Greek culture, and in particular it eclipsed the Greek mathematical culture with its em-

phasis on proof. According to a famous story related by Plutarch in [19], the Greek mathematician Archimedes was killed during the Roman conquest of the Greek city of Syracuse when he refused to come with an invading soldier until he was done studying a geometric diagram drawn in the sand. The British logician Alfred North Whitehead, one of the authors of the *Principia Mathematica*, thus remarked on the difference between the Greek and Roman cultures, "No Roman ever lost his life because he was absorbed in contemplation of a mathematical diagram" [25]. As a result, we find fewer new developments in geometry or in logic for quite a long time after 200 B.C. Still, the *Elements* were always studied and carefully preserved, first by the Greeks and Romans, and then, after the destruction of Alexandria in 640 A.D., by Arabs in Arabic translations. The most important Arabic contribution to mathematics was probably their development of the subject of algebra. In fact, the word *algebra* comes from the arabic title of a book on the subject written by the Arabic mathematician Muhammad ibn Musa al-Khwarizmi in the ninth century A.D. In this work, al-Khwarizmi gives numerical methods for solving several different types of equations, followed by geometric proofs that these methods work. Thus, Arabic mathematics combined the subjects of algebra and geometry, using the Greek theory of geometry as the foundation for their developing theory of algebra.

It is not until the European Renaissance that we find steps away from the use of geometry as the foundation of mathematics. The first step came with the invention of analytic geometry in the 1630s by Pierre de Fermat and René Descartes. These men realized that it was possible to use algebra as a tool for studying geometry, and in doing so, they took the first steps towards a mathematics with arithmetic rather than geometry at its core. In Greek mathematics, geometry was viewed as

the foundation for all other branches of mathematics, and so the Greek theories of arithmetic and algebra were based on their theory of geometry. The development of analytic geometry allowed mathematicians to instead base the theory of geometry on the theory of numbers, and thus it set mathematics on the path to arithmetization. The development of integral and differential calculus by Isaac Newton and Gottfried Leibniz independently in the 1660s and 1670s represented another big step in this direction, calculus being a tremendously powerful tool for studying geometric curves by using methods that are essentially arithmetical. The logical conclusion of this path was the definition of the Real numbers in terms of the rationals, themselves defined in terms of the natural numbers, by Dedekind, Cantor, and others around 1870. With this development, geometry, with the Real numbers at its core, could be seen as a mere extension of arithmetic. Thus, while Plato is quoted by Plutarch as having said that "God ever geometrizes" [18], by the early 1800s this had become Jacobi's "God ever arithmetizes" [3].

Another factor that influenced the shift from geometry to arithmetic as the foundation of mathematics was the discovery of the consistency of non-Euclidean geometries in the 1820s by Gauss, Lobachevsky, and Bolyai. From the time of Euclid, students of *The Elements* had been unsatisfied with Euclid's fifth postulate. They felt that it was too inelegant and complex to be a postulate, and that it should therefore be possible to prove from the remaining postulates. Many proofs were proposed and even published, but each turned out to have made some additional assumptions. Finally, two thousand years after Euclid wrote, Gauss, Lobachevsky, and Bolyai each realized that there are consistent geometries in which the first four postulates hold, but the fifth does not. Thus, the fifth postulate cannot not be proven from the first four, because if it could, it would have to be true whenever

they were. The discovery of these other geometries greatly weakened Euclidean geometry's claim to be the basis for all other mathematics. Before their discovery, it was thought that Euclidean geometry was just a codification of the laws of the natural world, and so it was a natural foundation on which to base the rest of mathematics. After people realized that other geometries were possible and that Euclidean geometry wasn't necessarily the true geometry of the physical world, it no longer had a claim to greater certainty than any other mathematical theory.

The transition from mathematics with geometry at its core to mathematics with arithmetic at its core had a profound influence on the way in which people viewed geometric diagrams. When geometric proofs were seen as the foundation of mathematics, the geometric diagrams used in those proofs had an important role to play. Once geometry had come to be seen as an extension of arithmetic, however, geometric diagrams could be viewed as merely being a way of trying to visualize underlying sets of Real numbers. It was in this context that it became possible to view diagrams as being "theoretically unnecessary," mere "props to human infirmity."

As the rest of mathematics became arithmetized, so too did logic. The first steps in arithmetizing logic were taken Leibniz in the 1670s and 1680s, when he tried to develop a kind of algebraic system capturing Aristotle's rules for working with syllogisms. Leibniz's objective of finding a way of reducing syllogistic logic to algebra was finally realized two hundred years later by George Boole in 1847. Over the next forty years various other people extended Boole's logical algebra in order to make it applicable to more of mathematics. Notable among them was the American Charles Sanders Pierce, who modified Boole's algebra to incorporate the use of relations and quantifiers. Finally, in 1879, Gottlob Frege published a

book containing a logical system roughly equivalent to modern first-order predicate logic.

Interestingly, at the same time that these mathematicians were looking at ways to arithmetize logic, others were looking at ways to diagramize logic. The first method for using geometric diagrams of circles to solve syllogistic reasoning problems was given by Euler in 1761. His method of using circles to represent classes of objects was updated and improved by John Venn's introduction in 1880 of what are now known as Venn Diagrams. These were in turn updated and improved by C. S. Pierce's introduction in 1897 of what he called Existential Graphs. (This is the same C. S. Pierce who had introduced quantifiers into Boole's algebra.) These Existential Graphs are notable not only for their expressive power, but also for the fact that Pierce gave a collection of explicit rules for manipulating them. Also worth mentioning here is C. L. Dodgeson (Lewis Carroll), who in 1886 published a book called *The Game of Logic*, in which he proposed his own system of logic diagrams, equal in expressive power to those of Venn.

In the last decade of the nineteenth century, formal logic was well enough developed that careful axiomatizations of mathematical subjects could be given in formal languages. Around 1890, Giuseppe Peano published axiom systems for a number of mathematical subjects in a formal "universal" language that was based on the formalisms developed by Boole and Pierce. Among these were the axiomatization of arithmetic that now bears his name and an axiomatization of Euclidean geometry. Peano's axiomatization of geometry, along with several others, was eclipsed by David Hilbert's *Foundations of Geometry*, the first version of which was written in 1899. By this point in time, Euclid's axiomatization and proofs had come to be seen as being insufficiently rigorous for a number of rea-

sons, among them his use of diagrams. For example, the proof of Euclid's first proposition, discussed in the previous section, requires finding a point where the two circles intersect. Euclid seems to assume that this is always possible on the basis of the diagram, but none of his postulates appear to require the circles to intersect. Hilbert's axiomatization was meant to make it possible to eliminate all such unstated assumptions. In fact, Hilbert showed that there is a unique geometry that satisfies his axioms, so that any fact that is true in that geometry is a logical consequence of his axioms. However, a proof from Hilbert's axioms may not look anything like Euclid's proof of the same fact. For example, Hilbert's axioms do not mention circles, so any proof of Euclid's first proposition will have to be very different from Euclid's proof.

Hilbert's axiomatization of geometry was part of a larger movement to try to put mathematics on the firmest possible foundation by developing all of mathematics carefully from a small number of given axioms and rules of inference. This movement found its greatest expression in the *Principia Mathematica* of Bertrand Russell and Alfred North Whitehead, written between 1910 and 1913, which succeeded in developing a huge portion of mathematics from extremely simple axioms about set theory. However, it turned out that the goal of finding a finite set of axioms from which all of mathematics could be derived was impossible to achieve. In 1930, Kurt Gödel proved his First Incompleteness Theorem, which says approximately that no finite set of axioms is strong enough to prove all of the true facts about the natural numbers. The proof of this theorem involved translating logical statements into numbers and proofs into arithmetical operations on those numbers, and so it can be seen as having completed the arithmetization of logic. In any case, after Gödel's theorem was proven, logicians had to content themselves

with more modest goals. In general, they still tried to reason from a small number of carefully specified axioms and rules of inference, because then if the axioms were true in a given domain and the rules of inference were sound, then any theorems proven would be correct.

It was not until recently that modern logic was applied to the study of reasoning that made use of diagrams. In the late 1980s, Jon Barwise and John Etchemendy developed a series of computer programs that were meant to help students visualize the concepts of formal logic. These programs, *Turing's World*, *Tarski's World*, and *Hyperproof*, included diagrams of a blocks world, and they inspired Barwise and Etchemendy to look more closely at forms of reasoning that used diagrams. In 1989, they published an article, "Visual Information and Valid Reasoning," reprinted in [2], that asserted that diagrammatic reasoning could be made as rigorous as traditional sentential reasoning and challenged logicians to look at diagrammatic reasoning more seriously.

Sun-Joo Shin, a student of theirs, began looking at the work that had been done with logic diagrams a hundred years before. As we have seen, the development of systems of logic diagrams roughly mirrored the development of formal algebraic logical systems up to the end of the nineteenth century, but at that point they were for the most part abandoned as the theory of formal systems continued to develop in the twentieth century. Shin finally brought twentieth century developments in logic to bear on the theory of logic diagrams. She clarified Peirce's system of Existential Graphs, and showed that the system thus obtained was both sound and complete—that the diagrams that could be derived from a given diagram system were exactly those that were its logical consequences. She also extended this system to include a more general form of disjunction and showed that the resulting diagrams had

the same expressive power as the monadic first-order predicate calculus.

The first person to try to formalize the uses of diagrams in Euclidean geometry was Isabel Luengo, also a student of Jon Barwise. In her thesis [16], finished in 1995, she introduced a formal system for manipulating geometric diagrams by means of formal construction and inference rules, and introduced the definition of "geometric consequence," which extends the notion of logical consequence to domains that include construction rules. However, her system does not incorporate the crucial idea that two diagrams should considered equivalent if and only if they are topologically equivalent, and as a result her system is unsound. For a detailed discussion of her formal system and an explanation of why it is unsound, see Appendix B.

Chapter 2

Syntax and Semantics of Diagrams

2.1 Basic Syntax of Euclidean Diagrams

If we want to discuss the role of diagrams in geometry, we must first say what is meant by the term diagram in this context. Figure 2.1 shows two examples of the sort of diagrams we want to consider. They contain dots and edges representing points, straight lines and circles in the plane, but note that a diagram may not

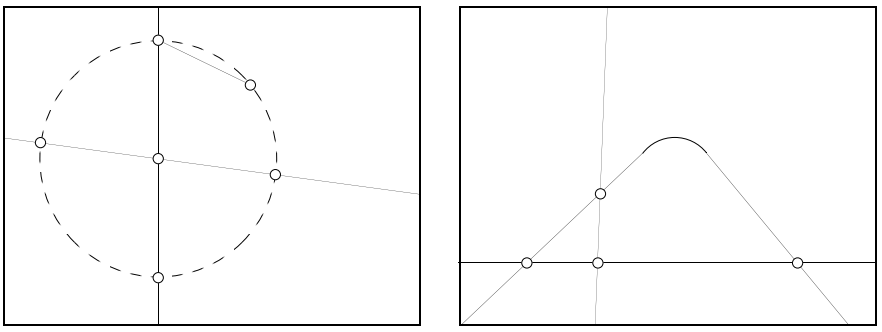


Figure 2.1: Two primitive diagrams.

look exactly like the configuration of lines and circles that it represents; in fact, it may represent an impossible configuration, like the second diagram in Figure 2.1.

Formally, we define a diagram as follows:

Definition 2.1.1. *A **primitive Euclidean diagram** D is a geometric object in the plane that consists of*

1. *a rectangular box drawn in the plane, called a **frame**;*
2. *a finite set $DOTS(D)$ of **dots** which lie inside the area enclosed by the frame, but cannot lie directly on the frame;*
3. *two finite sets $SOLID(D)$ and $DOTTED(D)$ of **solid and dotted line segments** which connect the dots to one another and/or the frame, and such that each line segment*
   (a) *lies entirely inside the frame,*
   (b) *is made up of a finite number of connected pieces that are either straight lines or else arcs of circles, which intersect each other only at their endpoints, and such that each of these pieces intersects at most one other piece at each of its endpoints,*
   (c) *does not intersect any other segment, any dot, the frame, or itself except at its endpoints, and*
   (d) *either forms a single closed loop, or else has two endpoints, each of which lies either on the frame or else on one of the dots;*
4. *a set $SL(D)$ of subsets of $SOLID(D)$, such that each segment in $SOLID(D)$ lies in exactly one of the subsets; and*

5. *a set CIRC(D) of ordered pairs, such that the first element of the pair is an element of DOTS(D) and the second element of the pair is a subset of DOTTED(D), and such that each dotted segment in DOTTED(D) lies in exactly one of these subsets.*

The intent here is that the primitive diagram represents a Euclidean plane containing points, straight lines and line segments, and circles. The dots represent points, the solid line segments in SOLID(D) represent straight line segments, and the dotted line segments in DOTTED(D) represent parts of circles. SL(D) tells us which solid line segments are supposed to represent parts of the same straight line, and CIRC(D) tells us which dotted line segments are supposed to represent parts of the same circle, and where the center of the circle is. (This comment is intended only to motivate the definitions being made now, and will be explained more carefully later on.) The sets in SL(D) are called ***diagrammatic lines***, or ***dlines*** for short, and the pairs in CIRC(D) are called ***diagrammatic circles*** or ***dcircles***. Elements of dlines are said to lie on the dline, and likewise, elements of the second component of a dcircle are said to lie on the dcircle; the first component of a dcircle is called the ***center*** of the dcircle. Each solid line segment must lie on exactly one dline, and each dotted line segment must lie on exactly one dcircle. A dline or dcircle is said to intersect a given dot (or the frame) n times if it has n component segments with endpoints on that dot (or on the frame), counting a segment twice if both of its endpoints lie on the frame or on the same dot. Notice that it follows from the preceding definition that dlines and dcircles can only intersect other dlines and dcircles at dots (or on the frame, but this will eventually be disallowed).

We are now going to put some constraints on these diagrams to try to make

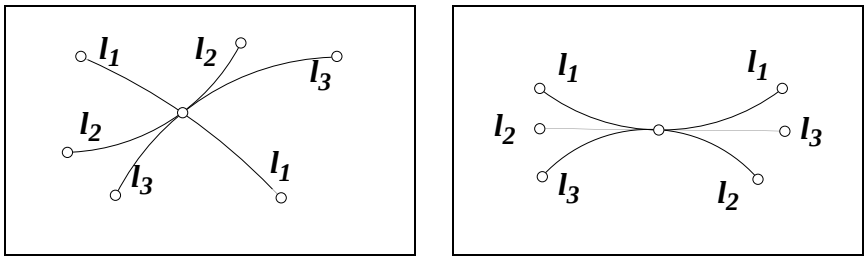


Figure 2.2: Examples of diagrammatic tangency.

sure that they look as much as possible like real configurations of points, lines, and circles in the plane. To begin, we would like to ensure that the dlines and dcircles come together at a dot in a way that mimics the way that real lines and circles could meet at a point. To this end, we first define the notion of diagrammatic tangency:

Definition 2.1.2. *If each of e and f is a dcircle or dline that intersects the dot d exactly twice, then e and f are defined to be* ***diagrammatically tangent*** *(or* ***dtangent****) at d if they do not cross each other at d.*

This means that if s_{e1} and s_{e2} are the segments that are part of e which intersect d and, likewise, s_{f1} and s_{f2} are the segments from f that intersect d, then if s_{f1} occurs between s_{e1} and s_{e2} when the segments that intersect d are listed in clockwise order, then s_{f2} also occurs between s_{e1} and s_{e2} in this list. For example, in the first diagram in Figure 2.2, l_2 and l_3 are diagrammatically tangent to one another, while l_1 and l_2 are not. We are going to require the dcircles and dlines to intersect at d in such a way that the dtangency relation is transitive—in other words, so that if e and f intersect at d without crossing and f and g intersect at d without crossing, then e and g don't cross either (although they might both lie on the same side of f). This says that the situation in the second diagram in Figure 2.2, in which l_2 crosses l_3 but not l_1, cannot occur. Since dtangency is automatically symmetric

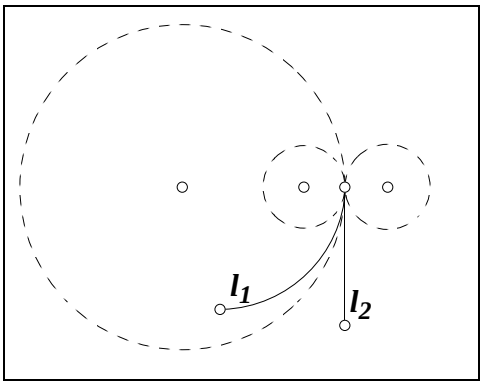


Figure 2.3: A non-viable primitive diagram

and reflexive, this makes it into an equivalence relation. We can then extend the notion of diagrammatic tangency to dlines that only intersect d once by specifying that if e is such a dline, and e intersects d directly between two members of the same dtangency equivalence class, then e is dtangent to all of the members of that equivalence class. Thus, l_1 and l_2 in Figure 2.3 are dtangent to one another under this definition. A dline that only intersects d once is said to ***end*** at d.

We can now define a dot d to be viable as follows:

Definition 2.1.3. *A dot is* ***viable*** *if*

1. *any dcircle that intersects the dot intersects it exactly twice;*
2. *any dline that intersects the dot intersects it at most twice;*
3. *the dcircles and dlines that intersect d do so in such a way so as to make the dtangency relation transitive; and*
4. *no two dlines are dtangent at d.*

A primitive diagram D is viable if every dot in D is viable.

It follows from the preceding that if one member of a dtangency equivalence class crosses f at d, then all of the other members of the dtangency class also cross

f at d; otherwise, some other member of the class would be dtangent to f at d, forcing them all to be dtangent to f at d. It also follows that each dtangency equivalence class can contain at most one dline, which may or may not end at d, since dlines are not allowed to be dtangent to other dlines. Notice that viability is a local property of diagrams—it says that the diagram is locally well-behaved at each dot. The two diagrams in Figure 2.1 are viable, while the three diagrams in Figures 2.2 and 2.3 are not. Note that our definition of viability allows viable diagrams to contain segments of lines, but not arcs of circles.

Next, we would like to ensure that the dlines and dcircles of our diagrams behave like real lines and circles. We do this with the following definition.

Definition 2.1.4. *A primitive diagram D is* ***well-formed*** *if it is viable and*

1. *no dotted line segment in D intersects the frame;*
2. *no two line segments intersect the frame at the same point;*
3. *every dline and dcircle in D is connected—that is, given any two dots that a dline or dcircle P intersects, there is a path from one to the other along segments in P;*
4. *every dline has exactly two ends, where the* ***ends*** *of a dline are defined to be the points where it intersects the frame or a dot which it only intersects once; and*
5. *every dcircle in D is made up of segments that form a single closed loop such that the center of the dcircle lies inside that loop.*

We call a dline that intersects the frame twice a ***proper dline***; one that intersects the frame once a ***d-ray***; and one that doesn't intersect the frame at all a ***dseg***

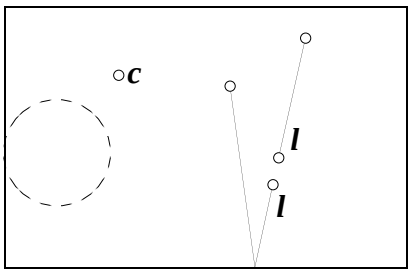


Figure 2.4: A viable diagram that isn't well-formed.

(not to be confused with the solid line segments that make it up). A well-formed primitive diagram is also called a ***wfpd***. Figure 2.4 shows a viable diagram that isn't well-formed and violates each of the four clauses of the definition. Both of the diagrams in Figure 2.1, however, are well-formed.

It should be noted that in principle, the diagrams drawn here should also tell you which segments make up each dline and dcircle. In the case of the first diagram in Figure 2.1, if we know that there are three dlines and one dcircle in this diagram, there are three different ways that the segments can be assigned to dlines and dcircles that make this a wfpd, as the reader should be able to check. Notice that if we are told that there are no dtangencies in a wfpd in which every dline is proper, then there is only one way to assign segments to dlines and dcircles that is consistent with the diagram being well-formed, because you can determine which segments belong to the same dline or dcircle at a given dot by looking at the clockwise order in which the segments intersect the dot. In practice, it is usually clear which segments are intended to belong to the same dline or dcircle, and we won't indicate this unless it is unclear. We could also prove a theorem showing that every viable primitive diagram is equivalent to one in which two segments that intersect at a given dot are on the same dline iff they locally lie on a straight line, and are on the same dcircle iff they locally lie on some circle.

Finally, we have the following:

Definition 2.1.5. *A primitive diagram is* ***nicely well-formed*** *if it is well-formed and*

1. *no two dlines intersect more than once;*
2. *no two dcircles intersect more than twice;*
3. *no dline intersects any dcircle more than twice;*
4. *if a dline is diagrammatically tangent to a dcircle, then they only intersect once;*
5. *if a dline intersects a dcircle twice, then the part of the dline that is between the two intersection points must lie on the inside of the dcircle; and*
6. *given any two non-intersecting proper dlines, if there is a third dline that intersects one of them, then it also intersects the other.*

The last clause of this definition makes non-intersection of dlines an equivalence relation, and corresponds to the uniqueness of parallel lines. The first diagram in Figure 2.1 is nicely well-formed under two of the three assignments of segments to dlines and dcircles that make it a wfpd. The second diagram in Figure 2.1 is not nicely well-formed, since it contains two dlines that intersect twice. Nicely well-formed primitive diagrams are also called ***nwfpd***s.

Notice that the conditions for being viable are local conditions, the conditions for being well-formed are global conditions effecting individual dlines and dcircles, and the conditions for being nicely well-formed effect how dlines and dcircles can interact with one another globally.

2.2 Advanced Syntax of Diagrams: Corresponding Graph Structures and Diagram Equivalence Classes

We have now defined a primitive diagram to be a particular kind of geometric object. These diagrams contain somewhat too much information, though. The diagrams are supposed to show the topology of how lines and circles might lie in the plane. So we'd really like to look at equivalence classes of diagrams that contain the same topological information. In order to do this, we are going to define for each diagram an algebraic structure called a ***corresponding graph structure*** (abbreviated ***cgs***). The definition will be somewhat technical, but the idea is simple: the diagram's corresponding graph structure just abstracts the topological information contained in the diagram. Another way of saying this is that our definition will have the property that two diagrams will have isomorphic corresponding graph structures just if they have the same topological structure. A diagram D's corresponding graph structure will contain four kinds of information: a graph G that contains information about how the dots, frame, and segments intersect; for each point of intersection, information about the clockwise order in which the segments and frame intersect the point; for each doubly connected component DCC of G, a two-dimensional cell complex showing how the different regions of DCC (the connected components of the complement of DCC) lie with respect to one another; and for every connected component of G (except for the outermost component), information about which region of the graph it lies in.

(Recall that two vertices v_1 and v_2 in a graph G are said to be ***connected*** if there is a path from v_1 to v_2 in G, and they are said to be ***doubly connected*** if

for any edge e of G, there is a path from v_1 to v_2 in the graph obtained from G by removing edge e. Being connected or doubly connected are equivalence relations, and their equivalence classes are called the connected or doubly connected components of G.)

The notion of a cgs will be useful because we really want to think of two diagrams as being the same if they contain the same topological information, and so we will form equivalence classes of diagrams that have the same (isomorphic) corresponding graph structures. The corresponding graph structures are nice, constructive, algebraic objects that we can manipulate, reason formally about, or enter into a computer, rather than working directly with the equivalence classes. The data structures that **CDEG** uses to represent diagrams are essentially a version of these corresponding graph structures.

We start by defining the appropriate type of algebraic structure to capture the topology of a diagram.

Definition 2.2.1. *A **diagram graph structure** S consists of*

1. *a set of vertices $V(S)$;*
2. *a set of edges $E(S)$;*
3. *for each vertex v in $V(S)$, a (cyclical) list $L(v)$ of edges from $E(S)$ (which lists in clockwise order the edges that are connected to v, telling us how to make the edges and vertices into a graph);*
4. *a two-dimensional cell-complex for each doubly connected component of the graph;*
5. *a function er_S from the non-outermost connected components of the graph to*

the two-cells of the cell-complexes (er stands for "enclosing region", and this function tells us which region each connected component lies in);

6. *a subset $DOTS(S)$ of $V(S)$;*

7. *two subsets of $E(S)$, called $SOLID(S)$ and $DOTTED(S)$;*

8. *a set $SL(S)$ of subsets of $E(S)$; and*

9. *a set $CIRC(S)$ of pairs whose first element is a vertex and whose second element is a set of edges.*

We can now show how to construct a given diagram's corresponding graph structure. First note that the segments of a diagram D intersect the frame in a finite number of points, which divide the frame into a finite number of pieces. We refer to these points as ***pseudo-dots*** and to these pieces as ***pseudo-segments***.

Definition 2.2.2. *A diagram D's **corresponding graph structure** is a diagram graph structure S with the following properties:*

1. *$V(S)$ contains one vertex $G(d)$ for each dot or pseudo-dot d in D.*

2. *$E(S)$ contains one edge for every segment and pseudo-segment in D.*

3. *If d is any dot or pseudo-dot in D, then $L(G(d))$ lists the edges corresponding to the segments and pseudo-segments that intersect d, in the clockwise order in which the segments and pseudo-segments intersect d.*

4. *For each doubly connected component P of the graph G defined by $V(S)$, $E(S)$, and the lists $L(v)$, we define its **corresponding cell complex** C_P as follows:*

- *C_P contains two-dimensional cells, one-dimensional cells, and zero-dimensional cells.*
- *For each vertex v in P, C_P contains a corresponding 0-cell $C(v)$.*
- *For each edge e of P, C_P contains a corresponding 1-cell $C(e)$.*
- *Note that the segments and pseudo-segments of D that correspond to edges in P break up the plane into a finite number of connected regions, since there are only finitely many of them and they are piecewise arcs of circles and lines. Furthermore, because P is doubly connected, all but one of these (which we'll call the outer region) are simply connected. For each such simply connected region r, C_P contains a corresponding two-cell $C(r)$.*
- *C_P is put together by connecting the zero-cells to the one-cells so that the boundary of $C(e)$ is the set containing $C(v_1)$ and $C(v_2)$ iff e connects v_1 and v_2 in G; and then attaching the two-cells to the resulting cell-complex so that the boundary of $C(r)$ is the loop that traverses $(C(G(s_1)), C(G(s_2)), \ldots, C(G(s_n)))$ in order if and only if the boundary of r in D consists precisely of $(s_1, s_2, \ldots, s_n)$ in clockwise order.*

5. *For each connected component p of G that does not contain the edges corresponding to the pieces of the frame, $er_S(p)$ is the unique two-cell $c = C(r)$ such that*
   - *the parts of D that correspond to p lie entirely in r, and*
   - *if they also lie entirely in a region r' corresponding to some other two-cell S, then r is contained in r'.*

6. *The sets DOTTED(S), SOLID(S), DOTS(S), SL(S), and CIRC(S) are defined such that an element a of S is in one of these sets iff the corresponding element of D is in the corresponding set in D.*

This definition now allows us to say what it means for two diagrams to contain the same information.

Definition 2.2.3. *Two diagrams D and E are* ***equivalent*** *(in symbols, $D \equiv E$) if they have isomorphic corresponding graph structures.*

This is an equivalence relation, and we normally won't distinguish between equivalent diagrams. If two diagrams D and E are equivalent, then there is a natural map f between the dots and segments of one diagram and the dots and segments of the other; we say that D and E are equivalent ***via*** f. If two graphs have corresponding graph structures that are isomorphic except that the orientations are all reversed, then we say that the diagrams are ***reverse equivalent***.

Next, we would like extend our notion of a geometric diagram to allow us to mark diagrammatic angles and segments as being congruent to other diagrammatic angles and segments. A ***diagrammatic angle*** or ***di-angle*** is defined to be an angle formed where two dlines intersect at a dot in a diagram. (They do not have to be adjacent to one another.) A ***marked diagram*** is a primitive diagram in which some of the dsegs and/or some of the di-angles have been ***marked***. A dseg is marked by drawing a heavy arc from one of its ends to the other and drawing some number of slash marks through it. If the dseg is made up of a single solid line segment, then it can also be marked by drawing some number of slash marks directly through the line segment. A di-angle is marked by drawing an arc across the di-angle from one dline to the other and drawing some number of slash marks through it. The arc and slash marks are called a marker; two dsegs or di-angles

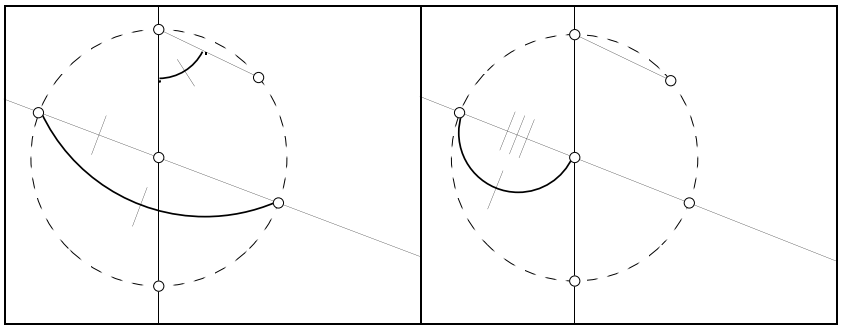


Figure 2.5: A diagram array containing two marked versions of the first primitive diagram in Figure 2.1.

marked with the same number of slashes are said to be marked with the same marker. A single dseg or di-angle can be marked more than once by drawing multiple arcs.

We would also like our diagrams to be able express the existence of multiple possible situations. In order to show these, we will use diagram arrays. A ***diagram array*** is an array of (possibly marked) primitive diagrams, joined together along their frames. (It doesn't matter how they are joined.) Diagram arrays are allowed to be empty. Figure 2.5 shows a diagram array containing two different marked versions of the first diagram in Figure 2.1.

We can extend our notion of diagram equivalence to marked diagrams and diagram arrays in the natural way. We define a ***marked diagram graph structure*** to be a diagram graph structure along with a new set MARKED whose elements are sets of dsegs and sets of ordered triples of the form <vertex, edge, edge>. We next define a marked primitive diagram D's corresponding marked graph structure to consist of the corresponding graph structure of D's underlying unmarked primitive diagram along with a set MARKED that for each segment marker in D contains the set of dsegs corresponding to the segments marked by that marker,

and for each di-angle marker in D contains the set of triples $<v, e_1, e_2>$ such that the di-angle with vertex corresponding to v and edges corresponding to e_1 and e_2 in clockwise order is marked with that marker. Two marked diagrams are defined to be equivalent if and only if their corresponding marked graph structures are isomorphic; and two diagram arrays are equivalent if and only if there is a bijection f from the diagrams of one to the diagrams of the other that takes diagrams to equivalent diagrams.

2.3 Diagram Semantics

So far, we have only talked about diagrams. Now that we know what a diagram is, we would like to discuss the relationship between diagrams and real geometric figures. By a ***Euclidean plane***, we mean a plane along with a finite number of points, circles, rays, lines, and line segments designated in it, such that all the points of intersection of the designated circles, rays, *etc.* are included among the designated points. The elements of Euclidean planes are the objects that we would like to reason about. We consider the designated points of a Euclidean plane to divide its circles and lines into pieces, which we call ***designated edges***.

It is very easy to turn a Euclidean plane P into a diagram. We can do this as follows: pick any new point n in P, pick a point p_l on each designated line l of P, and let m be the maximum distance from n to any designated point, any p_l, or to any point on a designated circle. m must be finite, since P only contains a finite number of designated points, lines and circles. Let R be a circle with center n and radius of length greater than m, and let F be a rectangle lying outside of R. Then if we let D be a diagram whose frame is F, whose segments are the parts of the

edges of P that lie inside F, whose dots are the designated points of P, and whose dlines and dcircles are the connected components of the lines and circles of P, then D is a nwfpd that we call P's ***canonical (unmarked) diagram***. (Strictly speaking, we should say a canonical diagram, since the diagram we get depends on how we pick n and the p_l; but all the diagrams we can get are equivalent, so it doesn't really matter.) We can also find P's ***canonical marked diagram*** by marking equal those dsegs or di-angles in D that correspond to congruent segments or angles in P. These canonical diagrams give us a convenient way of saying which Euclidean planes are represented by a given diagram.

Definition 2.3.1. *A Euclidean plane M is a* ***model*** *of the primitive diagram D (in symbols, $M \models D$, also read as "M satisfies D") if*

1. *M's canonical unmarked diagram is equivalent to D's underlying unmarked diagram, and*

2. *if two segments or di-angles are marked equal in D, then the corresponding segments or di-angles are marked equal in M's canonical marked diagram.*

M is a model of a diagram array if it is a model of any of its component diagrams.

This definition just says that $M \models D$ if M and D have the same topology and any segments or angles that are marked congruent in D really are congruent in M. Note that this definition makes a diagram array into a kind of disjunction of its primitive diagrams and that the empty diagram array therefore has no models.

It is immediate from the definitions that every Euclidean plane is the model of some diagram, namely its canonical underlying diagram, and that if D and E are equivalent diagrams, then if $M \models D$, then $M \models E$. In other words, the satisfaction relation is well-defined on equivalence classes of diagrams. The full converse of this

statement, that if $M \models D$ and $M \models E$, then $D \equiv E$, is not true, since D and E may have different markings. However, it is true if D and E are unmarked. Also, if D is a primitive diagram that isn't nicely well-formed, then it has no models. To see this, notice that if $M \models D$, then D's underlying unmarked diagram D' is equivalent to M's canonical unmarked diagram, which is nicely-well formed; so D' is also nicely well-formed, as diagram equivalence preserves nice well-formedness, and so D is nicely well-formed since its underlying unmarked diagram is nicely well-formed.

We are going to want to use diagrams to reason about their models. In order to do this, we are going to define construction rules that will allow us to perform operations on given diagrams which return other diagrams. So we will need some way of identifying diagrammatic elements across diagrams. To do this, we can use a ***counterpart relation***, denoted $\mathrm{cp}(x, y)$, to tell us when two diagrammatic objects that occur in different primitive diagrams are supposed to represent the same thing. Formally, the counterpart relation is a binary relation that can hold between two dots or two sets of segments in any of the primitive diagrams that occur in some discussion or proof, but never holds between two dots or sets of segments that are in the same primitive diagram. Informally, people normally use labels to identify counterparts. For example, two dots in two different diagrams might both be labeled A to show that they represent the same point. The idea of a counterpart relation is due to Shin [22].

Chapter 3

Diagrammatic Proofs

3.1 Construction Rules

We would now like to be able to use diagrams to model ruler and compass constructions. In order to do this, we will define several diagram construction rules. The rules work as follows: the result of applying a given rule to a given nwfpd D is a diagram array of (representatives of all the equivalence classes of) all the nwfpds that satisfy the rule (with corresponding parts of the diagrams identified by the counterpart relation). The new dlines and dcircles added by these rules are allowed to intersect any of the already existing dlines and dcircles, and the intersection points can be at new dots, as long as the resulting diagrams are still nicely well-formed. There will always be a finite number of resulting nwfpds, since each application of a rule will add a single new dot, dline, or dcircle, and the original diagram can only contain a finite number of dots and segments, none of which can be intersected more than twice by the new element, because of the conditions for niceness. We can apply the construction rules to diagram arrays by applying the rules to the individual primitive diagrams contained in the arrays. The diagram

Table 3.1: Diagram Construction Rules.

Diagram Construction Rules

C0. A dot may be added to the interior of any region, or along any existing segment, dividing it into two segments (unless the original segment is a closed loop, in which case it divides it into one segment).

C1. If there isn't already one existing, a dseg may be added whose endpoints are any two given existing distinct dots.

C2. Any dseg (or dray) can be extended to a proper dline.

C3a. Given two distinct dots c and d, a dcircle can be added with center c that intersects d if there isn't already one existing.

C3b. Given a dot c and a dseg S, a circle can be drawn about center c, with S designated to be a ***dradius*** of the dcircle. In general, we define a dseg to be a dradius of a dcircle if it is so designated by an application of this rule or if one of its ends lies on the dcircle and the other lies on the dcircle's center.

C4. Any dline or dcircle can be erased; any solid segment of a dline may be erased; and any dot that doesn't intersect more than one dline or dcircle and doesn't occur at the end of a dseg or dray can be erased. If a solid line segment is erased, any marking that marks a dseg or di-angle that it is a part of must also be erased.

C5. Any new diagram can be added to a given diagram array.

construction rules are given in Table 3.1.

Rule C3a is a special case of rule C3b, while C3b is derivable from C3a, as in Euclid's second proposition. Rules C1, C2, and C3a correspond to Euclid's first three postulates. Euclid's Postulates can be found in Appendix A.

As a relatively simple example of how these rules work, consider the diagram shown in Figure 3.1. What happens if we apply rule C1 to this diagram in order to connect points C and D? We get the diagram array of all nwfpds extending the

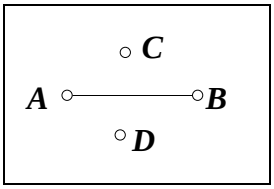


Figure 3.1: What can happen when points C and D are connected?

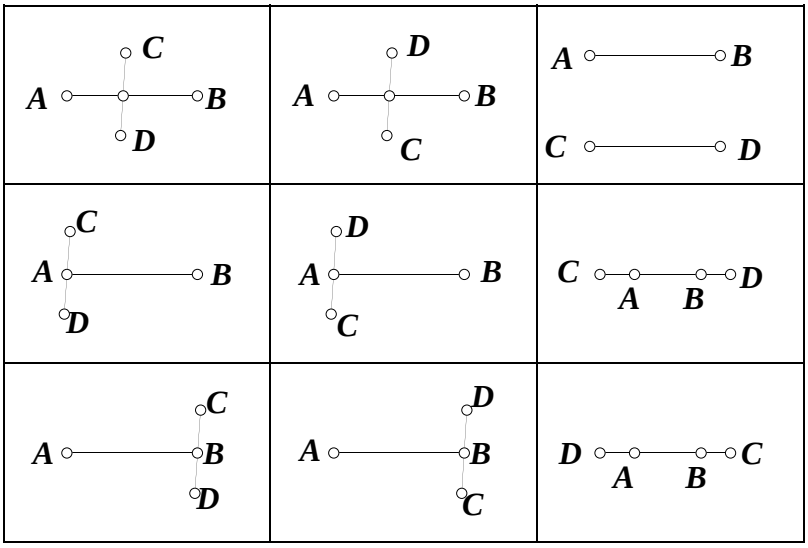


Figure 3.2: The result of applying rule C1 to points C and D in the diagram in Figure 3.1.

given diagram in which there is a dseg connecting points C and D. In this case, there are nine different topologically distinct possibilities, as **CDEG** confirms, which are shown in Figure 3.2. See Section 3.6 for a sample transcript showing **CDEG**'s output in this case.

A more useful example of these rules is given by the first four steps of the derivation of Euclid's first proposition shown in Figure 1.1, in which rule C3a is used twice, and then rule C1 is used twice. Notice that in this example, there is only one possible diagram that results from applying each of these rules. This is because many other possible diagrams have been eliminated because they are not

nicely well-formed. For example, consider the step between the third and fourth diagrams in Figure 1.1. Call the points that are being connected A and C. The fourth diagram is supposed to be the array of all diagrams extending the third diagram in which A and C have been connected by a dseg (and nothing else has been added). It is, because there is only one such diagram, but if we had picked our rules for nice well-formedness less carefully, there would have been others. Let's consider what would have happened if we had eliminated the fourth and fifth clauses in the definition of nice well-formedness (Definition 2.1.5), which say that if a dline intersects a dcircle twice, then the part of the dline that lies between the two intersection points must also lie inside the dcircle, and the dcircle cannot be dtangent to the dline at either of those points. Without these clauses, we would have gotten the array of ten diagrams shown in Figure 3.3. Thus, our definition of a nicely well-formed diagram saves us from considering many extra cases. Note that in this particular case, these extra diagrams could all be eliminated in one more step by using rule C2 to extend dseg AC into a proper dline. Since none of the extra cases can be extended in this way to give a nicely well-formed diagram (even without the fourth and fifth clauses of the definition), they would all have been eliminated.

A construction rule is said to be ***sound*** if it always models a possible real construction, meaning that if $M \models D$ and diagram E follows from D via this rule, then M can be extended to a model of E. The rules given in Table 3.1 are sound, because in any model, we can add new points, connect two points by a line, extend any line segment to a line, or draw a circle about a point with a given radius, and we can erase points, lines, and circles. In general, if every model M of D can be extended to a model of E, then we say that E is a ***geometric consequence*** of

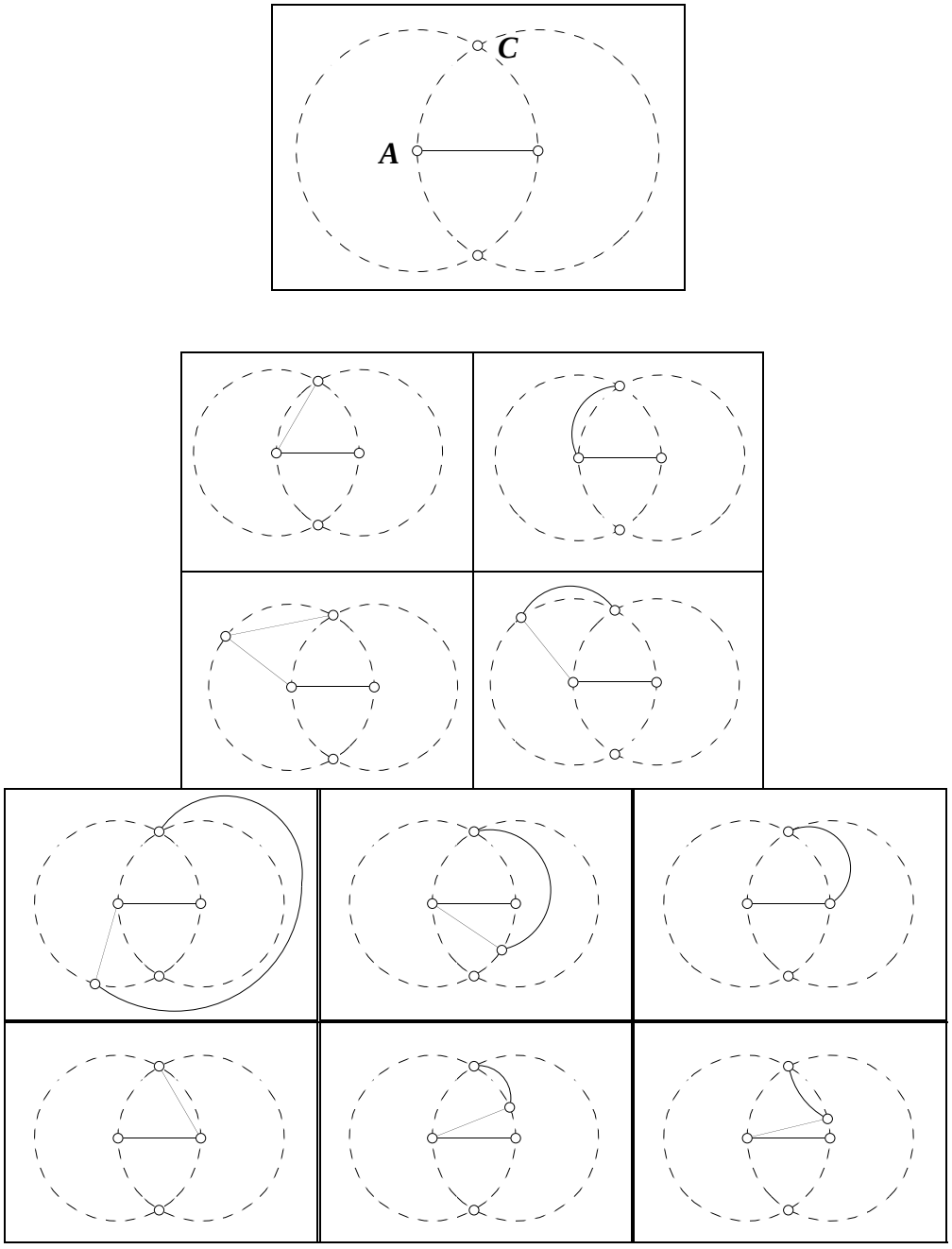


Figure 3.3: A modified construction.

D, and write $D \models\!\!\sqsubset E$. This definition of geometric consequence and the notation for it are due to Luengo [15].

A diagram E is said to be ***constructible from diagram*** D if there is a sequence of diagrams beginning with D and ending with E such that each diagram in the sequence is the result of applying one of the construction rules to the preceding diagram; such a sequence is called a construction. Because our construction rules are sound, it follows by induction on the length of constructions that if E is constructible from D, then E is a geometric consequence of D.

The computer system **CDEG** uses explicit algorithms to compute the diagram graph structure that results from applying one of the construction rules to a given diagram. These algorithms are based on the idea that if we want to know how a line can possibly continue from a given dot, it must either leave the dot along one of the already existing segments that leave the dot, or else it must enter one of the regions that the dot borders, in which case it must eventually leave that region at another dot or along another edge bordering the region, breaking the region into two pieces; along the way, it can intersect any of the pieces of any components that lie inside the region. This is reminiscent of Hilbert's axiom of plane order (II,4), which says that if a line enters a triangle along one edge, it must also leave the triangle, passing through one of the other two edges. In **FG**, this is a consequence of the definition of a nicely well-formed primitive diagram, rather than an explicitly stated axiom. This is typical: many of the facts that Hilbert adopts as his axioms of order and incidence are consequences of the diagrammatic machinery built into the definitions of **FG**.

3.2 Inference Rules

Once we have constructed a diagram, we would like to be able to reason about it. For this purpose, we have rules of inference. Unlike the construction rules, when a rule of inference is applied to a single diagram, we get back a single diagram (at most). A rule of inference can be applied to a diagram array by applying it to one of the diagrams in the array. The rules of inference are given in Table 3.2. Rules R4 and R5 decrease the number of diagrams in a diagram array, and the other rules of inference leave that number constant, so applying rules of inference never increases the number of diagrams in a diagram array. If diagram (array) F can be obtained from E by applying a sequence of construction, transformation, and inference rules, then we say that F is ***provable*** from E, and write $E \vdash F$. (The transformation rules will be explained in the next section.)

Rules R1 and R2 correspond to Euclid's common notions 1 and 2, to Hilbert's axioms III, 2 and III, 3, and to Luengo's inference rules R4.5 and R4.4. Rules R5a and R5b correspond to Euclid's fifth common notion. Hilbert assumes R5b as his axiom III, 4, and uses it to prove R5a from SAS as we will show how to do in Section 3.4, while Luengo incorporates a version of R5a into her definition of syntactic contradiction. (See [15], [10], and Appendix A.) R3 corresponds to Euclid's definition 15. We have already incorporated a version of the uniqueness of parallel lines into our definition of nice well-formedness, but we could just as well have added it here. Euclid's fourth postulate is derivable from our other rules using the symmetry transformations, and Euclid's fifth postulate is derivable from the uniqueness of parallel lines in the usual way.

The second half of the proof in Figure 1.1 uses these inference rules. Beginning with the fifth diagram in the proof, we can apply rule R3 twice and rule R1 once

Table 3.2: Rules of Inference.

Rules of inference

R1. If two dsegs or di-angles a and b are marked with the same marker and, in addition, a is also marked with another marker, then b can also be marked with the second marker.

R2. If there are four dsegs or di-angles a, b, c, and d such that a and b don't overlap and their union is also a dseg or di-angle e , and c and d don't overlap and their union is a dseg or di-angle f, then if a and c are marked with the same marker, and b and d are marked with the same marker, then e and f can be marked with the same new marker not already occurring in the given diagram.

R3. Any two dradii of a given dcircle may be marked with the same new marker.

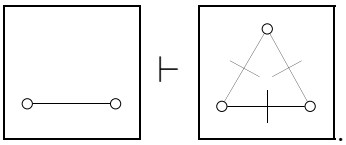
R4. Given a diagram array that contains two diagrams that are copies of one another, one of them may be removed.

R5a. (CS) If a diagram contains two dsegs, one of which is properly contained in the other, and both of which are marked with the same marker, then it can be removed from a diagram array.

R5b. (CA) If a diagram contains two di-angles, one of which is properly contained in the other, and both of which are marked with the same marker, then it can be removed from a diagram array.

R6. Any dseg or di-angle can be marked with a single new marker. Any marker can be removed from any diagram.

to obtain a diagram in which all three sides of the triangle are marked equal, and then using R7 and C4 we can erase the extra markings and the circles, leaving just the triangle. Thus, Figure 1.1 shows that



Call the first diagram here A, and the second B. Since A is certainly constructible from the empty primitive diagram, B is also provable from the empty primitive diagram. (We write this as "$\vdash B$".) Notice that, unlike what we're used to with linguistic systems, $A \vdash B$ is actually be a stronger statement than $\vdash B$, since diagrams A and B are related by the counterpart relation. So $\vdash B$ says that an equilateral triangle can be constructed, whereas $A \vdash B$ says that given any segment, an equilateral triangle can be constructed along that segment. Strictly speaking, $A \vdash B$ just means that we can get from A to B using our rules, and it is $A \not\subset B$ that means that an equilateral triangle can be constructed along any given segment. But it is an immediate consequence of the soundness of our rules that if $A \vdash B$, then $A \not\subset B$. It is easy to check that our rules are indeed sound. For example, to check that rule R1 is sound, assume that we are given two diagrams D and E such that $D \vdash E$ via rule R1. Then E differs from D only in that there are two dsegs or di-angles a and b in D and E such that in D, a is marked with two markings m and n but b is only marked with marking m, while in E, b is also marked with marking n. Since E differs from D only in that b is marked with marking n in E, to show that $D \models E$ it suffices to show that if M is a model of D and o is any element of D that is marked with marking n, then the pieces of M that correspond to o and b are congruent. Since M is a model of D, the pieces of M that correspond to a and b are congruent, since they are both marked with

marking m in D, and the pieces that correspond to a and o are congruent, since they are both marked by marking n in D; so the pieces that correspond to o and b are also congruent in M since congruence is a transitive relation in any Euclidean plane. So $M \models E$, which means that $D \models E$. The proofs that the other rules are sound are similar exercises in chasing definitions and then using a corresponding semantic fact about the models.

3.3 Transformation Rules

We would also like to be able to use diagrams to model isometries: translations, rotations, and reflections. To do this, we first need the notion of a ***subdiagram***. A primitive diagram A is a subdiagram of B if A is constructible from B using only rule C4. Next, we define a diagram T to be an ***super transformation diagram*** of A in D (via transformation t) if A is a subdiagram of D, D is a subdiagram of T, and there exists another diagram B and a function $t : A \to B$ such that B is also a subdiagram of T, and A and B are equivalent or reverse equivalent diagrams via the map t. T is a ***transformation diagram*** of A in D via t if T is an super transformation diagram of A in D via t, and no proper subdiagram S of T is still a super transformation diagram of A in D via t. If A and B are equivalent, then it is an ***unreversed*** transformation diagram, and if they are reverse equivalent, then it is a ***reversed*** transformation diagram. Now we can incorporate symmetry transformations into our system by adding the rules in Table 3.3. Note that simple rotations and translations are special cases of rule S1, and reflections are a special case of rule S2.

Each of these rules, like the construction rules, always yields a finite number

Table 3.3: Transformation Rules

Transformation Rules

S1. (glide) Given a diagram D, the subdiagram A, a dot a and a dseg l_1 ending at a in A, and a dot b and a dseg l_2 ending at b in D, the result of applying this rule is the diagram array of all unreversed transformation diagrams of A in D such that $t(a) = b$ and $t(l_1)$ lies along the same dline as l_2, on the same side of b as l_2.

S2. (reflected glide) Given a diagram D, the subdiagram A, a dot a and a dseg l_1 ending at a in A, and a dot b and a dseg l_2 ending at b in D, the result of applying this rule is the diagram array of all reversed transformation diagrams of A in D such that $t(a) = b$ and $t(l_1)$ lies along the same dline as l_2, on the same side of b as l_2.

of consequences when applied to a single diagram. This is because the unmarked diagram array that results from applying one of these rules and then erasing all markings is a subarray of the the array that is obtained by constructing a copy of A in the appropriate spot in D using the construction rules.

The system that contains the construction rules C0–C4, the transformation rules S1 and S2, and the rules of inference R1–R6 is called **FG** (for "Formal Geometry").

As an example of how these transformation rules work, consider the diagram found in Figure 3.4. It is a logical consequence of this diagram that EF is congruent to BC, and we should therefore be able to mark it with three slash marks. This is one particular case of the rule of inference SAS:

SAS. If a diagram contains two triangles, such that two sides of one triangle and the included di-angle are marked the same as two sides and the included di-angle of the other triangle, then the remaining sides of the triangles can be marked

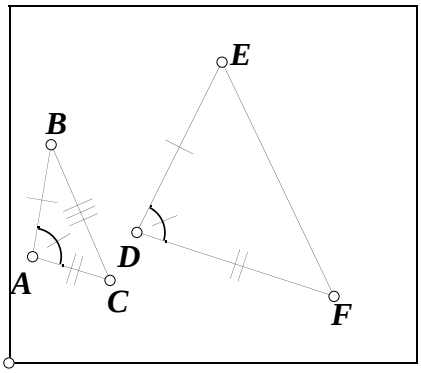


Figure 3.4: The hypothesis diagram for one case of SAS.

with the same new marker, and each of the remaining di-angles of the first triangle can be marked the same as the corresponding di-angle of the other.

In **FG**, SAS is a derived rule; it can be derived from our symmetry transformations along with CA and CS. The proof is essentially identical to Euclid's proof of his fourth proposition, with a lot of tedious extra cases showing all of the ways that the triangles could possibly intersect. The idea is to move the two triangles together using the symmetry transformations and to then check that they must be completely superimposed.

In **FG**, the proof of this case of SAS has two steps. The first step is to apply rule S1 to the diagram in Figure 3.4, moving triangle ABC so that A' (= t(A)) coincides with D, and so that the image $A'B'$ of AB lies along DE. The possible cases that result are shown in the diagram arrays in Figures 3.5 and 3.6. For the sake of readability, many of the markings have been left off these diagrams, although all markings that are later needed have been left. Also, properly speaking, these figures are only some of the cases that are given by rule S1, because any part of $A'B'C'$ that lies outside of DEF can intersect ABC in any one of a number of ways. But the diagrams do show all of possible cases in which $A'B'C'$ doesn't

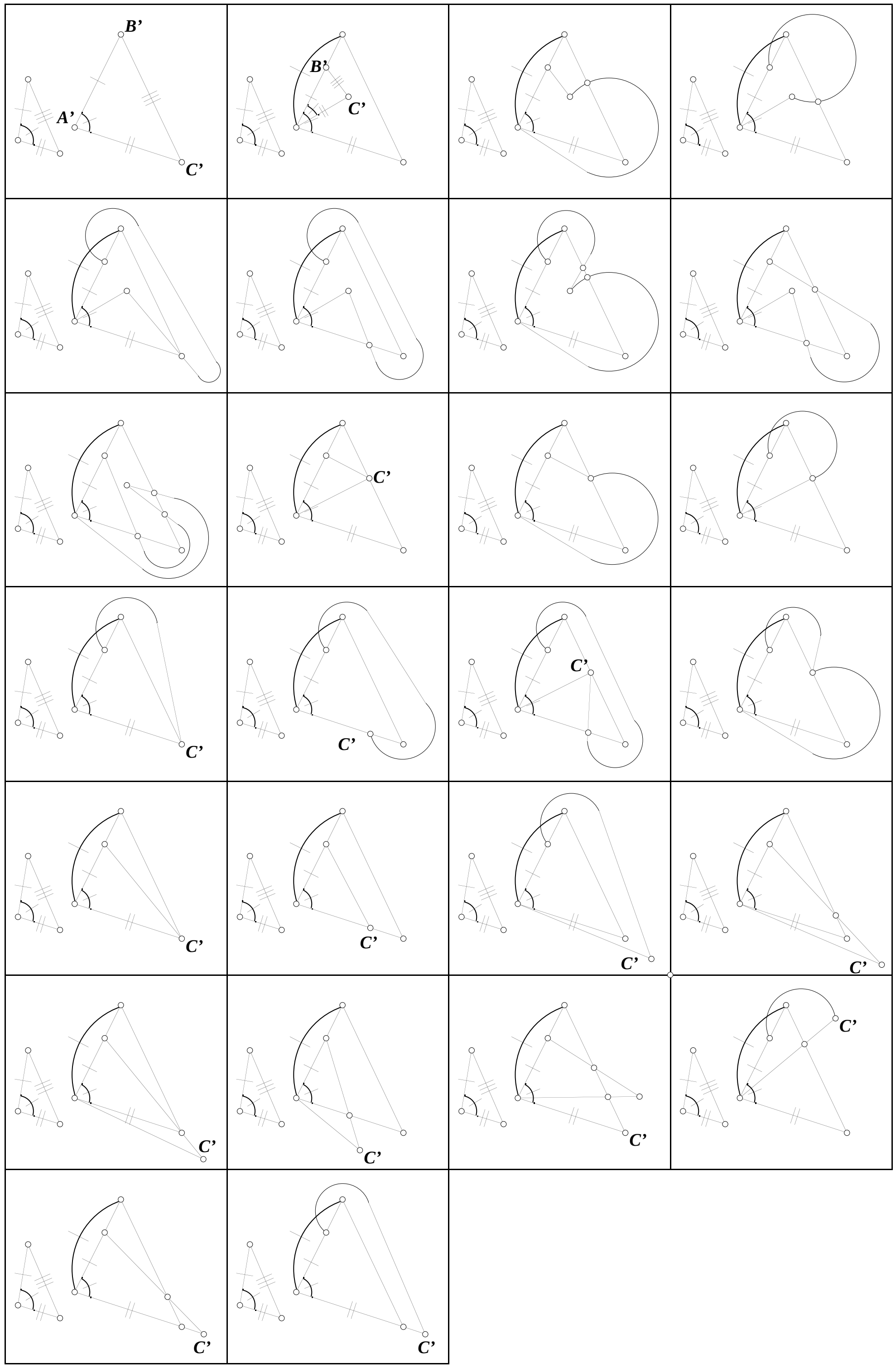


Figure 3.5: The first half of the cases that result from applying rule S1 to the diagram in Figure 3.4.

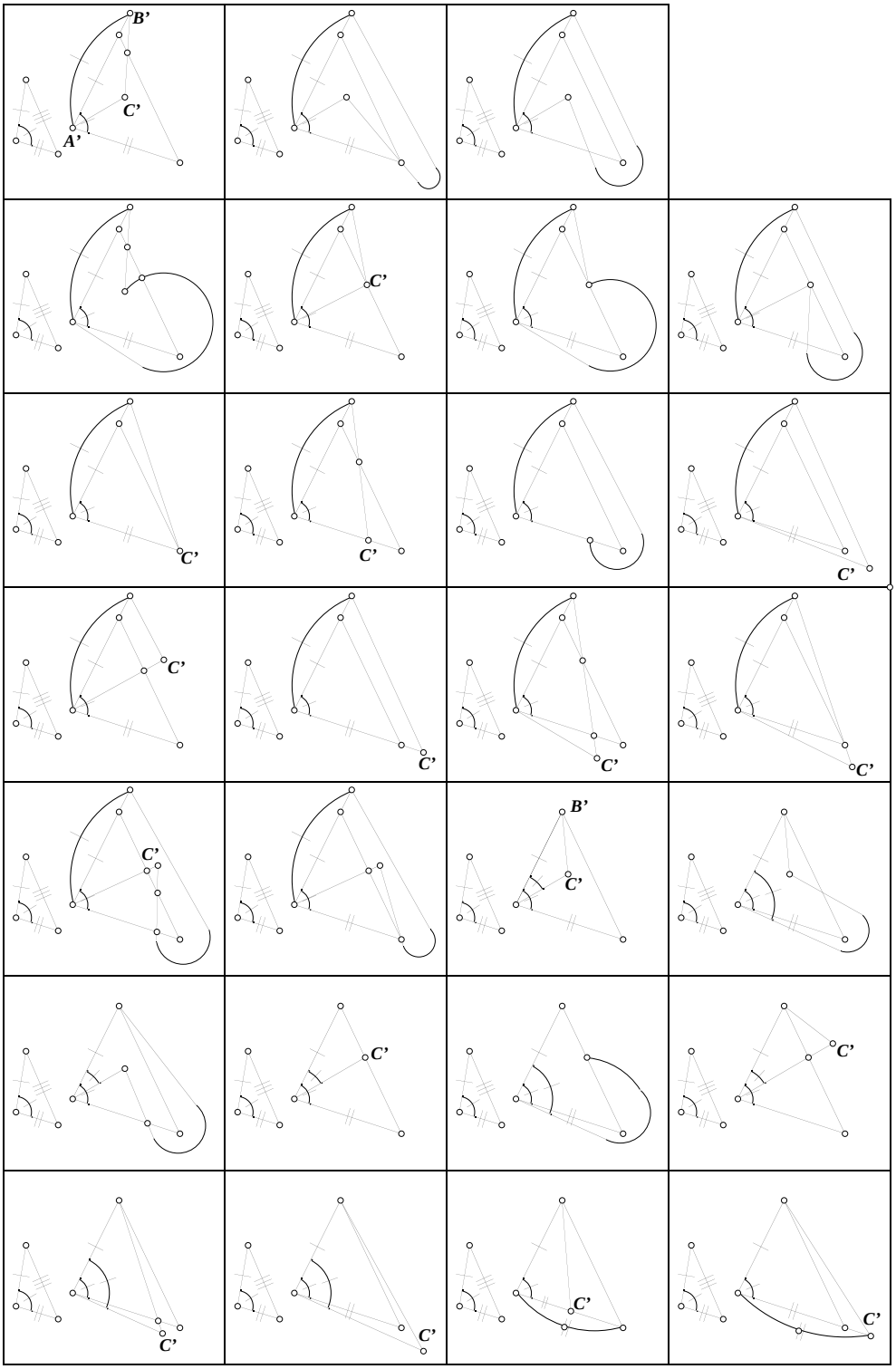


Figure 3.6: The second half of the cases that result from applying rule S1 to the diagram in Figure 3.4.

intersect ABC.

The second step is to remove all of the extra cases using the rules of inference CA and CS. All of the diagrams shown in Figure 3.5 except for the very first one can be eliminated by applying CS to $A'B'$ and DE. In Figure 3.6, all of the diagrams in the first four rows and the first two diagrams in the fifth row are eliminated in the same way. The rest of the diagrams can be eliminated by using CA, except for the last two diagrams, which can also be eliminated by using CS. The cases that weren't shown in Figures 3.5 and 3.6, in which $A'B'C'$ intersects ABC, can also all be eliminated using CS and CA. Thus, we have shown SAS for one particular case, in which the two original triangles don't intersect and have the same orientation. The proof for the other cases is similar.

3.4 Transformations and Weaker Systems

Most formal systems for doing geometry (Hilbert's, for example) don't contain rules for doing symmetry transformations; rather, they include a version of the rule of inference SAS. In **FG**, SAS is a derived rule that can be proven in the same way Euclid proved his fourth proposition.

However, in our system, we can also consider sets of rules that are weaker than **FG**, so that SAS can no longer be derived from them, but which are still strong enough to prove some of the things that are normally proved using SAS. For example, consider the system **GS** ("Geometry of Segments") in which we have all the rules of construction, transformation, and inference except for CA. SAS is not a derived rule of this system. To see this, consider a modified definition of satisfaction in which $M \models D$ iff M's canonical unmarked diagram is equivalent

to D's underlying unmarked diagram, and if two dsegs in D are marked with the same marker, then the corresponding dsegs in M's canonical marked diagram are also marked with the same marker (so that we have dropped the corresponding requirement for di-angles). All of the rules of **GS** are still sound with respect to the new notion of satisfaction (call it **GS**-satisfaction), but CA and SAS are no longer sound. This is because the definition of **GS**-satisfaction says that any two angles can be marked with the same marker even if they aren't really congruent, so it is possible to have an angle properly contained in another with the same marking; and the corresponding angles of two triangles with congruent sides could be marked with the same marking even if they aren't really congruent, so that the resulting triangles aren't congruent either. Thus, neither CA nor SAS is derivable in **GS**, since it is impossible to derive an unsound rule from sound rules. On the other hand, many consequences of SAS still hold: for example, the SSS rule for triangle congruence can still be derived. (This is plausible, since the SSS rule is still sound with respect to **GS**-satisfiability.)

Here is a description of how to derive the SSS rule in the system **GS**: given two triangles whose sides are marked equivalent, use the symmetry transformations and CS to move the second triangle so that its first side coincides with the first side of the first triangle and the two triangles are oriented the same way. Either the second triangle lies precisely on top of the first, in which case we're done, or else we have a situation that looks like the first diagram in Figure 3.7. Reflect the two triangles over their common base line, giving the situation shown in the second diagram in Figure 3.7. Construct the circles c_1 and c_2 with centers A and B through point C. It follows from CS that if a circle is drawn with center Z through a point X and ZX is marked congruent to some other segment ZY also ending at

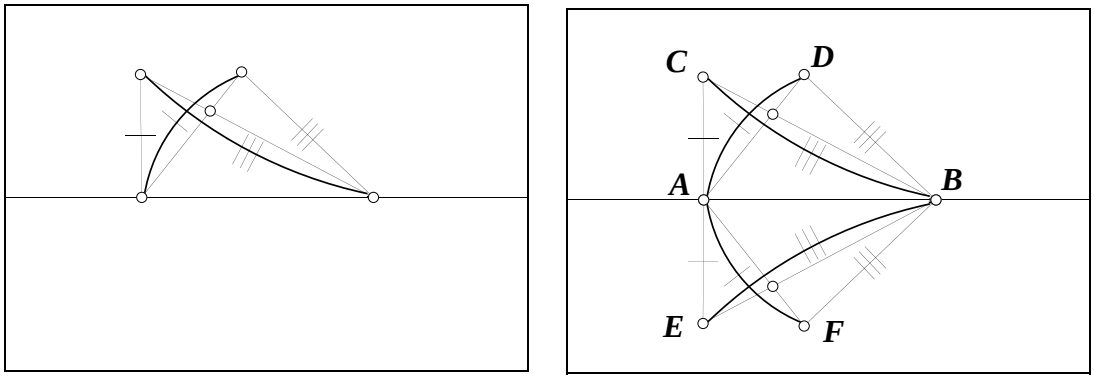


Figure 3.7: Steps in the proof of SSS.

Z, then the circle must also pass through Y; otherwise, if the circle intersects ray ZY at point W, then ZW can be marked congruent to ZX and therefore marked congruent to ZY, but one of ZY and ZW must be properly contained in the other, a contradiction by CS. The two circles c_1 and c_2 must therefore each intersect the four distinct points C, D, E, and F; but two distinct circles can only intersect in at most two points; a contradiction.

So what is the relationship between CA and SAS? They are in fact equivalent in **GS**. In the previous section, we showed how to derive SAS in **FG**; this shows that SAS can be derived from CA in **GS**. But CA can also be derived from SAS in **GS**, as follows. Let us be given a diagram in which two di-angles, one contained in the other, are marked with the same marking, as in Figure 3.8, and let us denote the di-angles BAE and BAF. We need to show how to eliminate this diagram in **GS**. To do this, we can mark off equal length segments AC and AD along AE and AF (using rule C0 to add a dot C along AE, using rule C3a to draw a circle about A through C, labeling the intersection of this circle with AF as D, and then using R3 to mark AC and AD the same length). Then, if we connect C and D to B, we will be left with a situation like that shown in Figure 3.8. Marking AB

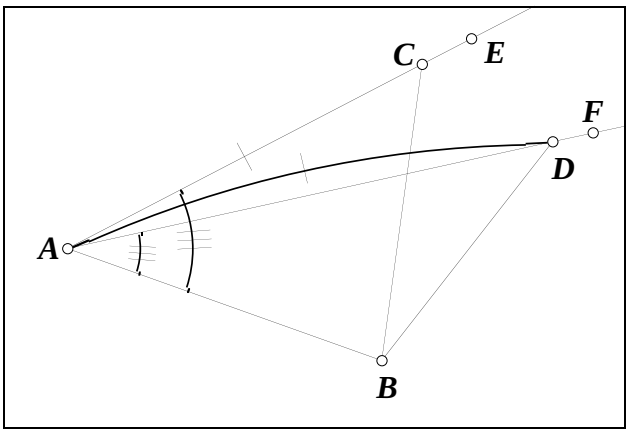


Figure 3.8: Deriving CA from SAS in **GS**.

with a new marker, and applying SAS to triangles CAB and DAB, we can mark CB and DB congruent with a new marker. Notice that we now have the same situation encountered in the proof of SSS and shown in Figure 3.7a, in which we have two different triangles with congruent sides on a single base. As before, we can show that this situation is impossible by reflecting the triangles over the base and then drawing two circles which would have to intersect in four places. This shows that SAS implies CA in **GS**, and so SAS and CA are equivalent in **GS**.

Similarly, we can define a system **GA** ("Geometry of Angles"), which contains all of the rules of **FG** except for CS, and a corresponding notion of **GA**-satisfaction in which $M \models D$ iff M's canonical unmarked diagram is equivalent to D's underlying unmarked diagram and if two di-angles in D are marked with the same marker, then the corresponding di-angles in M's canonical marked diagram are also marked with the same marker (so that here we have dropped the corresponding requirement for dsegs). Again, all of the rules of **GA** are sound with respect to **GA**-satisfaction, but neither CA nor SAS are; this shows that neither CA nor SAS can be derived in **GA**. Furthermore, we can again show that SAS and CS are equivalent in **GA**. CS implies SAS in **GA** as before; again, this is shown by

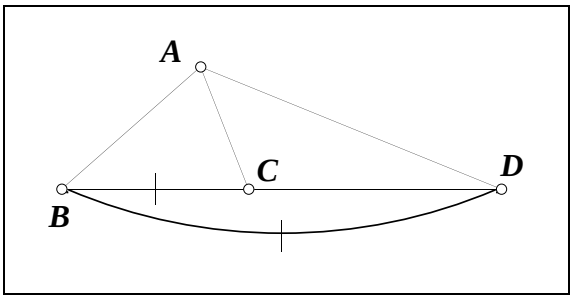


Figure 3.9: Deriving CS from SAS in **GA**.

the proof that SAS is derivable in **FG**. So it suffices to show that CS is derivable from SAS in **GA**. To show this, let us be given a diagram in which one segment is contained in another with the same marking; call the first segment BC, and call the second BD. Next, pick (or construct) another point A that doesn't lie on the line BD. This gives the situation shown in Figure 3.9. Marking angle CBA and segment BA congruent to themselves with new markers, we can apply SAS to triangles CBA and DBA. This allows us to mark angle BAC congruent to angle BAD; but BAC is contained in BAD, and so we can eliminate this diagram by CA. This shows that CS is derivable from SAS in **GA**, and that SAS and CS are therefore equivalent in **GA**. This proof that CA and SAS together imply CS is identical to Hilbert's proof of the uniqueness of segment construction in [10].

Finally, we can look at a formal system that doesn't contain either CA or CS, but instead contains SAS. Let **BG** ("Basic Geometry") be the formal system containing all of the rules of **FG** except for CS and CA, and let **GSAS** ("Geometry of SAS") be **BG** with the added rule SAS. We have already shown that SAS and CS together imply CA in **BG**, and that SAS and CA together imply CS in **BG**, so this means that CA and CS are equivalent in **GSAS**. However, neither CS nor CA is derivable in **GSAS** without the other. To see this, define MM-satisfaction ("Meaningless Marker satisfaction") so that $M \models D$ iff M's canonical unmarked

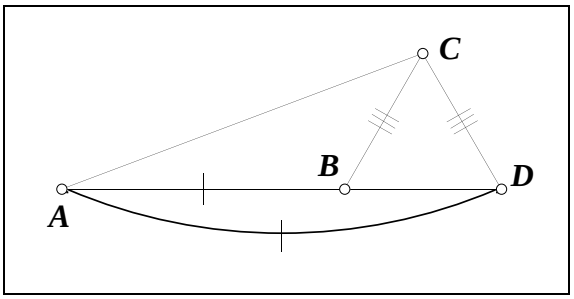


Figure 3.10: Deriving CS from SSS in **GA**.

diagram is equivalent to D's underlying unmarked diagram. This allows any angle or segment to be marked the same as any other angle or segment, so that the markings have become meaningless. All of the rules of **BG** are sound with respect to MM-satisfaction, and so is SAS, because we can safely mark any dsegs or di-angles congruent without changing the models of a diagram; but neither CS nor CA are sound with respect to MM-satisfaction, since there are lots of diagrams satisfying the hypotheses of these rules which are still MM-satisfiable.

We have shown that the following interesting situation holds:

Theorem 3.4.1. *CS, CA, and SAS are independent of one another in* **BG***—that is, no one of them is provable from any other in* **BG***. However, any two of them are equivalent in the presence of the third, so that any one of them is provable from the other two.*

Notice that while SSS is provable from CS in **BG**, CS is not provable from SSS, because SSS is sound with respect to MM-satisfaction. So SSS is a weaker axiom than CS relative to **BG**. Relative to **GA**, however, the two axioms are equivalent: if we are given a diagram in which AB and AD are marked congruent and B lies on AD, as in Figure 3.10, we can construct an equilateral triangle on BD as in Euclid's first proposition. Calling the new vertex of this triangle C, we can connect

C to A. If we mark AC with a new marker, we can apply SSS to triangles CBA and CDA. This allows us to conclude that angle ACB is congruent to angle ACD, which gives us the condition to apply CA and eliminate the diagram. So adding SSS and CA to **BG** gives us all of **FG**, while adding SSS and CS to **BG** just gives us **GS**. Adding SSS and SAS to **BG** gives us a system that is weaker than **FG**, because it is sound with respect to MM-satisfaction, but may be stronger than **GSAS**. (I conjecture but haven't proven that SSS isn't provable in **GSAS**.)

We could go on proving results like this for quite some time. For another example, the Isosceles Triangle Theorem (ITT), which says that if two sides of a triangle ABC are congruent, then its corresponding angles are also congruent, is not provable in **BG**, but can be proven by applying either SSS or SAS to ABC and CBA, so it is provable in both **GSAS** and **GS**. ITT isn't provable in **BG**because it isn't valid with respect to **GA**-satisfaction (but all of the rules of **BG** are). On the other hand, SAS isn't provable in **BG** + ITT, since ITT is valid with respect to **GS**-satisfaction and SAS isn't.

A nice way to think about all of these results is in terms of the lattice of subtheories of **FG**.

Definition 3.4.1. *A **derivation theory** is set $\mathcal{T}$ of pairs of diagram arrays with the property that if $(d_1, d_2) \in \mathcal{T}$ and $(d_2, d_3) \in \mathcal{T}$, then $(d_1, d_3) \in \mathcal{T}$.*

Definition 3.4.2. *The **theory of diagrammatic formal system** F, denoted Th(F), is the set of pairs (d_1, d_2) such that $d_1 \vdash d_2$ in F.*

The set of subtheories of Th(**FG**), that is, the set of all subsets of Th(**FG**) which are themselves derivation theories, forms a lattice under the partial ordering given by the set inclusion relation. (A lattice is a partially ordered set in which

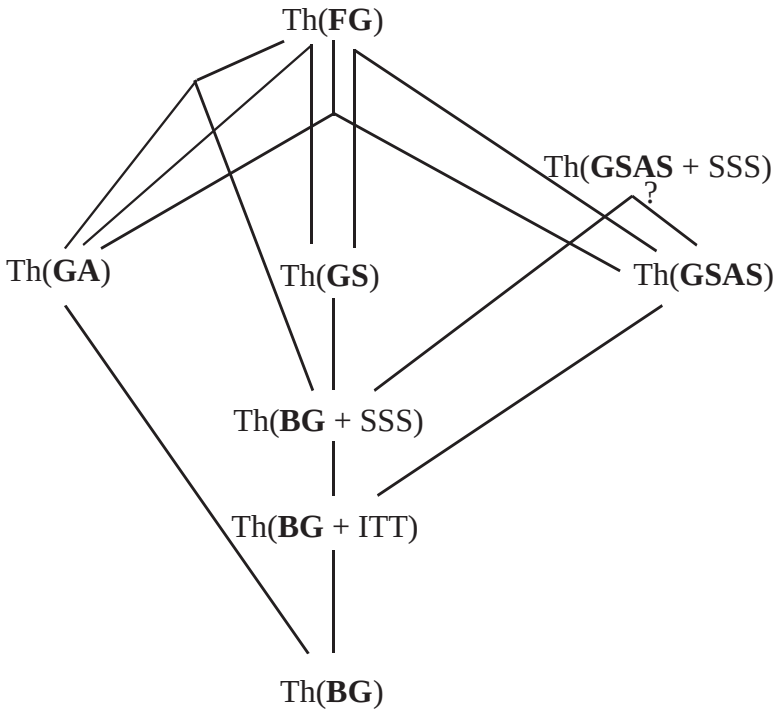


Figure 3.11: Part of the lattice of subtheories of Th(**FG**).

any two elements a and b have a least upper bound, written "$(a \vee b)$" and called their ***join***, and a greatest lower bound, written "$(a \wedge b)$" and called their ***meet***.) All of the above results about the relationships between the various axioms CA, CS, SAS, SSS, and ITT can be restated as facts about this lattice. For example, the fact that CS (and therefore all of **FG**) can be derived from SAS and CA can be restated by saying that (Th(**GSAS**) $\vee$ Th(**GA**)) = Th(**FG**).

All of the above results can be put together to give the structure of part of this lattice, shown in Figure 3.11.

3.5 Lemma Incorporation

One benefit of formalizing our proof system is that the formal results can shed light on proof practices in the informal system. For example, it is very common for informal proofs of geometric facts to rely on other theorems that have already been proven. This use of previously proved lemmas in proofs is a normal practice in most of mathematics, but it is particularly common in geometry, and while it is clear how lemma incorporation works formally in traditional sentential proofs, it is less clear how it should work in diagrammatic proofs. In a sentential proof, whenever a lemma is used, the proof of the lemma can be inserted to give a proof that doesn't rely on the lemma. Furthermore, using the lemma doesn't shorten the proof at all: the length of the new proof is the same as the length of the original proof plus the length of the proof of the lemma.

In a diagrammatic proof, on the other hand, the diagram in which you want to apply a lemma is generally much more complicated than the diagram in which you first proved the lemma. This makes it less clear how lemma incorporation should work formally, but points towards one of the reasons that it is useful: we can save ourselves some work by proving lemmas in the simplest possible environments. In fact, some diagrammatic proofs can be made exponentially shorter by using lemmas. This is because lemma incorporation can partially make up for one of the great weaknesses of diagrams: unnecessary case analysis stemming from too much information in the diagram.

For example, consider a diagram like the first one in Figure 3.12 that contains n circles and a line segment. What happens when we try to extend this line segment to a line? Each circle could end up on either side of the line (or it could be on the line), so there are at least 2^n different cases which could result. One of

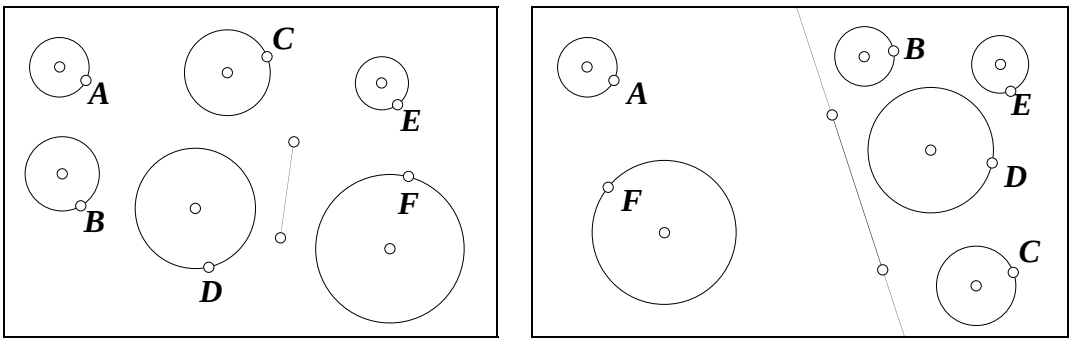


Figure 3.12: Extending a line can give rise to exponentially many new cases.

these is shown by the second diagram in Figure 3.12. This is a real disadvantage of working diagrammatically: applying one construction rule can give us exponentially many new cases to consider. What's worse, the extra cases might be totally superfluous—what if the line segment was being extended for some part of the proof that had nothing to do with the circles? In that case, one of the strengths of this system—that it can do complete case analysis of the diagrams—becomes a weakness, because we are forced to consider cases that may never play any important role in a proof.

This is where lemma incorporation becomes useful. By proving lemmas in a simple environment and then applying them, we can avoid unnecessary case analysis. As an example, consider what happens if we want to apply SAS to a diagram that contains n disconnected components along with the two triangles. If we apply SAS as a lemma, we can conclude that the triangles are congruent in one step, so that the total length of the proof is the same regardless of how big n is, even if we include the length of the proof of the lemma in the length of the proof. But if we try to mimic the proof of the lemma directly in the diagram with the n extra disconnected components, we will get at least 2^n extra cases, because the proof involves moving one triangle to the other, and each of the disconnected components

might or might not end up lying inside the moved triangle. Commentators often point out that Euclid proved SAS using superposition and then used SAS rather than superposition to prove his other theorems, and sometimes claim that this indicates that he viewed superposition as being a suspect method of proof. (See for example Thomas Heath's commentary on Euclid's fourth proposition in [7, pp. 224–231, pp. 249-250].) Using SAS as a lemma makes proofs simpler and shorter than the corresponding proofs that use superposition directly, however, so we shouldn't be surprised that this is what Euclid did.

So how does lemma incorporation work in the diagrammatic world? We would like to claim that, just as in the sentential world, we can incorporate lemmas into our proofs in such a way that any diagram that we can derive from another using a lemma could have been derived without using the lemma. First, we need a way to combine the diagram from the lemma with the original diagram.

Definition 3.5.1. *The* ***unification*** *of primitive diagrams a and b (written as unif(a, b)) is the diagram array containing all minimal diagrams d that contain both a and b as subdiagrams.*

Recall that a primitive diagram a is a subdiagram of d if a can be obtained from d by using rule C4 to erase pieces of d. We say that a segment in d ***comes from*** a if it is part of a set of segments in d that is related by the counterpart relation to a set of segments in a, and that a dot in d comes from a if any of the segments that it intersects come from a. Notice that a diagram d is in unif(a, b) just if it contains a and b as subdiagrams, and every dot and segment in d comes from either a or b.

Next, we'd like to extend these ideas to diagram arrays. If A and B are diagram arrays, we say that A is a ***subdiagram*** of B via the matching function $f : B \to A$ iff for every primitive diagram b in B, $f(b)$ is a primitive diagram a in A such

that a is a subdiagram of b. Likewise, if 2^A denotes the set of subsets of the set of diagrams in A, then given two diagram arrays A and B and a function $g : B \to 2^A$, the ***unification*** $\text{Unif}_g(A, B)$ of A and B with respect to g is the smallest diagram array such that for each primitive diagram b in B and each primitive diagram a in $g(b)$, $\text{Unif}_g(A, B)$ contains all the diagrams in $\text{unif}(a, b)$. Finally, if $A \vdash A^*$ via derivation D, then we can define an ***ancestor relation*** ancestor_D between primitive diagrams in A and primitive diagrams in A^* in the natural way: every primitive diagram in a diagram array at one step of the derivation is descended from a unique primitive diagram in the preceding array, unless rule R4 removing copies is applied, in which case the remaining copy has two ancestors in the preceding generation. Thus, ancestor_D relates $a_0 \in A$ and $a_1 \in A^*$ if a_1 is descended from a_0.

As an example of how this works, consider Figure 3.13. Let B be the diagram array on the first line, let A be the diagram array on the second line, and let A^* be the diagram array on the last line. A is a subdiagram of B via the matching function f shown by the arrows between the first and second lines. A^* can be derived from A in three steps: first, the third diagram in A is removed using rule R4, since it is a copy of the second diagram in A; next, a new diagram is added to the array using rule C5; and finally, rule C1 is applied to points C and D in the first diagram in the array. This yields A^*, in which the first nine primitive diagrams correspond to the nine different ways of connecting points C and D, and the last two primitive diagrams are the same as in the preceding array. The arrows in Figure 3.13 show how the diagrams are descended from the diagrams in the preceding array at each step. The first nine diagrams in A^* are descended from the first diagram in A; the tenth diagram is descended from both of the other

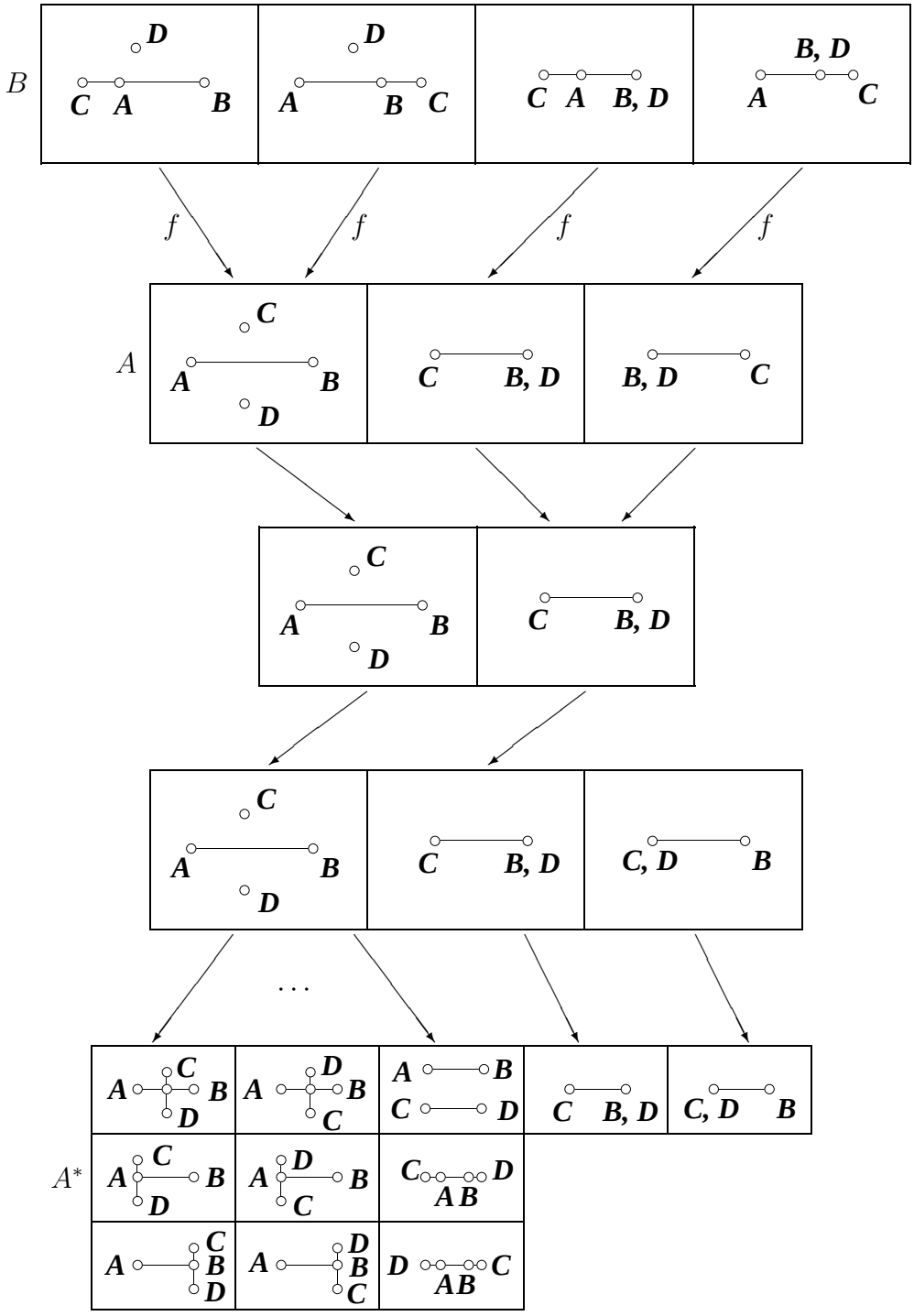


Figure 3.13: An example of lemma incorporation.

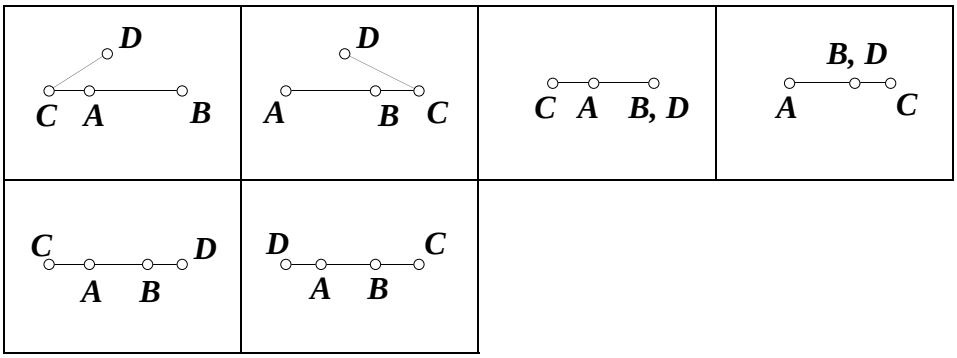


Figure 3.14: The result of unifying B and A^* in Figure 3.13.

diagrams in A; and the eleventh diagram in A^* isn't descended from any diagram in A.

How would we apply this derivation to B? Intuitively, it seems that we would want to unify each diagram in B with the diagrams that are descended from the corresponding diagrams in A. This would give us the unification of A^* and B with respect to the function g that takes each diagram b in B to the set of descendants of $f(b)$. In this particular case, we would unify each of the first two diagrams in B with each of the first nine diagrams in A^*, and we would unify the other two diagrams in B with the tenth diagram in A^*. Doing this gives us the diagram array B^* shown in Figure 3.14.

In this case, it is clear that B^* can be derived from B directly, by applying rule C1 to points C and D in the first two diagrams in B. We would like to know that any diagram like this, the result of unifying a diagram B with a diagram derived from a subdiagram of B, could have been derived directly from B. This is what the Lemma Incorporation Theorem tells us.

Theorem 3.5.1 (Lemma Incorporation). *Assume that A is a subdiagram of B via matching function f, and $A \vdash A^*$ via derivation D. Let g and U be defined*

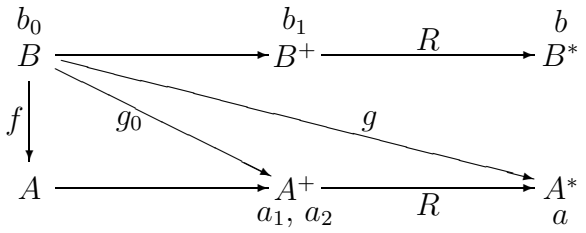


Figure 3.15: Lemma Incorporation.

so that $g(b) = \{d \in A^* | ancestor_D(f(b), d)\}$ *and* $U = Unif_g(A^*, B)$. *Then* $B \vdash U$.

Proof. We prove this by induction on the length of the proof that $A \vdash A^*$.

(Base Case) In this case, $A = A^*$, and ancestor_D is the identity function, so $\text{Unif}_g(A^*, B) = \text{Unif}_f(A, B) = B$, since A is a subdiagram of B via f. So $B \vdash \text{Unif}_g(A^*, B) = B$ trivially.

(Inductive Case) Assume that $A \vdash A^+$, that $A^+ \vdash A^*$ in one step via construction, inference, or transformation rule R, and that $\text{Unif}_{g_0}(A^+, B)$ has already been derived from B, by the inductive hypothesis, where $g_0(b) = \{d \in A^+ | \text{ancestor}_D(f(b), d)\}$. Let $U = \text{Unif}_g(A^*, B)$, $B^+ = \text{Unif}_{g_0}(A^+, B)$, and let B^* denote the result of applying rule R to the primitive diagrams in B^+ that were obtained as unifications of the primitive diagram in A^+ to which R was applied, unless R is C5 or R4. The resulting situation is illustrated in Figure 3.15. We apply the rule R to the same objects in B^+ that it was applied to in A^+; this is possible since A^+ is a subdiagram of B^+. Since $B \vdash B^*$ (because $B \vdash B^+$ by the inductive hypothesis and $B^+ \vdash B^*$ by rule R), it suffices to show that $B^* \subseteq U$, since then $B^* \vdash U$ by rule C5. We will show this for all of the rules except for C5 and R4, in which cases we will show that $U = B^+$, so that U is again derivable.

- *R is a construction, inference, or transformation rule adding object(s) l to diagram $a_1 \in A^+$.* Here, l can consist of dots, segments, and/or markers.

We want to show that $B^* \subseteq U$. Pick $b \in B^*$. If b wasn't modified by R, then $b \in B^+$, and b corresponds to a diagram $a_2 \in A^+$ other than a_1 such that $b \in \text{unif}(a_2, b_0)$ for some diagram $b_0 \in B$. Since $a_2 \neq a_1$, $a_2 \in A^*$, so $b \in \text{unif}(a_2, b_0) \subseteq \text{Unif}_g(A^*, B) = U$. On the other hand, if b was modified by R, then b is a descendent of some diagram $b_1 \in B^+$, where $b_1 \in \text{unif}(a_1, b_0)$ for some $b_0 \in B$. This means that b_1 contains a_1 as a subdiagram, and the pieces of b_1 that don't come from a_1 must come from b_0. Since b is b_1 with added object l, it contains a subdiagram a that consists of a_1 with added object l. This subdiagram is therefore in A^*. Thus, since all of the pieces that are in b_1 that don't come from a_1 come from b_0, all of the pieces of b that don't come from a come from b_0 also, since a and b consist of a_1 and b_1 with l added in the same way. So b is in $\text{unif}(a, b_0)$ and therefore in U; so $B^* \subseteq U$.

- *R is a rule (either C4 or the second part of R6) that removes some object(s) l from the diagram.* To show $B^* \subseteq U$, assume $b \in B^*$ is descended from $b_1 \in B^+$ and that $b_1 \in \text{unif}(a_1, b_0)$, where $a_1 \in A^+$ and $b_0 \in B$. Either a_1 wasn't modified by R, in which case $b \in U$ as in the previous case, or else there exists a diagram $a \in A^*$ that consists of a_1 with l removed. In this case, $b \in \text{unif}(a, b_0)$. We know this because any diagrammatic object *do* in b is in $b_1 = \text{unif}(a_1, b_0)$ and not in l, so it is in either a_1 or b_0, and isn't in l, so if it's in a_1, then it is in fact in a. So *do* is either in a or in b_0, so it's in $\text{unif}(a, b_0)$. So $B^* \subseteq U$.

- *R is a rule (either R5a, or R5b) that removes an inconsistent diagram D from A^+.* Pick $b \in B^*$. Then b was also in B^+. (Because some other diagram was

removed.) So $b \in \text{unif}(a_1, b_0)$ for some $a_1 \in A^+$ and $b_0 \in B$, and a_1 wasn't the diagram that was removed from A^+ (since then b would have been removed from B^+). It therefore follows that $a_1 \in A^*$, and so $b \in \text{Unif}_g(A^*, B)$.

- *R is rule C5, adding a new diagram n to diagram array A^+.* Then, since n isn't in the image of g, $\text{Unif}_g(A^*, B) = \text{Unif}_{g_0}(A^+, B) = B^+$. So $U = B^+$ and U has therefore already been proven in this case.

- *R is rule R4, removing a diagram d_2 that is a copy of some other diagram d_1 in A^+.* We want to show that in this case, $B^+ = U$, so that U has already been derived from B. First, we show that $B^+ \subseteq U$. Choose $b_1 \in B^+$. We know that $b_1 \in \text{unif}(b_0, a_1)$ for some $b_0 \in B$ and $a_1 \in A^+$. If $a_1 \neq d_2$, then a_1 is still in A^*, so $b_1 \in \text{unif}(a_1, b_0) \subseteq \text{Unif}_g(A^*, B) = U$ as required. On the other hand, if $a_1 = d_2$, this means that $g_0(b_0)$ contains d_2, in which case $g(b_0)$ contains d_1, since d_1 is an ancestor of d_2. Since d_1 and d_2 are copies of one another, $\text{unif}(b_0, d_2) = \text{unif}(b_0, d_1)$. So, in particular, $b_1 \in \text{unif}(b_0, d_2) = \text{unif}(b_0, d_1) \subseteq \text{Unif}_g(A^*, B) = U$, as required. So $B^+ \subseteq U$.

  On the other hand, if $b \in U$, then $b \in \text{unif}(b_0, a)$ for some $b_0 \in B$ and $a \in A^*$ such that $a \in g(b_0)$. Then either $a \in g_0(b_0)$, in which case $b \in \text{unif}(b_0, a) \subseteq \text{Unif}_{g_0}(A^+, B) = B^+$, or else $a = d_1$ and $d_2 \in g_0(b_0)$, in which case $b \in \text{unif}(b_0, a) = \text{unif}(b_0, d_1) = \text{unif}(b_0, d_2) \subseteq \text{Unif}_{g_0}(A^+, B) = B^+$. So $b \in B^+$; so $U \subseteq B^+$; and so $U = B^+$, which is what we were trying to show.

So U is derivable in each case, as required.□

3.6 CDEG

In this section, we demonstrate how the computer system **CDEG** works by using it to derive Euclid's first proposition, and also to do the case analysis discussed in Figures 3.1 and 3.2.

CDEG is driven by a text interface, and also has the ability to open windows with pictures displaying the diagram being worked on. The following is a transcript of a **CDEG** session. First we start **CDEG** and ask it what commands are available:

```
Welcome to CDEG 1.0!
(type h for help)
CDEG(1/1)% h
Options are:
<s>ave, <l>oad, se<t> pd, <v>iew current pd,
<a>dd dot to segment, add dot to <r>egion,
con<n>ect dots, <d>raw circle, <p>rint diagram,
<e>rase diagram, <m>ark radii, <c>ombine markers,
e<x>tend segment, add mar<k>ers, get <h>elp, <q>uit
```

The prompt here (`CDEG(1/1)%`) tells us that we are currently working with the first primitive diagram in a diagram array that contains 1 primitive diagram. Since we have just started the program, this is the empty primitive diagram. We can view it by typing "v":

```
CDEG(1/1)% v
```

This causes **CDEG** to open a window showing the diagram. This diagram is shown in Figure 3.16. It contains a single region bounded by the frame; **CDEG**

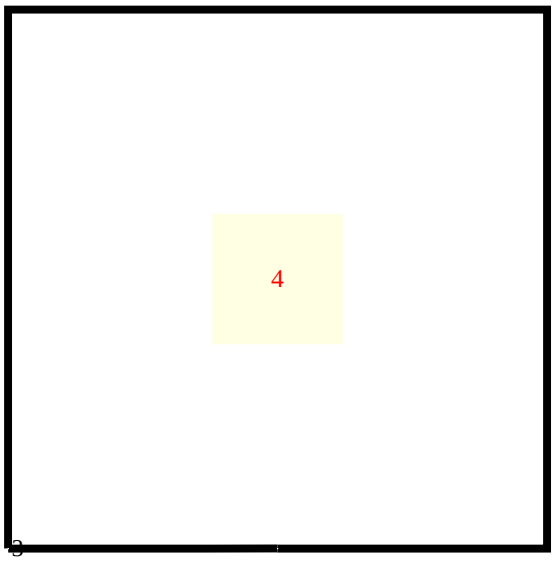


Figure 3.16: The empty primitive diagram as drawn by **CDEG**.

has assigned this region the number 4. **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: 4
CDEG(1/1)% r
Enter region number: 4
```

Now, let's look at the resulting diagram. We could use the `<v>iew` command to look at the resulting diagram, but we are going to instead use the `<p>rint diagram` command to print a text representation of the diagram. This is analogous to looking at a diagram's corresponding graph structure.

```
CDEG(1/1)% p
```

```
Diagram #1:
dot13 is surrounded by: region4
dot12 is surrounded by: region4
frame3 ends at loop in regions region4 and outerregion


region4 has boundry: frame3
        and contents:
Component #1: dot13
Component #2: dot12
```

So we see that the two new dots are numbered 12 and 13. We can connect them using the `con<n>ect dots` command.

```
CDEG(1/1)% n
Enter first dot's number: 12
Enter second dot's number: 13
CDEG(1/1)% v
```

The resulting diagram is shown in Figure 3.17. We will `<s>ave` this diagram so that we can come back to it, and then `<d>raw a circle` centered at dot 12 and going through dot 13.

```
CDEG(1/1)% s
Enter file name: seg.cd
CDEG(1/1)% d
Enter center dot's number: 12
Enter radius dot's number: 13
CDEG(1/1)% v
```

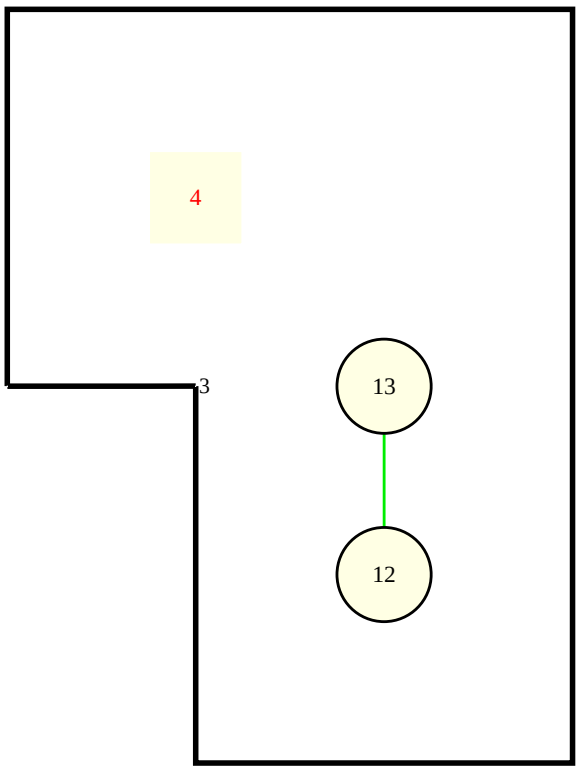


Figure 3.17: A **CDEG** diagram showing a single line segment.

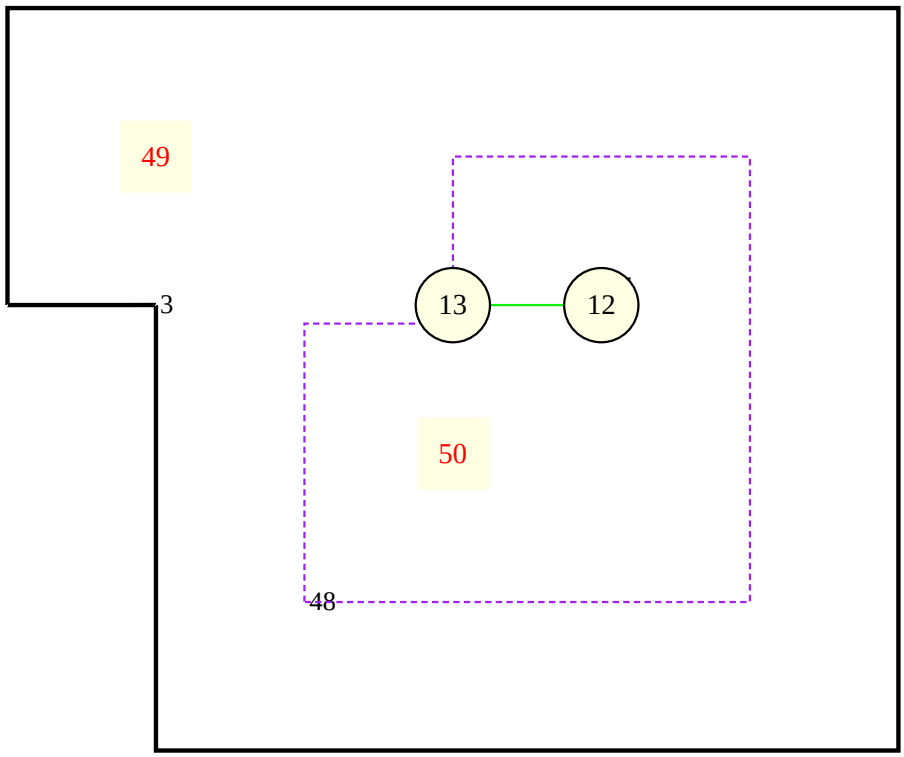


Figure 3.18: A **CDEG** diagram showing the second step in the proof of Euclid's First Proposition.

The resulting diagram is shown in Figure 3.18. The resulting dcircle is not at all circular, but all we care about here is the topology of the diagram. Next, we want to draw another circle centered at dot 13 and going through dot 12.

```
CDEG(1/1)% d
Enter center dot's number: 13
Enter radius dot's number: 12
CDEG(1/1)% v
```

This diagram is shown in Figure 3.19. Note that the segments in this diagram are colored differently depending on which line or circle they are part of.

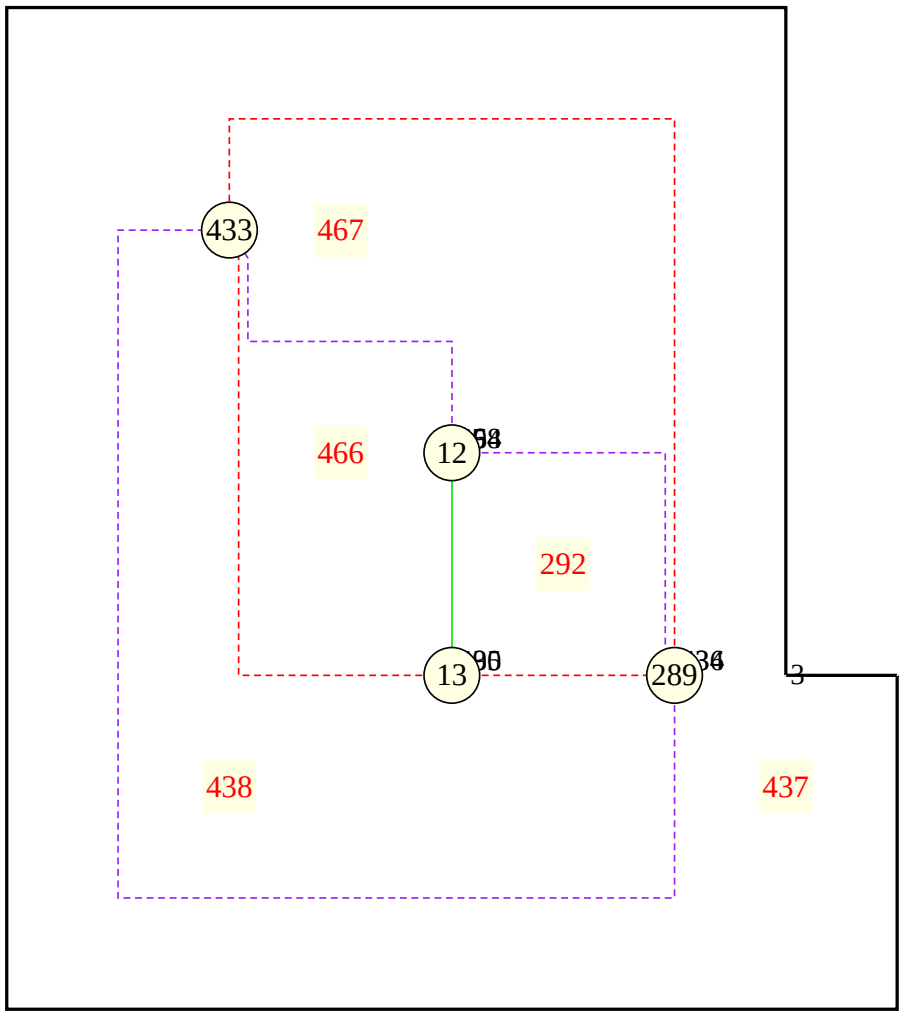


Figure 3.19: A **CDEG** diagram showing the third step in the proof of Euclid's First Proposition.

Next, we will form a triangle by connecting the endpoints of the segment to one of the points, dot number 289, on the intersection of the two circles.

```
CDEG(1/1)% n
Enter first dot's number: 12
Enter second dot's number: 289
CDEG(1/1)% n
Enter first dot's number: 13
Enter second dot's number: 289
CDEG(1/1)% v
```

The resulting diagram is shown in Figure 3.20. If we print out this much more complicated diagram, it looks like this:

```
CDEG(1/1)% p
Diagram #1:
dot433 is surrounded by: dottedseg434 region437 dottedseg436
     region438 dottedseg435 region466 dottedseg468 region467
dot289 is surrounded by: dottedseg290 region438 dottedseg436
     region437 dottedseg434 region467 dottedseg294 region1461
 solid1462 region1762 solid1763 region1761
dot13 is surrounded by: region466 dottedseg435 region438
     dottedseg290 region1761 solid1763 region1762 solid15
dot12 is surrounded by: region1762 solid1462 region1461
     dottedseg294 region467 dottedseg468 region466 solid15
solid1763 ends at dots dot13 and dot289
solid1462 ends at dots dot12 and dot289
```

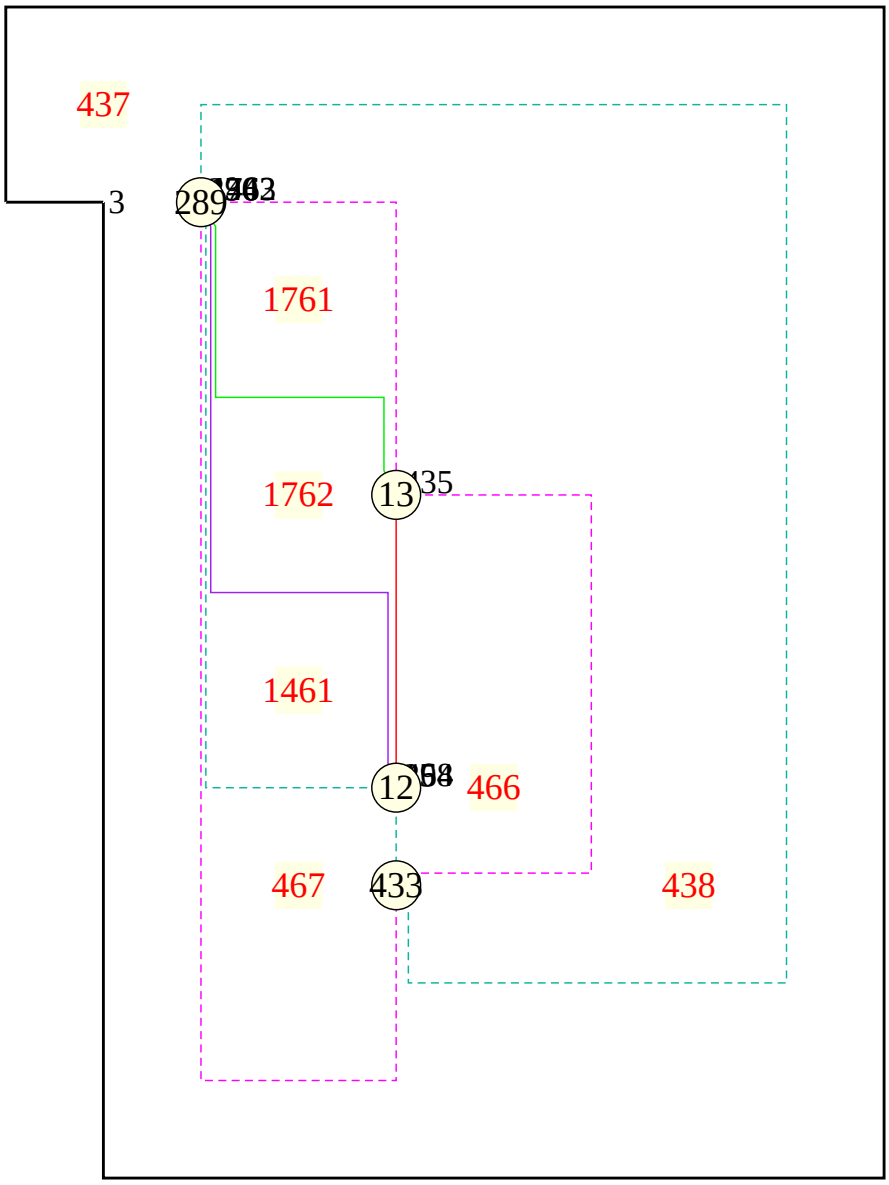


Figure 3.20: A **CDEG** diagram showing the triangle obtained in the proof of Euclid's First Proposition.

```
dottedseg468 ends at dots dot433 and dot12

dottedseg436 ends at dots dot289 and dot433

dottedseg434 ends at dots dot289 and dot433

dottedseg435 ends at dots dot433 and dot13

dottedseg294 ends at dots dot12 and dot289

dottedseg290 ends at dots dot13 and dot289

solid15 ends at dots dot12 and dot13

frame3 ends at loop in regions region437 and outerregion

dline1463 is made up of dot289 solid1763 dot13

dline484 is made up of dot289 solid1462 dot12

dline14 is made up of dot13 solid15 dot12

circle87 has center dot13 and boundry dottedseg468 dot433
     dottedseg436 dot289 dottedseg294 dot12

circle23 has center dot12 and boundry dottedseg290 dot289
     dottedseg434 dot433 dottedseg435 dot13


region1761 has boundry: solid1763 dot13 dottedseg290 dot289
        and contents:


region1762 has boundry: solid1763 dot289 solid1462 dot12 solid15
     dot13
        and contents:


region1461 has boundry: solid1462 dot289 dottedseg294 dot12
        and contents:
```

```
region466 has boundry: dottedseg468 dot433 dottedseg435 dot13
     solid15 dot12
        and contents:

region467 has boundry: dottedseg468 dot12 dottedseg294 dot289
     dottedseg434 dot433
        and contents:

region438 has boundry: dottedseg436 dot289 dottedseg290 dot13
     dottedseg435 dot433
        and contents:

region437 has boundry: frame3
        and contents:
Component #1: dottedseg436 dot433 dottedseg434 dot289
```

Note that each dot lists the regions and segments that surround it in clockwise order; each segment lists its endpoints; each line and circle lists the dots and segments that make it up; and each region lists the segments and dots that are found around its boundary in clockwise order, and the segments and dots that make up the boundary of any connected components that are found inside the region.

Next, we want to mark segment 1763 congruent to segment 15. We can do this using the `<m>ark radii` command. This command lets us mark congruent

two radii of the same circle; in order to use it, we must identify the circle that the radii are part of by identifying one of the segments that make it up. The radii are given as a list of the segments that make them up (since one radius may be made up of several diagrammatic pieces). **CDEG** checks to make sure that the given segments are in fact radii of the specified circle before it marks them; if they aren't, it returns an error message.

```
CDEG(1/1)% m
Enter the number of one seg on the circle: 468
Enter first radius dseg number:
Enter next seg index, or 0 to quit:1763
Enter next seg index, or 0 to quit:0
Enter second radius dseg number:
Enter next seg index, or 0 to quit:15
Enter next seg index, or 0 to quit:0
```

Similarly, we can mark segment 1462 congruent to segment 15 because they are both radii of the other circle.

```
CDEG(1/1)% m
Enter the number of one seg on the circle: 435
Enter first radius dseg number:
Enter next seg index, or 0 to quit:15
Enter next seg index, or 0 to quit:0
Enter second radius dseg number:
Enter next seg index, or 0 to quit:1462
Enter next seg index, or 0 to quit:0
```

```
CDEG(1/1)% v
marker2480 marks DSeg(solid15) DSeg(solid1462)
marker2479 marks DSeg(solid1763) DSeg(solid15)
```

The diagram that is displayed here is the same as that previously displayed and shown in Figure 3.20. The congruence markings that have been added are displayed as accompanying text. Finally, we can combine these two markings using the `<c>ombine markings` command. This command takes the place of transitivity: if we have a dseg or di-angle that is marked with two different markers, we can combine them into one marker that marks everything that is marked by either marking.

```
CDEG(1/1)% c
Type of marker to combine: (choices are <s>eg or <a>ng) s
Enter dseg:

Enter next seg index, or 0 to quit:15
Enter next seg index, or 0 to quit:0

CDEG(1/1)% v
marker2480 marks DSeg(solid1462) DSeg(solid1763) DSeg(solid15)
```

Thus, we have shown how to construct an equilateral triangle on the given base.

Now, let's look at how **CDEG** handles a construction that results in an array of possibilities. We will look at the previously discussed construction shown in Figures 3.1 and 3.2. To get the starting diagram, we will load our saved diagram containing a single dseg, and add two new dots to it.

```
CDEG(1/1)% l
Enter file name: seg.cd
CDEG(1/1)% p
Diagram #1:
dot13 is surrounded by: region4 solid15
dot12 is surrounded by: region4 solid15
solid15 ends at dots dot12 and dot13
frame3 ends at loop in regions region4 and outerregion
dline14 is made up of dot13 solid15 dot12

region4 has boundry: frame3
        and contents:
Component #1: dot13 solid15 dot12 solid15

CDEG(1/1)% r
Enter region number: 4
CDEG(1/1)% r
Enter region number: 4
CDEG(1/1)% v
```

This diagram is shown in Figure 3.21.

After we connect the two new dots, the command prompt changes to indicate that the current diagram array contains 9 diagrams. We can look at each of these in turn by using the `se<t> pd` command, which controls which of the primitive diagrams in the array we are currently working with.

```
CDEG(1/1)% n
```

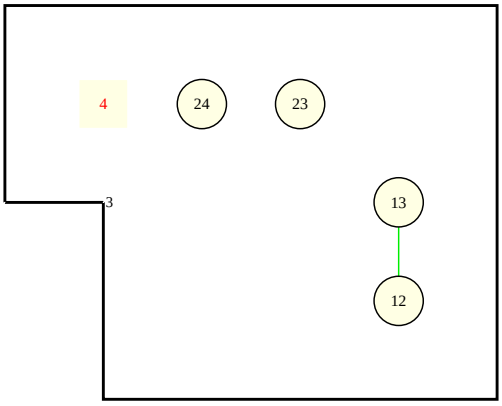


Figure 3.21: A **CDEG** diagram corresponding to the diagram shown in Figure 3.1.

```
Enter first dot's number: 23

Enter second dot's number: 24

CDEG(1/9)% v

CDEG(1/9)% t

Enter pd number: 2

CDEG(2/9)% v

CDEG(2/9)% t

Enter pd number: 3

CDEG(3/9)% v

CDEG(3/9)% t

Enter pd number: 4

CDEG(4/9)% v

CDEG(4/9)% t

Enter pd number: 5

CDEG(5/9)% v

CDEG(5/9)% t 6
```

```
Enter pd number: 6

CDEG(6/9)% v

CDEG(6/9)% t

Enter pd number: 7

CDEG(7/9)% v

CDEG(7/9)% t 8

Enter pd number: 8

CDEG(8/9)% v

CDEG(8/9)% t

Enter pd number: 9

CDEG(9/9)% v

CDEG(9/9)% q

Are you sure you want to quit? yes

 Bye!
```

These diagrams are shown in Figures 3.22 and 3.23; they are the same diagrams that were shown in Figure 3.2.

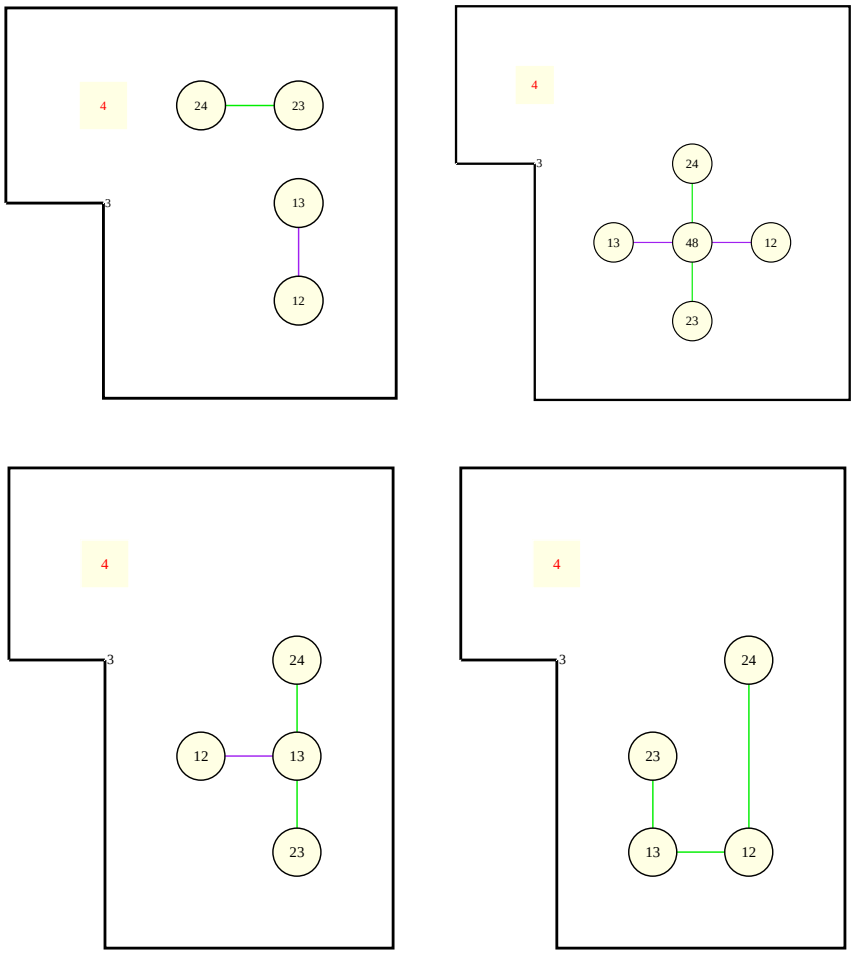


Figure 3.22: Four of the **CDEG** diagrams corresponding to those in Figure 3.2.

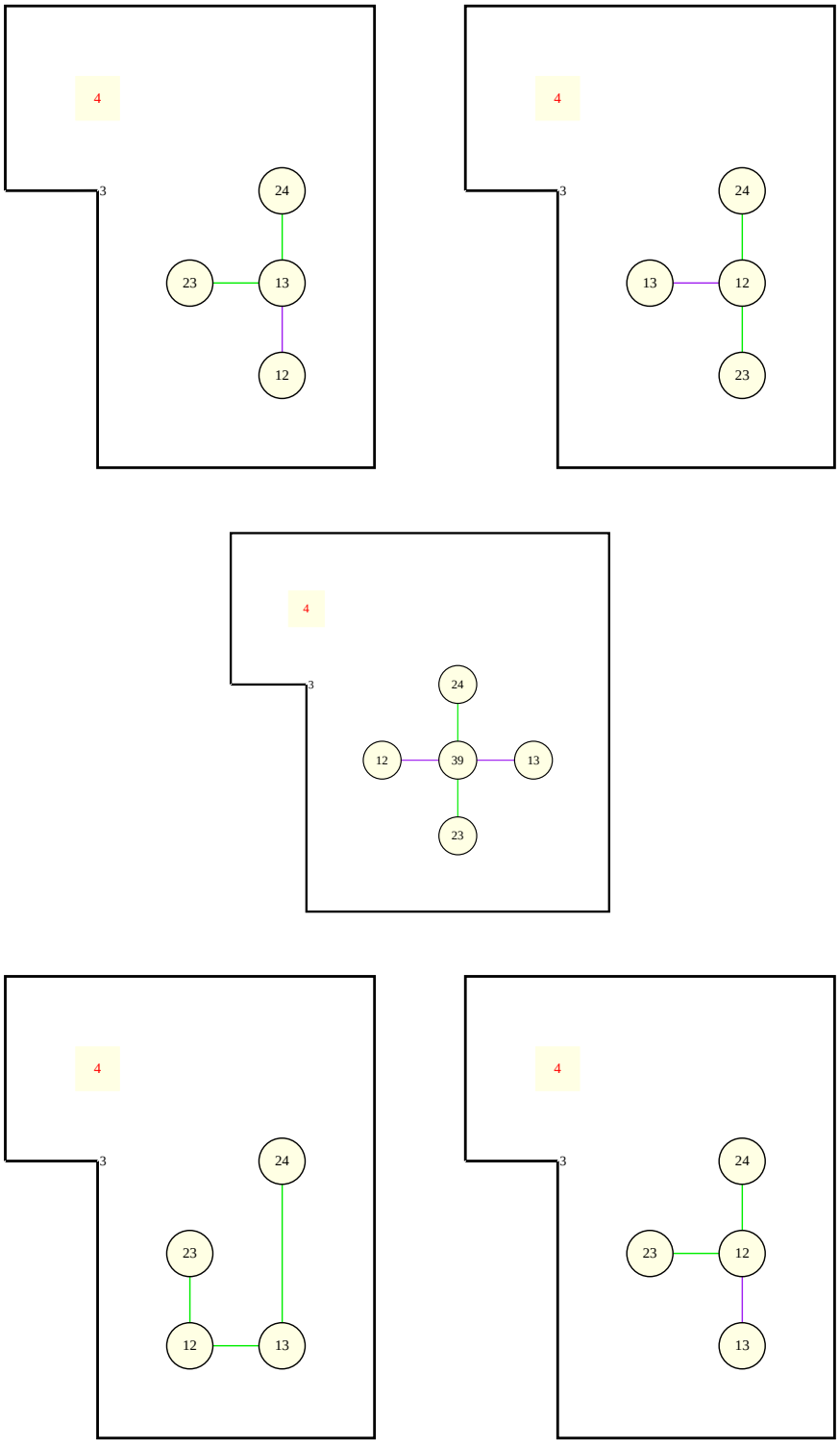


Figure 3.23: Five of the **CDEG** diagrams corresponding to those in Figure 3.2.

Chapter 4

Complexity of Diagram Satisfaction

4.1 Satisfiable and Unsatisfiable Diagrams

In Section 2.3, we showed that only nicely well-formed primitive diagrams have models. This isn't surprising, since the definition of a nicely well-formed primitive diagram was designed to eliminate diagrams that represented unrealizable situations. A more interesting question might be: how well did our definitions succeed at eliminating these unrealizable situations? That is: did we succeed in eliminating *all* of the unsatisfiable diagrams, or are there still some nwfpds with no models?

Unfortunately, the answer is that there are indeed unsatisfiable nwfpds. First of all, if we allow marked diagrams, it is easy to find examples of unsatisfiable diagrams. For example, any diagram that can be eliminated using inference rules R5a and R5b (CS and CA) is unsatisfiable. Another example is given by a diagram that contains a circle with a radius that is marked congruent to a (different) diameter; and there are many others. However, even unmarked diagrams may not

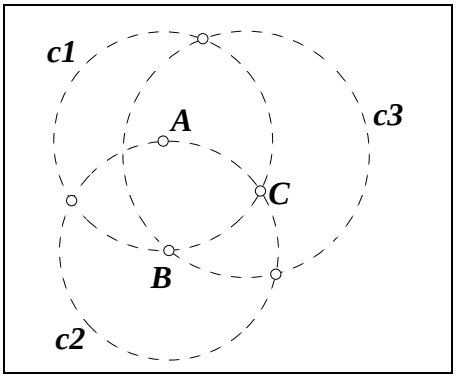


Figure 4.1: An unsatisfiable nwfpd.

be satisfiable. Figure 4.1 gives an example of such a nwfpd that is unsatisfiable: it follows from the diagram that $\overline{AB} \cong \overline{AC}$, since both segments are radii of $c1$, and similarly $\overline{AB} \cong \overline{BC}$, because both of these segments are radii of $c2$; so by transitivity, we should have $\overline{AC} \cong \overline{BC}$; but according to the diagram, B is on $c3$ and A isn't, so $\overline{AC} \ncong \overline{BC}$ and the diagram is unsatisfiable. The problem with this diagram seems to be that lengths have snuck in here via circles, so the diagram isn't just showing topological information; it's also showing geometric information about which line segments are the same length.

One might next hypothesize that any nwfpd that contains neither dcircles nor dlines that aren't proper has a model, but this isn't true either. Figure 4.2 shows such a nwfpd without a model. The easiest way to see that this diagram isn't satisfiable is to look at the rectangle in the center of the diagram. There are two crossing lines that go through the corners of this rectangle, and two other crossing lines that are copies of these that have been parallel transported downward along the sides of the rectangle from the upper corners to the lower corners. So the point of intersection of the crossing lines has also been moved down parallel to the sides of the rectangle. The line that goes through the original point of intersection and

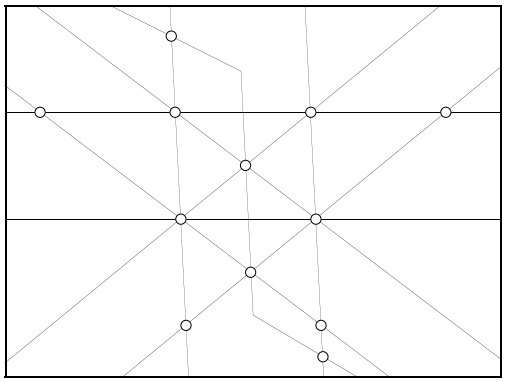


Figure 4.2: Another.

the transported point of intersection should therefore be parallel to the sides of the rectangle; but it intersects them, so the diagram isn't satisfiable. This shows that lengths can also sneak in via parallel lines.

Finally, one might hypothesize that at least any unmarked primitive diagram that doesn't contain any circles or parallel lines should be satisfiable, but even this isn't true. Figure 4.3 shows a nwfpd which only contains line segments, and which is nevertheless unsatisfiable because according to Desargues' theorem, in any model of this diagram, line XY would have to intersect line $B'C'$ at point Z. The usual proof of Desargues' theorem is as follows: imagine that the diagram shows a two-dimensional projection of a three-dimensional picture of a pyramid with base ABC and summit vertex E. Then the triangles ABC and $A'B'C'$ determine two different planes P_1 and P_2 in three-space. Lines AB and $A'B'$ meet in three-space, because they both lie in the plane determined by triangle ABE, and since AB lies in P_1 and $A'B'$ lies in P_2, their point of intersection X must lie in the intersection of P_1 and P_2. Likewise, if Y is the intersection of AC and $A'C'$ and Z is the intersection of BC and $B'C'$, then Y and Z also lie in the intersection of the two planes. Since two planes intersect in a line, this means that X, Y, and Z should

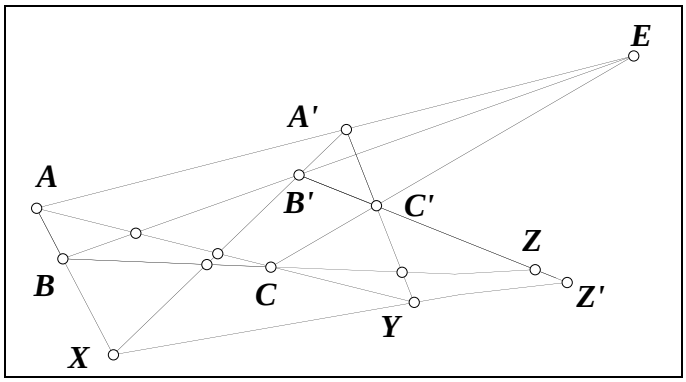


Figure 4.3: An unsatisfiable nwfpd containing nothing but unmarked dsegs.

be collinear; but in the given diagram, point Z doesn't fall on line the line XY, so the diagram is unsatisfiable.

The preceding examples show that our definition of what it means to be nicely well-formed is still too broad, because there are diagrams that are nicely well-formed but are still unsatisfiable. An obvious next question is: is there some additional set of conditions that can be added to those for nice well-formedness that will eliminate all of the unsatisfiable diagrams? It would be extremely convenient to find such a set of conditions. For example, consider what happens when we apply one of the construction rules to a satisfiable diagram. We get back an array of possible results. As it now stands, we know that because the construction rules are sound, at least one of the diagrams that we get back must be satisfiable, but many of them may not be satisfiable. If we could find a set of conditions that eliminated these unsatisfiable diagrams, then we wouldn't have to waste our time looking at these extra cases. So such a set of conditions would be extremely powerful.

The very fact that such a set of conditions would be so powerful might make us suspect they would be *too* powerful, and that such a set of conditions is impossible

to find. But somewhat surprisingly, it is possible to compute whether or not a given diagram is satisfiable. In section 4.2, we will show how to translate our definition of satisfiability into the first-order language of real arithmetic, so that given a diagram, we can find a corresponding sentence that is true if and only if the given diagram is satisfiable. We can then apply Tarski's Theorem, which says that there is a procedure for deciding if a given sentence of the first-order language of arithmetic is true or false (as a statement about the real numbers). (See [24].) This means that we could define a primitive diagram to be ***strongly nicely well-formed*** if it is nicely well-formed and Tarski's decision procedure says that it is satisfiable. Then the strongly nicely well-formed diagrams would exactly capture the possible configurations of real Euclidean planes. The problem with this approach is that the decision procedure given by Tarski's theorem can take intractably long to run. A set of conditions that correctly determine if a diagram is satisfiable but take exponentially long to evaluate aren't really useful.

So a new question is: is there a procedure that determines whether or not a given diagram is satisfiable in a reasonable amount of time? A procedure is usually considered to be tractable in the real world only if it runs in polynomial time. So, to be more specific, our question becomes: is there a polynomial-time algorithm for determining whether or not a given diagram is satisfiable? It turns out that the answer to this question is no, assuming that P $\neq$ NP as is widely believed to be the case. In the section 4.3 we will show that the diagram satisfiability problem is NP-hard, which means that no set of conditions that can be evaluated in polynomial time can determine if a given diagram is satisfiable.

4.2 Defining Diagram Satisfaction in First-Order Logic

Let's see how we can define diagram satisfaction in the theory of real arithmetic. Given a primitive diagram D that contains dots $d_1, \ldots, d_n$, dlines $l_1, \ldots, l_p$, and dcircles $c_1, \ldots, c_k$, such that all of D's dlines are proper, such that there are at least two dots on every dline and dcircle in D, and such that there is at least one dot on every ray coming out of each marked angle in D, we will define a formula of the language of real arithmetic

$$\mathrm{CF}_D(x_1, y_1, \ldots, x_n, y_n, m_1, b_1, \ldots, m_p, b_p, c_{x1}, c_{y_1}, r_1, \ldots, c_{xk}, c_{y_k}, r_k)$$

which is satisfiable over the real numbers iff D is satisfiable. This formula will be called D's ***corresponding formula***. It will be constructed so that $x_1, \ldots, r_k$ will satisfy CF iff the Euclidean plane containing the designated points (x_1, y_1) , ..., (x_n, y_n), the lines satisfying the equations $y = m_1x + b_1$, ..., $y = m_px + b_p$, and the circles satisfying the equations $(x - c_{x1})^2 + (y - c_{y_1})^2 = r_1^2$, ..., $(x - c_{xk})^2 + (y - c_{y_k})^2 = r_k^2$ satisfies D.

We will build up this formula from many simpler formulas. First of all, we are going to want the points (x_1, y_1) to be distinct and the r_i to be positive, so we define formulas that say this:

$$\begin{array}{lcl} \mathrm{DISTINCT}(x_1, \ldots, r_k) & := & \bigwedge_{i \neq j}((x_i \neq x_j) \vee (y_i \neq y_j)) \\ \mathrm{POSR}(x_1, \ldots, r_k) & := & \bigwedge_{i=1}^{k} r_i \geq 0 \end{array} .$$

We also need to make sure that every pair of circles and/or lines intersects only at

designated points. In order to do this, we must first define several other predicates.

$$\begin{aligned}
&\text{LC1INT}(m, b, c_x, c_y, r) := \\
&\qquad r^2(m^2+1) = m^2c_x^2 - 2mc_yc_x + 2mbc_x - 2bc_y + c_y^2 + b^2 \\
&\text{LC2INT}(m, b, c_x, c_y, r) := \\
&\qquad r^2(m^2+1) > m^2c_x^2 - 2mc_yc_x + 2mbc_x - 2bc_y + c_y^2 + b^2
\end{aligned}$$

LC1INT and LC2INT hold if the given line intersect the given circle exactly once or twice respectively. This is because a formula from analytic geometry tells us that the distance between the point (x_0, y_0) and the line $y = mx + b$ is given by

$$\frac{|x_0 - y_0 + b|}{\sqrt{m^2+1}}.$$

So LC1INT says that the distance between the given line and the center of the given circle is equal to the radius of the circle, while LC2INT says that it is less than the radius of the circle.

$$\begin{aligned}
&\text{CC1INT}(c_{x1}, c_{y_1}, r_1, c_{x2}, c_{y_2}, r_2) := \\
&\qquad [(c_{x1} - c_{x2})^2 + (c_{y_1} - c_{y_2})^2 = r_1^2 + r_2^2 + 2r_1r_2] \vee \\
&\qquad [(c_{x1} - c_{x2})^2 + (c_{y_1} - c_{y_2})^2 = r_1^2 + r_2^2 - 2r_1r_2] \\
&\text{CC2INT}(c_{x1}, c_{y_1}, r_1, c_{x2}, c_{y_2}, r_2) := \\
&\qquad [(c_{x1} - c_{x2})^2 + (c_{y_1} - c_{y_2})^2 > r_1^2 + r_2^2 + 2r_1r_2] \wedge \\
&\qquad [(c_{x1} - c_{x2})^2 + (c_{y_1} - c_{y_2})^2 < r_1^2 + r_2^2 - 2r_1r_2]
\end{aligned}$$

CC1INT and CC2INT hold if two given circles intersect once or twice respectively. CC1INT says that the distance between the centers of the given circles is equal to either the sum or the difference of their radii, while CC2INT says that it is between the sum and the difference of the radii.

We are now in a position to define the predicates that say that any intersections

of lines and/or circles occur only at one of the designated points.

$$\begin{aligned}
&\text{LLINT}(m_{j_1}, b_{j_1}, m_{j_2}, b_{j_2}) := \\
&\qquad (m_{j_1} = m_{j_2}) \vee \\
&\qquad \textstyle\bigvee_{i=1}^{n}((m_{j_1}x_i + b_{j_1} = y_i) \wedge (m_{j_2}x_i + b_{j_2} = y_i))
\end{aligned}$$

LLINT says that if the two given lines aren't parallel, then one of the given points lies on both of them, *i.e.*, at their point of intersection.

$$\begin{aligned}
&\text{LCINT}(m, b, c_x, c_y, r) := \\
&\qquad (\text{LC1INT}(m, b, c_x, c_y, r) \rightarrow \\
&\qquad\qquad \textstyle\bigvee_{i=1}^{n}((mx_i + b = y_i) \wedge ((x_i - c_x)^2 + (y_i - c_y)^2 = r))) \\
&\qquad \wedge (\text{LC2INT}(m, b, c_x, c_y, r) \rightarrow \\
&\qquad\qquad \textstyle\bigvee_{i \neq j}((mx_i + b = y_i) \wedge ((x_i - c_x)^2 + (y_i - c_y)^2 = r) \\
&\qquad\qquad\quad \wedge (mx_j + b = y_j) \wedge ((x_j - c_x)^2 + (y_j - c_y)^2 = r)))
\end{aligned}$$

$$\begin{aligned}
&\text{CCINT}(c_{x1}, c_{y_1}, r_1, c_{x2}, c_{y_2}, r_2) := \\
&\qquad (\text{CC1INT}(c_{x1}, c_{y_1}, r_1, c_{x2}, c_{y_2}, r_2) \rightarrow \\
&\qquad\qquad \textstyle\bigvee_{i=1}^{n}((x_i - c_{x1})^2 + (y_i - c_{y_1})^2 = r_1) \\
&\qquad\qquad\quad \wedge ((x_i - c_{x2})^2 + (y_i - c_{y_2})^2 = r_2))) \\
&\qquad \wedge (\text{CC2INT}(c_{x1}, c_{y_1}, r_1, c_{x2}, c_{y_2}, r_2) \rightarrow \\
&\qquad\qquad \textstyle\bigvee_{i \neq j}((x_i - c_{x1})^2 + (y_i - c_{y_1})^2 = r_1) \\
&\qquad\qquad\quad \wedge ((x_i - c_{x2})^2 + (y_i - c_{y_2})^2 = r_2) \\
&\qquad\qquad\quad \wedge ((x_j - c_{x1})^2 + (y_j - c_{y_1})^2 = r_1) \\
&\qquad\qquad\quad \wedge ((x_j - c_{x2})^2 + (y_j - c_{y_2})^2 = r_2))).
\end{aligned}$$

LCINT says that if the line and circle intersect once, then one of the given points lies on both of them, and if they intersect twice, then two of the listed points lie on both of them; and CCINT says the same thing for two circles. We

can now define the tuple $(x_1, \ldots, r_k)$ to be ***well-formed*** if its points are distinct, its r's are positive, and all its points of intersection between circles and lines are given by listed points; that is, if it satisfies

$$\begin{aligned}
\mathrm{WF}(x_1, \ldots, r_k) \;:=\; & \mathrm{DISTINCT}(x_1, \ldots, r_k) \wedge \mathrm{POSR}(x_1, \ldots, r_k) \\
& \wedge \textstyle\bigwedge_{i \neq j} \mathrm{LLINT}(m_i, b_i, m_j, b_j) \\
& \wedge \textstyle\bigwedge_{i,j} \mathrm{LCINT}(m_i, b_i, c_{xj}, c_{yj}, r_j) \\
& \wedge \textstyle\bigwedge_{i \neq j} \mathrm{CCINT}(c_{xi}, c_{yi}, r_i, c_{xj}, c_{yj}, r_j).
\end{aligned}$$

Next, we want to say that the given points, lines, and circles have the right graph structure. First, we define predicates that say that a point lies on a line, on a circle, or at the center of a circle.

$$\begin{aligned}
\mathrm{ONLINE}(x, y, m, b) &:= y = mx + b \\
\mathrm{ONCIRC}(x, y, c_x, c_y, r) &:= (x - c_x)^2 + (y - c_y)^2 = r^2 \\
\mathrm{ISCENT}(x, y, c_x, c_y, r) &:= (x = c_x) \wedge (y = c_y)
\end{aligned}$$

We now want to make sure that the points occur in the right order. For this purpose, we use two predicates that say that two points are adjacent to one another on a given line and that one point follows another in the clockwise direction on a given circle:

$$\begin{aligned}
&\mathrm{ADJONLINE}(x_{j_1}, y_{j_1}, x_{j_2}, y_{j_2}, m, b) := \\
&\qquad \mathrm{ONLINE}(x_{j_1}, y_{j_1}, m, b) \wedge \mathrm{ONLINE}(x_{j_2}, y_{j_2}, m, b) \\
&\qquad \wedge \textstyle\bigwedge_{i \neq j_1, j_2} (\mathrm{ONLINE}(x_i, y_i, m, b) \\
&\qquad\qquad \rightarrow (\neg((x_{j_1} < x_i < x_{j_2}) \vee (x_{j_2} < x_i < x_{j_1}))))
\end{aligned}$$

and

$$\begin{aligned}
&\text{CADJONCIRC}(x_{j_1}, y_{j_1}, x_{j_2}, y_{j_2}, c_x, c_y, r) := \\
&\qquad \text{ONCIRC}(x_{j_1}, y_{j_1}, c_x, c_y, r) \wedge \text{ONCIRC}(x_{j_2}, y_{j_2}, c_x, c_y, r) \\
&\qquad \wedge [(y_{j_1} \geq c_y \wedge y_{j_2} \geq c_y \wedge x_{j_1} < x_{j_2}) \rightarrow \\
&\qquad\qquad \textstyle\bigwedge_{i \neq j_1, j_2}((\text{ONCIRC}(x_i, y_i, c_x, c_y, r) \wedge y_i \geq c_y) \\
&\qquad\qquad\qquad \rightarrow (\neg((x_{j_1} < x_i < x_{j_2}))))] \\
&\qquad \wedge [(y_{j_1} \geq c_y \wedge y_{j_2} \geq c_y \wedge x_{j_2} < x_{j_1}) \rightarrow \\
&\qquad\qquad \textstyle\bigwedge_{i \neq j_1, j_2}(((\text{ONCIRC}(x_i, y_i, c_x, c_y, r) \\
&\qquad\qquad\qquad \rightarrow ((x_{j_1} < x_i < x_{j_2}) \wedge y_1 > c_y))))] \\
&\qquad \wedge [(y_{j_1} \leq c_y \wedge y_{j_2} \leq c_y \wedge x_{j_2} < x_{j_1}) \rightarrow \\
&\qquad\qquad \textstyle\bigwedge_{i \neq j_1, j_2}((\text{ONCIRC}(x_i, y_i, c_x, c_y, r) \wedge y_i \leq c_y) \\
&\qquad\qquad\qquad \rightarrow (\neg((x_{j_2} < x_i < x_{j_1}))))] \\
&\qquad \wedge [(y_{j_1} \leq c_y \wedge y_{j_2} \leq c_y \wedge x_{j_2} > x_{j_1}) \rightarrow \\
&\qquad\qquad \textstyle\bigwedge_{i \neq j_1, j_2}(((\text{ONCIRC}(x_i, y_i, c_x, c_y, r) \\
&\qquad\qquad\qquad \rightarrow ((x_{j_1} < x_i < x_{j_2}) \wedge y_1 < c_y))))] \\
&\qquad \wedge [(y_{j_1} < c_y \wedge y_{j_2} > c_y) \rightarrow \\
&\qquad\qquad \textstyle\bigwedge_{i \neq j_1, j_2}(\text{ONCIRC}(x_i, y_i, c_x, c_y, r) \\
&\qquad\qquad\quad \rightarrow (\neg(x_i < x_{j_1} \wedge y_i < c_y) \wedge \neg(x_i < x_{j_2} \wedge j_i > c_y)))] \\
&\qquad \wedge [(y_{j_1} > c_y \wedge y_{j_2} < c_y) \rightarrow \\
&\qquad\qquad \textstyle\bigwedge_{i \neq j_1, j_2}(\text{ONCIRC}(x_i, y_i, c_x, c_y, r) \\
&\qquad\qquad\quad \rightarrow (\neg(x_i > x_{j_1} \wedge y_i > c_y) \wedge \neg(x_i > x_{j_2} \wedge j_i < c_y)))].
\end{aligned}$$

Now we are in a position to write down a formula saying that the points, lines, and circles have the right graph structure. (Actually, it says something slightly stronger, since any points on circles must lie in the right orientation.) First we

define the sets

$$\mathrm{ADJ}_x = \{j_1, j_2 | d_{j_1}, d_{j_2} \text{ are adjacent on } x\},$$

where x can either be one of the lines l_i or one of the circles c_i. Then we can write the desired formula as follows:

$$\begin{aligned}
&\mathrm{GF}_D(x_1, \ldots, r_k) := \\
&\quad [\textstyle\bigwedge_{i=1}^{p} \bigwedge_{\{j|d_j \text{ lies on } l_i\}} \mathrm{ONLINE}(x_j, y_j, m_j, b_j))] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{p} \bigwedge_{\{j|d_j \text{ doesn't lie on } l_i\}} \neg\mathrm{ONLINE}(x_j, y_j, m_j, b_j)] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{k} \bigwedge_{\{j|d_j \text{ lies on } c_i\}} \mathrm{ONCIRC}(x_j, y_j, c_{xj}, c_{yj}, r_j))] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{k} \bigwedge_{\{j|d_j \text{ doesn't lie on } c_i\}} \neg\mathrm{ONCIRC}(x_j, y_j, c_{xj}, c_{yj}, r_j)] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{k} \bigwedge_{\{j|d_j \text{ is the center of } c_i\}} \mathrm{ISCENT}(x_j, y_j, c_{xj}, c_{yj}, r_j))] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{k} \bigwedge_{\{j|d_j \text{ isn't the center of } c_i\}} \neg\mathrm{ISCENT}(x_j, y_j, c_{xj}, c_{yj}, r_j)] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{p} \bigwedge_{(j_1,j_2)\in \mathrm{ADJ}_{l_i}} \mathrm{ADJONLINE}(x_{j_1}, y_{j_1}, x_{j_2}, y_{j_2}, m_j, b_j))] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{p} \bigwedge_{(j_1,j_2)\notin \mathrm{ADJ}_{l_i}} \neg\mathrm{ADJONLINE}(x_{j_1}, y_{j_1}, x_{j_2}, y_{j_2}, m_i, b_i)))] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{p} \bigwedge_{(j_1,j_2)\in \mathrm{ADJ}_{c_i}} \mathrm{ADJONCIRC}(x_{j_1}, y_{j_1}, x_{j_2}, y_{j_2}, c_{xi}, c_{y_i}, r_i))] \wedge \\
&\quad [\textstyle\bigwedge_{i=1}^{p} \bigwedge_{(j_1,j_2)\notin \mathrm{ADJ}_{c_i}} \neg\mathrm{ADJONCIRC}(x_{j_1}, y_{j_1}, x_{j_2}, y_{j_2}, c_{xi}, c_{y_i}, r_i))].
\end{aligned}$$

We also need to make sure that all the points lie in the correct regions. To do this, we use the following formulas:

$$\begin{aligned}
\mathrm{INCIRC}(x, y, c_x, c_y, r) &:= (x - c_x)^2 + (y - c_y)^2 < r^2 \\
\mathrm{OUTCIRC}(x, y, c_x, c_y, r) &:= (x - c_x)^2 + (y - c_y)^2 > r^2 \\
\mathrm{CW}(l_{x1}, l_{y_1}, l_{x2}, l_{y_2}, x, y) &:= \\
&\quad (l_{x2} - l_{x1})(y - l_{y_1}) < (x - l_{x1})(l_{y_2} - l_{y_1}) \\
\mathrm{CCW}(l_{x1}, l_{y_1}, l_{x2}, l_{y_2}, x, y) &:= \\
&\quad (l_{x2} - l_{x1})(y - l_{y_1}) > (x - l_{x1})(l_{y_2} - l_{y_1}).
\end{aligned}$$

INCIRC and OUTCIR say that the given point is inside or outside of the given circle. CW and CCW say that the point (x, y) lies on the clockwise or counter-

clockwise side of the directed line from (l_{x1}, l_{y_1}) to (l_{x2}, l_{y_2}). This meaning of the formulas follows from the geometric meaning of the cross product. CW says that the z component of the cross product of the vector from (l_{x1}, l_{y_1}) to (l_{x2}, l_{y_2}) and the vector from (l_{x1}, l_{y_1}) to (x, y) is negative, and CCW says that it is positive.

We can now define a formula that says that all of the points lie in the correct regions of the diagram. Recall that we require D to have at least two different dots on each dline l_i, so we can pick $a_{i,1}$ and $a_{i,2}$ so that $d_{a_{i,1}}$ and $d_{a_{i,2}}$ both lie on l_i and they aren't equal. Define

$$\begin{aligned}
\mathrm{CWP}(i) &:= \{j | d_j \text{ lies on the clockwise side of} \\
&\qquad \text{the directed line from } d_{a_{i,1}} \text{ to } d_{a_{i,2}}\}; \\
\mathrm{CCWP}(i) &:= \{j | d_j \text{ lies on the counterclockwise side of} \\
&\qquad \text{the directed line from } d_{a_{i,1}} \text{ to } d_{a_{i,2}}\}; \\
\mathrm{INP}(i) &:= \{j | d_j \text{ lies inside } c_i\}; \text{ and} \\
\mathrm{OUTP}(i) &:= \{j | d_j \text{ lies outside } c_i\}.
\end{aligned}$$

We can now define the formula as follows:

$$\begin{aligned}
&\mathrm{CREG}_D(x_1, \ldots, r_k) := \\
&\qquad \textstyle\bigwedge_{i=1}^{p} \bigwedge_{j \in \mathrm{CWP}(i)} \mathrm{CW}(x_{a_{i,1}}, y_{a_{i,1}}, x_{a_{i,2}}, y_{a_{i,2}}, x_j, y_j) \wedge \\
&\qquad \textstyle\bigwedge_{i=1}^{p} \bigwedge_{j \in \mathrm{CCWP}(i)} \mathrm{CCW}(x_{a_{i,1}}, y_{a_{i,1}}, x_{a_{i,2}}, y_{a_{i,2}}, x_j, y_j) \wedge \\
&\qquad \textstyle\bigwedge_{i=1}^{k} \bigwedge_{j \in \mathrm{INP}(i)} \mathrm{INCIRC}(x_j, j_j, c_{xj}, c_{y_j}, r_j) \wedge \\
&\qquad \textstyle\bigwedge_{i=1}^{k} \bigwedge_{j \in \mathrm{OUTP}(i)} \mathrm{OUTCIRC}(x_j, j_j, c_{xj}, c_{y_j}, r_j).
\end{aligned}$$

The last thing that we need to do is to make sure that segments and angles that are marked congruent are really the same size. To do this, we need a predicate that says that the segment between (x_{i_1}, y_{i_1}) and (x_{i_2}, y_{i_2}) is congruent to the segment

between (x_{i_3}, y_{i_3}) and (x_{i_4}, y_{i_4}):

$$\begin{aligned}&\mathrm{CONGS}_D(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}, x_{i_3}, y_{i_3}, x_{i_4}, y_{i_4}) := \\ &\qquad ((x_{i_2} - x_{i_1})^2 + (y_{i_2} - y_{i_1})^2 = ((x_{i_4} - x_{i_3})^2 + (y_{i_4} - y_{i_3})^2.\end{aligned}$$

We also need a similar predicate that says that the angle θ_1 determined by the three points (x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), and (x_{i_3}, y_{i_3}) is congruent to the angle θ_1 determined by the points (x_{i_4}, y_{i_4}), (x_{i_5}, y_{i_5}), and (x_{i_6}, y_{i_6}). (Here, the first point gives the vertex of the angle, and the sides of the angle are rays going through the other two points, which are given in clockwise order.) First we use the fact that

$$\frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} = \cos\theta$$

for any two vectors $\mathbf{A}$ and $\mathbf{B}$ and angle θ between them to write a formula that says that $\cos^2\theta_1 = \cos^2\theta_2$, as follows:

$$\begin{aligned}&\mathrm{ECOS2}_D(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}, x_{i_3}, y_{i_3}, x_{i_4}, y_{i_4}, x_{i_5}, y_{i_5}, x_{i_6}, y_{i_6}) := \\ &\qquad ((x_{i_3} - x_{i_1})(x_{i_2} - x_{i_1}) + (y_{i_3} - y_{i_1})(y_{i_2} - y_{i_1}))^2 * \\ &\qquad ((x_{i_5} - x_{i_4})^2 + (y_{i_5} - y_{i_4})^2)((x_{i_6} - x_{i_4})^2 + (y_{i_6} - y_{i_4})^2) = \\ &\qquad ((x_{i_6} - x_{i_4})(x_{i_5} - x_{i_4}) + (y_{i_6} - y_{i_4})(y_{i_5} - y_{i_4}))^2 * \\ &\qquad ((x_{i_2} - x_{i_1})^2 + (y_{i_2} - y_{i_1})^2)((x_{i_3} - x_{i_1})^2 + (y_{i_3} - y_{i_1})^2).\end{aligned}$$

If the squares of the two cosines are equal, then the cosines are equal as long as they have the same sign; we can check this by making sure that the two dot products have the same sign:

$$\begin{aligned}&\mathrm{ECOS}_D(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}, x_{i_3}, y_{i_3}, x_{i_4}, y_{i_4}, x_{i_5}, y_{i_5}, x_{i_6}, y_{i_6}) := \\ &\qquad \mathrm{ECOS2}_D(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}, x_{i_3}, y_{i_3}, x_{i_4}, y_{i_4}, x_{i_5}, y_{i_5}, x_{i_6}, y_{i_6}) \wedge \\ &\qquad ((x_{i_3} - x_{i_1})(x_{i_2} - x_{i_1}) + (y_{i_3} - y_{i_1})(y_{i_2} - y_{i_1})) * \\ &\qquad ((x_{i_6} - x_{i_4})(x_{i_5} - x_{i_4}) + (y_{i_6} - y_{i_4})(y_{i_5} - y_{i_4})) > 0.\end{aligned}$$

Now, if the cosines of the two angles are equal, then either the angles are equal, or else they sum to 360°. So we can check that they are the same by making sure that they are both greater than or both less than 180°. We can do this by making sure that (x_{i_3}, y_{i_3}) falls on the same (clockwise or counterclockwise) side with respect to the vector from (x_{i_1}, y_{i_1}) to (x_{i_2}, y_{i_2}) as (x_{i_6}, y_{i_6}) falls with respect to the vector from (x_{i_4}, y_{i_4}) to (x_{i_5}, y_{i_5}). The following predicate accomplishes this:

$$\begin{aligned}&\text{CONGANG}_D(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}, x_{i_3}, y_{i_3}, x_{i_4}, y_{i_4}, x_{i_5}, y_{i_5}, x_{i_6}, y_{i_6}) :=\\ &\qquad \text{ECOS}_D(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}, x_{i_3}, y_{i_3}, x_{i_4}, y_{i_4}, x_{i_5}, y_{i_5}, x_{i_6}, y_{i_6}) \wedge\\ &\qquad (\text{CW}(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}, x_{i_2}, y_{i_2}) \rightarrow \text{CW}(x_{i_4}, y_{i_4}, x_{i_5}, y_{i_5}, x_{i_6}, y_{i_6})).\end{aligned}$$

We are now finally in a position to define a formula that says that all of the angles and segments marked congruent are congruent. First we need to define the following sets:

$$\begin{aligned}\text{Cong-segs}_D &:= \{(d_{i_1}, d_{i_2}, d_{i_3}, d_{i_4}) | \text{the dseg through } d_{i_1} \text{ and } d_{i_2}\\ &\qquad \text{is marked congruent to the dseg through } d_{i_3} \text{ and } d_{i_4}\}\\ \text{Cong-angs}_D &:= \{(d_{i_1}, d_{i_2}, d_{i_3}, d_{i_4}, d_{i_5}, d_{i_6}) | \text{the di-angle given by}\\ &\qquad d_{i_1}, d_{i_2}, \text{ and } d_{i_3} \text{ is marked congruent}\\ &\qquad \text{to the di-angle given by } d_{i_4}, d_{i_5}, \text{ and } d_{i_6}\}\end{aligned}$$

We can use these to define the desired formula:

$$\begin{aligned}&\text{CONG}_D(x_1, \ldots, r_k) :=\\ &\qquad [\textstyle\bigwedge_{(d_{i_1}, d_{i_2}, d_{i_3}, d_{i_4}) \in \text{Cong-segs}_D} \text{CONGS}_D(x_{i_1}, y_{i_1}, \ldots, x_{i_4}, y_{i_4})] \wedge\\ &\qquad [\textstyle\bigwedge_{(d_{i_1}, \ldots, d_{i_6}) \in \text{Cong-angs}_D} \text{CONGANG}_D(x_{i_1}, y_{i_1}, \ldots, x_{i_6}, y_{i_6})].\end{aligned}$$

Finally, we can define the originally promised formula CF as follows:

$$\begin{aligned}\text{CF}_D(x_1, \ldots, r_k) &:= \text{WF}(x_1, \ldots, r_k) \wedge \text{GF}_D(x_1, \ldots, r_k)\\ &\qquad \wedge \text{CREG}_D(x_1, \ldots, r_k) \wedge \text{CONG}_D(x_1, \ldots, r_k).\end{aligned}$$

Note the following facts: 1) $\mathrm{CF}_D(x_1, \ldots, r_k)$ is quantifier free. 2) D is a satisfiable diagram iff the existential closure of CF_D is satisfiable over the Real numbers, which is a decidable question by Tarski's theorem. (In fact, the sentence in question contains only existential quantifiers, and it is known that such sentences can be decided in polynomial space. For details, see [21].) 3) If A is a diagram array consisting of primitive diagrams $\{D_1, \ldots, D_m\}$, and each of these primitive diagrams contains only proper dlines, has at least two dots on each dline and dcircle, and has at least one dot on every ray coming out of a marked angle, then A is satisfiable iff the existential closure of $\mathrm{CF}_{D_1} \vee \cdots \vee \mathrm{CF}_{D_m}$ is satisfiable over the Reals. 4) Given any diagram array E, we can use the construction rules to find a diagram array E' whose primitive diagrams only contain proper dlines and have at least two dots on each dline and dcircle and one dot on every ray that emanates from a marked angle, such that E' is satisfiable iff E is. It follows from these facts that the general question of satisfiability of diagrams is decidable.

4.3 NP-hardness

In this section, we show that the problem of determining if a given diagram is satisfiable is NP-hard. The problem of determining if a given boolean formula is satisfiable is well known to be NP-complete, so it suffices to show how to reduce the boolean satisfiability problem to the diagram satisfiability problem in log-space. (See [11] for a proof that the boolean satisfiability problem is NP-complete.)

We will consider a ***Boolean formula*** to be a string composed of three types of symbols: boolean variables $(x_1, x_2, x_3, \ldots)$, parenthesis, and the logical operators AND ($\wedge$) and NOT ($\neg$). A string of these symbols is a boolean formula if it is a

boolean variable, in which case it is called an atomic formula, or if it is of the form $(A \wedge B)$ or $\neg A$, where A and B are boolean formulas. The ***proper subformulas*** of a formula are defined as follows: the proper subformulas of $A \wedge B$ are A, B, and the proper subformulas of A and B; the proper subformulas of $\neg A$ are A and its proper subformulas; and atomic formulas have no proper subformulas. The ***subformulas*** of F are F along with all of the proper subformulas of F. Given an assignment of truth values (true and false) to the boolean variables of a formula, the truth value of the formula can be determined as follows: if the formula is a boolean variable, then the truth value of the formula is the same as the truth value of the variable; if the formula is of the form $(A \wedge B)$, then the formula is true if and only if both of the subformulas A and B are true; and if the formula is of the form $\neg A$ then it is true just if A is false. A boolean formula is satisfiable if and only if there is an assignment of truth values to its propositional variables that makes the whole formula true.

For every boolean formula F, we can define a corresponding diagram $D(F)$ which is satisfiable if and only if F is. Let $F_1, F_2, F_3, \ldots, F_{f-1}$ be the proper subformulas of F, arranged in order of increasing complexity, so that if F_j is a subformula of F_k then $j \leq k$, and let F_f be F. For each i, $0 \leq i \leq f+1$, we are going to define a subdiagram $D_i(F)$. $D(F)$ will be a diagram that contains disjoint copies of all of these subdiagrams. In order to construct $D(F)$, we first pick $6f+8$ distinct markers: six di-angle markers a_i,b_i,g_i,e_i,h_i, and m_i for each subformula F_i of F; six other di-angle markers, labeled here by one slash mark and the names Y_1, Y_2, Z, H, and R; and two dseg markers, shown as two and three slash marks. Marker R will be used to mark right angles and will also be designated by drawing the usual right angle symbol in the diagrams. In the following discussion, the

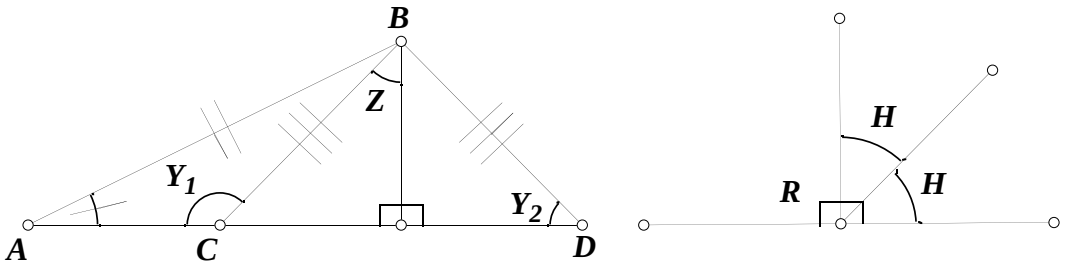


Figure 4.4: $D_0(F)$

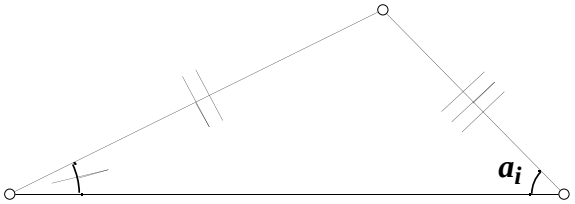


Figure 4.5: Subdiagram contained in $D_i(F)$ if F_i is an atomic formula or a conjunction.

marker names will also be used to refer to the measures of the angles that they represent; it should be clear from context which meaning is intended.

We let $D_0(F)$ be the subdiagram shown in Figure 4.4 and let $D_{f+1}(F)$ be the subdiagram shown in Figure 4.8. For $1 \leq i \leq f$, we define $D_i(F)$ as follows:

- If F_i is atomic, then $D_i(F)$ is the subdiagram shown in Figure 4.5.
- If F_i is $\neg F_j$, then $D_i(F)$ is the subdiagram shown in Figure 4.6.
- If F_i is $(F_j \wedge F_n)$, then $D_i(F)$ is the subdiagram that contains both of the subdiagrams shown in Figures 4.5 and 4.7.

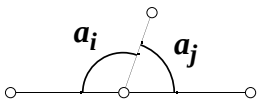


Figure 4.6: $D_i(F)$ when F_i is $\neg F_j$.

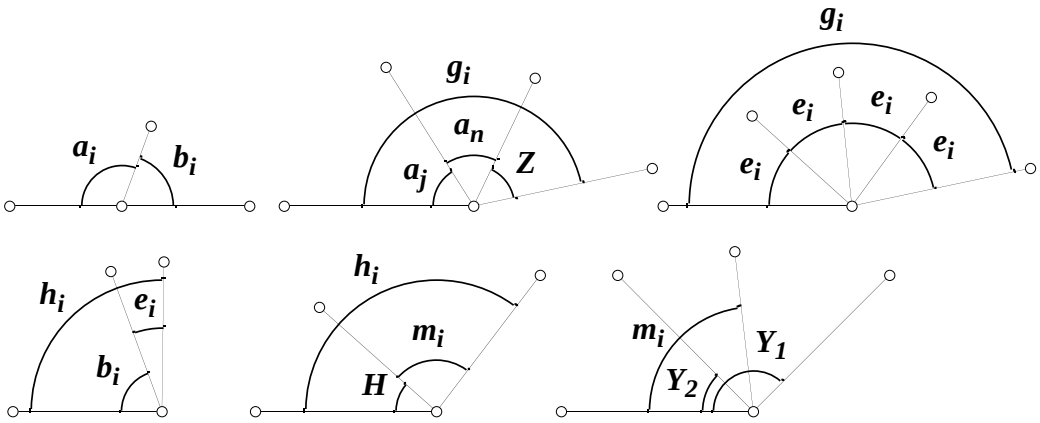


Figure 4.7: Subdiagram contained in $D_i(F)$ if F_i is $(F_j \wedge F_n)$.

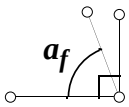


Figure 4.8: $D_{f+1}(F)$

Let $D_i^\circ(F)$ be the (smallest) diagram containing $D_0, D_1, \ldots, D_i$ as disjoint subdiagrams, and let $D(F) = D_{f+1}^\circ(F)$.

Note that if F has length n, then it has at most n subformulas, and the subdiagram put into $D(F)$ for each subformula has a size bounded by a constant, so the size of $D(F)$ is linear in the length of F. In fact, in order to compute $D(F)$ from F, the only thing that you need to keep track of is which subformula in F you are currently on, which you can do in log-space.

Now, note that Figure 4.5 along with Figure 4.4 forces angle a_i to be equal to one of Y_1 or Y_2, since there are only two triangles that can be built with the given angle and sides. (For proof, see [7, p. 304-7].) It follows by induction on the complexity of F_i that a_i must be equal either to Y_1 or else to Y_2, for every i: if F_i is atomic or a conjunction, then $D_i(F)$ contains Figure 4.5, and if it is $\neg F_j$, then a_i is supplementary to a_j, which is one of Y_1 or Y_2 by inductive hypothesis, and so

a_i is Y_2 or Y_1, since Y_1 and Y_2 are supplements. Furthermore, note that Y_2 and Z are complimentary, so $Y_2 = 90° - Z$ and $Y_1 = 90° + Z$.

We want to show these two possible values of each a_i correspond to the two possible truth values of F_i, so that $a_i < 90°$ iff F_i is true. More specifically, we will say that a model M ***agrees*** with assignment t on a subformula F_i if $a_i = Y_2$ and F_i is true under t, or if $a_i = Y_1$ and F_i is false under t. We will show that any model of any of the D_i° that agrees with a truth assignment t on the atomic subformulas of F also agrees with t on all of the other subformulas. This means that the subdiagrams in Figures 4.6 and 4.7 act as logical NOT and AND gates, so that if F_i is $(F_j \wedge F_n)$, then $a_i < 90°$ iff $a_j < 90°$ and $a_n < 90°$, and if F_i is $\neg F_j$, then $a_i < 90°$ iff $a_j \geq 90°$. This is shown in the following lemma:

Lemma 4.3.1. *Let t be a function assigning truth values to the boolean variables of F. Then:*

a. *For each $i \leq f$, there is a model M_i of $D_i^\circ(F)$ that agrees with t on all atomic subformulas F_k with $k \leq i$.*

b. *Furthermore, if M is any model of $D_i^\circ(F)$ that agrees with t on all atomic subformulas F_k with $k \leq i$, then it must also agree with t on all other subformulas F_k with $k \leq i$.*

Proof. by induction on i.

Base case: $i = 0$. In this case, we just have to show that $D_0^\circ(F)$, which is just $D_0(F)$, has a model. It has one: take any isosceles triangle, draw a perpendicular through the vertex, extend the base to one side, and connect it to the vertex to get a model of the first half of D_0. To get a model of the other half, bisect a

straight angle into two right angles, then divide one of the right angles into two equal pieces.

Inductive cases:

1. F_i is an atomic formula of the form x_j. By the inductive hypothesis, $D^\circ_{i-1}(F)$ has a model M_{i-1} that agrees with t on all subformulas F_k such that $k < i$. That model satisfies all of $D^\circ_i(F)$ except for the triangle added by $D_i(F)$. Any triangle added by $D_i(F)$ will have to be congruent to one of the two triangles ABC or ABD; conversely, any model that extends M_{i-1} and contains a new disjoint triangle which is congruent to ABC or ABD will satisfy $D^\circ_i(F)$. So if $t(x_j) = \text{true}$, let M_i be such an extension of M_{i-1} in which the new triangle is congruent to ABC, and otherwise let it be such an extension of M_{i-1} in which the new triangle is congruent to ABD. Then M_i agrees with t on F_i, which shows part a. For part b, note that any model M of $D_i(F)$ that agrees with t on atomic formulas has a submodel that is a model of D_{i-1}; so by part b of the inductive hypothesis, $a_k = Y_2$ iff F_k is true under t if $k \leq i-1$, and this is also true if $k = i$, since F_i is atomic. This shows part b.

2. F_i is a formula of the form $\neg F_j$. By the inductive hypothesis, there is a model M_{i-1} of $D^\circ_{i-1}(F)$ that agrees with t on atomic formulas (by part a), and therefore agrees with t on all subformulas F_i with $k < i$ (by part b). Let M_i be a model extending M_{i-1} that also contains a new straight angle divided into two pieces such that the clockwise piece is congruent to the other angles that are marked by a_j. Then M_i is a model of $D_i(F)$, proving part a. To show part b, note that any model M of $D_i(F)$ that agrees with t on the atomic formulas of F must agree with t on all the subformulas F_k such that $k < i$, as before, so it suffices to show that it agrees with t on F_i, that is,

that $a_i = Y_2$ in M iff F_i is true under t. Since j must be less than i (because the subformulas of F were arranged in order of increasing complexity), and a_i and a_j are supplements, $a_i = Y_2$ iff $a_j = Y_1$ iff F_j is false under t iff F_i is true under t. This proves b.

3. F_i is a formula of the form $(F_j \wedge F_n)$. We want to show that $D_i(F)$ forces a_i to be less than $90°$ iff a_j and a_n are both less than $90°$. First note that in any model of $D_i(F)$, $m_i = (a_j + a_n + Z)/4 + b_i - 45°$ (from pieces 2-5 of Figure 4.7); $Y_1 < m_i < Y_2$ (from piece 6 of Figure 4.7); and b_i is equal to either Y_1 or Y_2 ($90° + Z$ or $90° - Z$), because it is supplementary to a_i. There are three cases to consider:

   (a) $a_j = a_n = Y_2 = 90° - Z$. Then $m_i = (180° - Z)/4 + b_i - 45° = 45° - Z/4 + b_i - 45° = b_i - Z/4$. Since $90° - Z < m_i < 90° + Z$, this means that $90° - Z < b_i - Z/4 < 90° + Z$, so $90° - 3Z/4 < b_i < 90° + 5Z/4$. Since b_i is either $90° - Z$ or $90° + Z$, this means that it must be $90 + Z = Y_1$. So since a_i is supplementary to b_i, this means that $a_i = Y_2$.

   (b) $a_j = Y_1$ and $a_n = Y_2$, or $a_j = Y_2$ and $a_n = Y_1$. Then $m_i = b_i + Z/4$; so $90° - 5Z/4 < b_i < 90° + 3Z/4$ so b_i must be equal to Y_2 to keep m_i between Y_1 and Y_2; so $a_i = Y_1$.

   (c) $a_j = a_n = Y_2 = 90° + Z$. Then $m_i = b_i + 3Z/4$, and b_i must be equal to Y_2 as before, so $a_i = Y_1$.

So let M be any model of $D_i^\circ(F)$ that agrees with t on atomic formulas. By the inductive hypothesis, M agrees with t on all subformulas F_k with $k < i$. To show part b, it suffices to show that M agrees with t on F_i. Now, if F_i is true under t, then F_j and F_n must also be true under t (by the truth table for

AND), so by the inductive hypothesis, $a_j = a_n = Y_2$, which is the first case above, so $a_i = Y_2$ in M, as required. On the other hand, if F_i is false under t, then one of F_j or F_n must also be false. So, by the inductive hypothesis, one or both of a_i and a_n must be equal to Y_2. So we are in either the second or third case above, and in both of these cases, $a_i = Y_1$ in M, also as required. This proves part b. For part a, note that we can extend M_{i-1} with pieces satisfying $D_i(F)$ as long as we make a_i equal to Y_1 or Y_2 according to the three cases above, because the first five pieces of Figure 4.7 serve only to define m_i, and the last piece will be satisfied as long as that m_i lies between Y_2 and Y_1.

This proves the lemma. □

We can now prove the following theorem:

Theorem 4.3.1. *Any boolean formula F is satisfiable if and only if $D(F)$ is satisfiable.*

Proof. ($\Rightarrow$) Assume that F is satisfiable. Then there is a truth assignment t of the boolean variables of F under with F is true. So, by the lemma, there is a model M of $D^\circ_f(F)$ that agrees with t on F_i for all $i \leq f$. In particular, M agrees with t on F_f. Since $F_f = F$ and F is true under t, this means that $a_f = Y_2$ in M. The only difference between $D^\circ_f(F)$ and $D(F)$ is that $D(F)$ contains D_{f+1}(F). So it suffices to show that there is an extension of M that satisfies $D_{f+1}(F)$. But $D_{f+1}(F)$ just says that $a_f < 90°$. So since $a_f = Y_2 < 90°$ in M, M can be extended to a model M' that satisfies $D(F)$.

($\Leftarrow$) Now assume that F is unsatisfiable. Then F is false under all possible truth assignments of its boolean variables. So, by the lemma, $a_f = Y_1 > 90°$ in

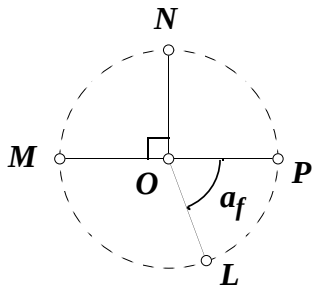


Figure 4.9: $D''_{f+1}(F)$.

any model of $D^\circ_f(F)$, because any model of $D^\circ_f(F)$ agrees with some possible truth assignment on atomic formulae. So no model of $D^\circ_f(F)$ can be extended to a model of $D(F)$, since a_f would have to be less than $90°$ in any such model because it would have to satisfy D_{f+1}. So $D(F)$ is unsatisfiable. □

We have shown how to reduce the question of whether or not a given boolean formula is satisfiable to the question of whether or not a given diagram is satisfiable, and this reduction can be done in log-space. Therefore, because the boolean satisfiability problem is NP-complete, we have the following corollary:

Corollary 4.3.1. *The diagram satisfiability problem is NP-hard.*

Next, consider the ***case analysis problem***: let D be a satisfiable primitive diagram, and let $S(D)$ be the set of satisfiable diagrams that can result from extending a given line segment in D outward until it intersects another segment. The problem of figuring out exactly what diagrams are in $S(D)$ is also NP-hard. To see this, let F be a boolean formula, let $D''_{f+1}(F)$ be the subdiagram shown in Figure 4.9, and let $D''(F)$ be the smallest diagram containing $D_0(F), D_1(F), \ldots, D_f(F)$ and $D''_{f+1}(F)$. Then F is satisfiable iff $D''(F)$ has a model in which $a_f = Y_2$, iff $D''(F)$ has a satisfiable extension in which dseg OL

is extended into di-angle MON. So since $D''(F)$ can also be produced from F in log-space, we also have:

Corollary 4.3.2. *The case analysis problem is also NP-hard.*

Chapter 5

Conclusions

What kind of implications does a system like this have for the actual practice of geometry? I think that it has several. First and foremost, it tells us the proper rules by which geometric diagrams can be carefully and correctly used. Even if we may choose to ignore some of the rules in practice (after all, it's hard to do the required case analysis by hand), at least we will have a good idea what we would have to check in order to make sure that a proof is valid. This is, after all, the same situation that holds in the rest of mathematics, in which people normally give informal proofs, checking enough details to convince themselves that the proof could be made formal. Secondly, a formalism like this shows once and for all that proofs that use diagrams are in no way inherently less rigorous than sentential proofs. The two different styles of proofs certainly have different strengths and weaknesses, but neither can lay a greater claim to inherent rigourousness. It is of course possible to prove fallacies using diagrams, but only if they are used incorrectly, just as it is possible to prove that $0 = 1$ by using algebra incorrectly. Thus, the twentieth century bias against the use of geometric diagrams can be seen for what it is—a bias, which hopefully will slowly dissipate. Then, instead of

viewing the use of a diagram in a proof as a mark of "human frailty," people can view diagrams as useful tools to be understood and used wisely. After all, people are going to use diagrams in their proofs in either case.

Another reason that a formal system like this is useful is that it helps to explain the ways in which people have traditionally used diagrams. For example, the observation that the use of lemmas can lead to an exponential decrease in the number of cases that need to be considered helps to explain why geometry has such a long history of basing proofs on collections of previously proven facts. It also provides an alternative explanation for the fact that Euclid used superposition to prove SAS and then used SAS to prove other results rather than continuing to use superposition. Many commentators have asserted that this shows that Euclid viewed superposition as being a suspect method of proof. While it is possible that this was the case, the formal system **FG** shows that proofs that use superposition can be made as rigorous as other proofs, and that there are still good reasons for preferring the use of SAS as a lemma to the direct use of superposition in general. **FG** also sheds light on the old dispute over how many different cases need to be considered in proving a theorem. Euclid's normal practice was to give the proof for a single case only, but many later commentators have pointed out that there can be other cases that need to be considered, often with corresponding changes in the proof. It has not been previously clear exactly how many cases needed to be considered, which has contributed to the idea that the use of diagrams is inherently informal. (In fact, there has sometimes been disagreement over this: see for example Thomas Heath's commentary on Euclid's proposition 2 [7, pp. 145–146], in which he discusses the ancient Greek commentator Proclus' case analysis, poking fun at his "anxiety to subdivide [cases].") The semantics of diagrams that

we have given here makes it clear that two cases are distinct and may require different proofs if they are topologically different. Case analysis aside, it is striking how similar proofs in **FG** can be to those given by Euclid. In fact, many of Euclid's proofs that have been often criticized for making unstated assumptions, such as the proof of his first proposition, turn out to look exactly the same in **FG**, because the assumptions are taken care of by the underlying diagrammatic machinery. Thus, **FG** shows that some of the aspects of Euclid's proofs that have been viewed as flaws can be viewed as correct uses of a diagrammatic method that was not fully explained.

A formal system like this also has implications for the way in which geometry is taught. Geometry and logic have long been taught together, and for good reasons: geometric proofs are relatively accessible, geometry has a long tradition of deductive reasoning from assumptions, and it is hard to see *what* is true in geometry if you don't know *why* it is true. Furthermore, geometry gives us a very natural example of a set of axioms with several different possible models, making it an ideal subject in which to present many of the ideas of modern logic, such as logical consequence, independence of axioms, and so on. However, the usual presentation of logic in high school geometry classes is in some ways counterproductive. Most high school geometry classes rely heavily on two column proofs. These proofs are really pseudo-formal, because the steps that are allowed to be taken are never fully specified in advance. This has been a necessary side effect of using diagrams in these proofs: since formal rules for working with diagrams haven't previously been well understood, it hasn't been possible to give clear rules for using them in two column proofs. Students working with such rules may see that there are correct answers that the teacher will accept, but it may not be possible for a student

to check the correctness of a given proof for him or herself; thus, these kinds of rules may end up teaching the students that in mathematics, things are true if the teacher says that they are true. A true formal system fixes this problem, because it makes it perfectly clear in advance what constitutes a solution to a given problem. A more user-friendly version of a computer system like **CDEG** would make it possible to make such a formal system available in geometry classrooms. This would also make it easier to teach students about the differences between formal proofs and informal proofs.

Finally, a system like this is enormously helpful in proving metamathematical results about geometry, as we have seen. In fact, questions like "How hard is it to determine if a given diagram is satisfiable?," "Is CS a stronger axiom than SSS?," or "Are there true facts about 2-dimensional Euclidean geometry that cannot be proven by standard Euclidean methods?" can't even be articulated clearly until we have a formalization like this. (Another way of stating the third of these questions is "Is **FG** complete?," and it is an open question.) The development of **FG** has allowed these kinds questions to be meaningfully articulated; some have been answered here, but many more remain.

Appendix A

Euclid's Postulates

For reference, Euclid's Postulates and Common Notions, along with several of his definitions from Book I of *The Elements*, are given in Tables A.1, A.2, and A.3, as translated in [7].

Table A.1: Some of Euclid's definitions from Book I of *The Elements*.

Definitions

10. When a straight line set up on a straight line makes the adjacent angles equal to one another, each of the equal angles is ***right***, and the straight line standing on the other is called a ***perpendicular*** to that on which it stands.

15. A ***circle*** is a plane figure contained by one line such that the straight lines falling upon it from one point among those lying within the figure are equal to one another;

16. And the point is called the ***centre*** of the circle.

Table A.2: Euclid's Postulates from *The Elements*.

Postulates

Let the following be postulated:

1. To draw a straight line from any point to any point.

2. To produce a finite straight line continuously in a straight line.

3. To describe a circle with any centre and distance.

4. That all right angles are equal to one another.

5. That, if a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which are the angles less than the two right angles.

Table A.3: Euclid's Common Notions from *The Elements*.

Common Notions

1. Things which are equal to the same thing are also equal to one another.
2. If equals be added to equals, the wholes are equal.
3. If equals be subtracted from equals, the remainders are equal.
4. Things which coincide with one another are equal to one another.
5. The whole is greater than the part.

Appendix B

Isabel Luengo's DS1

This appendix contains a summary of Isabel Luengo's formal system **DS1** for geometry and an explanation of why it is unsound. Her description of this system can be found in chapter VII ("A Diagrammatic Subsystem of Hilbert's Geometry") of the book *Logical Reasoning with Diagrams* (Allwein and Barwise, eds.)[15], which contains the same material (slightly abbreviated) as chapters 2 and 3 of her thesis[16].

First, her syntactic objects. These are geometric objects in the plane. She recognizes four kinds of primitive syntactic objects: Boxes, Points*, Lines*, and Indicators. A Box is a dashed rectangle, a point* is a dot, a line* is a (genuinely) straight line in the plane, and an indicator is a collection of slash marks, possibly sitting on an arc. (The indicators will be used to mark segments as having equal lengths.) She recognizes four relations that can hold between diagrammatic objects: In, which tells you if a given object lies entirely inside a box; On*, which tells you if a point* intersects a line*; Indicates, which tells you if an indicator indicates a pair of points*; and Between* (this is the one that will turn out to be important in the subsequent discussion), which is defined as follows: "Point* A is

between* B and C if and only if there is a line* that goes through* A, B, and C, and A is between B and C on that line*." Note that A cannot be between* B and C unless there is a line* drawn through A, B, and C.

Next, she defines a diagram to be any finite combination of primitive diagrammatic objects, and a well-formed diagram to be one which contains a single box, such that all other primitive diagrammatic objects are In the box, such that given any two distinct points* there is at most one line* through them, and such that every indicator indicates a segment (where segment is a derived term meaning two points* on a line* l and including the part of the line* between them; an indicator indicates a segment iff it indicates the pair of points). She then defines two diagrams to be copies of one another iff there is a bijection between them preserving the four relations In, On*, Between*, and Indicates. (She actually gives a slightly more complicated equivalent definition.) She shows that this is an equivalence relation, and says that from this point forward, D will mean the equivalence class of all diagrams that are copies of D. Note that this notion of equivalence doesn't contain any topological information about how the diagram lies in the plane. In her dissertation, Luengo also discusses a formal system **DS2** that includes the relation of a ray being in the interior of an angle, and several extensions of this formal system that include the relation of points lying on the same side of a given line. These formal systems do contain some topological information. I won't discuss them further here; but they are based on **DS1**, and the examples given below that show that **DS1** is unsound also apply to these extensions.

Now, the semantics of **DS1**. A function $I : D \to E$ is an ***interpretation function*** for D iff it is a total function from the points* and lines* of D to the points and lines of E, where E is a Euclidean plane (which is defined to be anything

satisfying Hilbert's axioms), such that I takes points* in D to points in E, lines* in D to lines in E, such that point* A is on* line* L iff $I(L)$ goes through point $I(A)$, and (here's the key part) such that if A, B, and C are points* then A is between* B and C iff $I(A)$ is between $I(B)$ and $I(C)$. This definition is where the trouble first arises: note that because the definition of between* requires the points* to lie on a common line*, if D is a diagram containing three points* that don't lie on any common line*, then I isn't an interpretation function for D if $I(A)$, $I(B)$, and $I(C)$ are collinear. If $I(A)$, $I(B)$, and $I(C)$ are collinear with $I(A)$ between $I(B)$ and $I(C)$ then we would have to have A between* B and C; but it can't be, because there is no line* l that A, B, and C all lie on*. (Intuitively, it would make more sense just to require that if A is between* B and C, then $I(A)$ is between $I(B)$ and $I(C)$. If I satisfies this weaker condition, call it a ***weak interpretation function***.)

If I is an interpretation function for D, then it is a ***pre-model*** if 1) if two segments S_1 and S_2 are marked equal in D, then $I(S_1)$ and $I(S_2)$ are equal in length; and 2) if A and B are on the same side of line* l with respect to point* C, then $I(A)$ and $I(B)$ are on the same side of $I(l)$ with respect to $I(C)$. A and B are defined to be on the same side of L with respect to C iff A, B, and C are all on* L and C isn't between* A and B. So (2) is equivalent to saying that if A, B, and C all lie on* L and C isn't between* A and B, then $I(A)$, $I(B)$, and $I(C)$ all lie on $I(L)$, and $I(C)$ isn't between $I(A)$ and $I(B)$. This is already part of the definition of being an interpretation function, but it wouldn't be under the weaker definition of an interpretation function. Note that if I is a weak interpretation function satisfying these two conditions (call such a function a ***weak pre-model***), then if $I(C)$ is between $I(A)$ and $I(B)$ and A, B, and C all lie on a common line*

L, then C must lie between* A and B, by condition (2).

Luengo then defines three "deduction principles" (later called construction rules). The first of these is rule P1 (line* introduction): given a diagram D with points* A and B not on a single line* of D, one can deduce a diagram E that is "just like" D except that it has a new line* through A and B. There are a number of different ways that this rule can be construed. Because they are somewhat involved, I'm going to defer discussing them until we need to use this rule later on. The other two rules are rule P2 (point* introduction): given D containing a line L, deduce any diagram E identical to D, but with a new point* that is on* L and not on any other line*; and rule P3 (existence of segments): this is a more complicated rule that says that given a marked segment length M, a point* A on a line* L, and a given side D of A on L, you can add a new point* B on* L, on the given side of A, and mark the segment from A to B with marker M. If there are other points on the given side of A, you get a disjunction of diagrams showing the different ways B could lie on L with respect to the existing points.

A premodel M of D is defined to be a ***model*** of D if for any diagram E (or disjunctive set of diagrams S) obtainable from D via one of the three deduction principles, M can be extended to a premodel of E (or S).

We now reach a central proposition of her paper, Proposition 3.8: A pre-model M is a model iff it is 1-1. Both directions of this proposition are false.

The if direction is false because we can find diagrams D and D^* such that D has a 1-1 premodel, D^* is constructible from D, and D^* doesn't have any premodels, because it contains three points* that don't lie on a common line*, but would have to be collinear in any possible premodel. As noted above, if A, B and C are points* in D^* that don't lie on a common line*, then if $I(A)$, $I(B)$, and $I(C)$ are

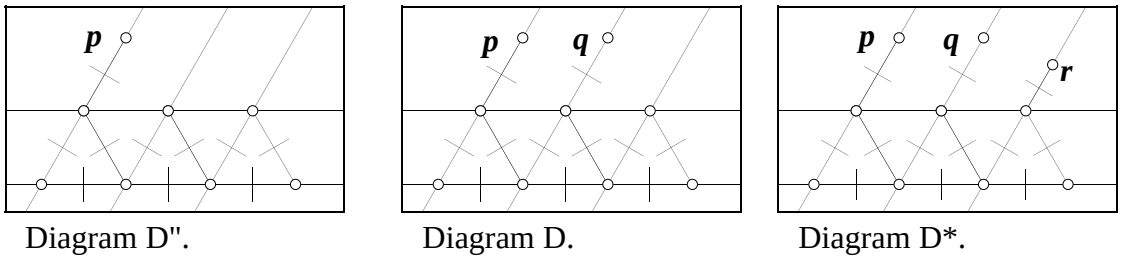


Figure B.1: A counterexample to the soundness of **DS1**.

collinear, then I can't be an interpretation function for D^*. So in this case D^* couldn't have any premodels.

Figure B.1 shows an example of how this can happen. Diagram D has a 1-1 premodel that is the one that you would expect it to have, and diagram D^* is obtainable from diagram D by rule P3. But in any possible premodel of D^*, p, q, and r would have to be collinear–they each sit the same distance from L along lines that make a 60 degree angle with L. So as discussed above, D^* can't have any premodels in her system, since there is no line through p, q, and r. But D^* is derivable from D, which has a 1-1 premodel; so the if direction of the proposition is false. Notice that more or less the same example shows that her system is in fact unsound. D doesn't have a model by her definition (because D^* doesn't have a premodel), but it is constructible from diagram D'', which does have a model. Since D'' has a model and D doesn't, but D is constructible from D'', the system is unsound. Notice that this counterexample doesn't rely on our interpretation of rule P1; so far, we have only used rule P3.

The only if direction of proposition 3.8 can also be seen to be false as follows: let E be a diagram containing two points* p_1 and p_2 (and nothing else). Let M be a premodel of E such that $M(p_1) = M(p_2)$, so M isn't 1-1. The only deduction principle that applies is Line* introduction. So let E' be a diagram obtained from

E by adding a line* L through p_1 and p_2. Let $N(p_1) = N(p_2) = M(p_1) = M(p_2)$, and let $N(L)$ be some line going through $N(p_1)$. N is a premodel of E'. So M was a non-1-1 model of E. The proof that Luengo gives here actually shows something different: that if M is a non-1-1 premodel of E, then there is a diagram E'' that can be obtained from E by applying a sequence of deduction principles, such that M can't be extended to a pre-model of E''. In any case, this direction of the proposition isn't as important, as we can simply require all interpretation functions to be 1-1.

The above example suggests that her system might be made sound by only requiring premodels to have weak interpretation functions. Under the weaker definition, diagram D^* above would have a premodel. Unfortunately, there would still be a problem. Luengo proves in her proposition 6.5 that with this change her system is not sound. The problem is with rule P1. Recall rule (P1): given a diagram D with points* A and B not on a single line* of D, one can deduce a diagram E that is "just like" D except that it has a new line* through A and B.

As I previously remarked, there are a number of different ways that this rule can be construed. Here are the four possibilities that I've considered.

(Version 1) E is a diagram containing a new line* l such that A and B are on* l and no other point* C in E is on* l, and removing l from E leaves a diagram equivalent to D. This way of construing the rule takes the words "just like" to be understood to be referring to the On* relation. An immediate problem here is that it isn't obvious whether or not we can always find such a diagram E. In fact, we can't. A counterexample is given by Desargues' Theorem, the proof of which was discussed in Section 4.1. Let D be a diagram obtained as follows: Draw a triangle ABC. Draw a point E somewhere not on the triangle. Draw in the lines AE, BE,

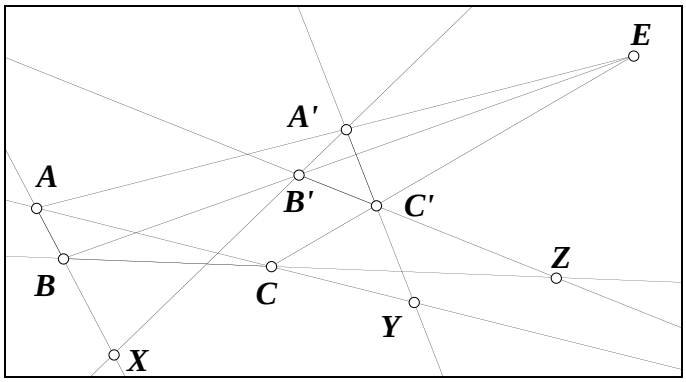


Figure B.2: Desargues' theorem.

and CE. Pick points A' on AE, B' on BE, and C' on CE. Connect A', B', and C' to obtain a new triangle $A'B'C'$. Extend the six line segments that make up the sides of the two triangles to lines. Mark point X at the intersection of lines AB and $A'B'$, point Y at the intersection of lines AC and $A'C'$, and point Z at the intersection of lines BA and $B'A'$. What you have now is diagram D, shown in Figure B.2. Desargues' theorem says that in D (and in any equivalent diagram) points X, Y, and Z must be collinear. If we want to apply rule P1 to points X and Y, there is no way to draw a straight line through X and Y that doesn't also go through Z. So if we want to construe the rule this way, we'll have to accept that it won't be possible to apply the rule to all diagrams that have two points* not on a common line*.

(Version 2) The rule could also mean that E is the diagram in which A and B are really connected by a new straight line, and any other point* C is on* the new line iff it happened to intersect the real straight line between A and B in D. We get this version of the rule if we take the words "just like" to refer literally to D as a geometric object itself. This version of the rule suffers from a different problem: it can be applied to any diagram in which there were two points* A and B that

didn't lie on a common line*, but it isn't well defined on equivalence classes of copies of diagrams. For example, the point* C might lie directly between A and B in D, but not in some other diagram D' that was equivalent to D. To fix this problem, we can modify the rule as follows:

(Version 3) E is obtainable from D by P1 as long as there is some diagram D' that is equivalent to D such that E can be obtained from D' by connecting A and B by a straight line (and, as in the previous version, any other point* C lies on* the new line iff C really lies between A and B in D'). I think that this is the best way to interpret the rule. It doesn't suffer from either of the above problems with versions one and two. In fact, if E is obtainable from D by either Version 1 or Version 2 of the rule, it will also be obtainable by Version 3. Version 3 also has the property that, unlike the previous versions, there are multiple diagrams that can be obtained from a given diagram by applying rule P1 to two given points. This seems to be what Luengo intends. In her thesis (p. 23), she writes that this rule states that "any extension of the diagram that meets a certain condition is obtainable from the diagram."

There is one other version of the rule that has been suggested: (Version 4) E is obtainable from D iff for every diagram D' that is equivalent to D, adding the straight line that runs through points* A and B gives a diagram E' that is equivalent to E. This is equivalent to saying that E is obtainable from D by version 3 of the rule, and is the only diagram obtainable from D by version 3. This version of the rule shares version 1's property that there sometimes isn't any diagram that can can be derived from D by applying the rule to two given points* that don't lie on any common line*. (This will happen any time that there are two or more diagrams derivable by version 3 of the rule.) More importantly, though,

in order to apply this rule, one has to check that for every diagram D' that is a copy of D, the result of drawing in the straight line through the given points* is a copy of E. There isn't an obvious general method for doing this; we can't check directly, because there are usually an infinite number of different copies of D. And in the cases where we can in fact show that this property holds for every such copy D', the proof may involve a great deal of prior geometric knowledge, as in the example given for version 1, in which we had to apply Desargues' theorem. Obviously, it is highly undesirable to have to use complicated geometric facts to determine if one diagram follows from another syntactically in a formal system for doing geometry. It seems to me that this problem makes this version of the rule unworkable in practice. (This version also seems inconsistent with Luengo's own use of this rule, for example in her proposition 6.5.)

Thus, the version of P1 that seems to have the fewest problems and seems most consistent with Luengo's intent is version 3. Unfortunately, under the weak definition of premodel, this rule is unsound. In fact, versions 1 - 3 of P1 are all unsound. To see this, consider diagram D^* from above, and assume that point r has been drawn a different distance above the triangles than p and q were drawn. Then by applying any of the first three versions of rule P1, to points p and q in D^*, we can obtain a diagram D''' with a line* L running through p and q, but not r; but D''' can't have any premodels (even under the weaker definition), since in any premodel, $I(L)$ would have to run through $I(r)$ if it ran through $I(p)$ and $I(q)$. But D''' was obtained from D^*, which does have (weak) 1-1 premodels. Notice that with the original definition of premodel you don't run into the second problem, because your model never contains extra points that might show up on the new line that you're constructing.

So, under the weaker definition of premodel, versions 1-3 of rule P1 are unsound. Version 4, on the other hand, is sound: any model M of a diagram D is itself a copy of D once we add in a box and indicators, so if E is obtained by applying version 4 of P1 to points* A and B in D, and N is the extension of M in which the line L through $M(A)$ and $M(B)$ has been added, then N satisfies E by the definition of version 4 of rule P1; but this version of the rule is unworkable in practice, as discussed above.

There is actually one other possible modified version of rule P1 that is sound and slightly better than version 4. I'll call this modified version P1$'$: given a diagram D with points* A and B not on a single line* of D, one can deduce the disjunctive set of diagrams $S = \{E|$ there exists a copy D' of D such that E is obtained from D' by adding the straight line through A and $B\}$. Rule P1$'$ is sound, again because any model M of D will give you a copy D' of D. Unlike version 4 of rule P1, given any diagram D with points* A and B not on* a common line*, there always exists a disjunctive set of diagrams S such that S follows from D by rule P1$'$ applied to A and B. Intuitively, this modified rule makes sense. If we connect two points by a line, we have no way of knowing in advance which other points the new line will intersect, but we know that there there will be some way of connecting the two points by a straight line. However, P1$'$ is still unworkable in practice, because in general, we have no way of checking that we've found all of the diagrams in S.

The only way that I can see around this difficulty is to relax the requirement that lines* be actually straight. If we don't require the lines* to be straight, then we can combinatorially compute the possible ways that the new line* could intersect the old points*, and using non-straight lines we can definitely realize all

these possibilities. This is more or less the way that **FG** handles this issue. Using straight lines introduces too much geometric information into the diagram.

Bibliography

[1] Aaboe, Asger, *Episodes from Early Mathematics*, New York: Random House, 1964.

[2] Barwise, Jon, and Gerard Allwein, eds., *Logical Reasoning with Diagrams*, New York: Oxford University Press, 1996.

[3] Bell, E. T., *Men of Mathematics*, New York: Simon & Schuster, 1937.

[4] Bourbaki, Nicolas, *Elements of the History of Mathematics*, Berlin: Springer-Verlag, 1994.

[5] Dipert, Randall R., "History of Logic," *Encyclopaedia Britannica*, Online version, `http://www.britannica.com`, 2001.

[6] Emerson, Ralph Waldo, *Society and Solitude*, Boston: James R. Osgood and Company, 1876.

[7] Euclid, *Elements*, T. L. Heath, ed., second edition, New York: Dover, 1956.

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

[9] Gardner, Martin, *Logic Machines and Diagrams*, Chicago: University of Chicago Press, 1982.

[10] Hilbert, David, *Foundations of Geometry*, translated by Leo Unger, La Salle, Ill.: Open Court Publishing Co., 1971.

[11] Hopcroft, John, and Jeffery Ullman, *Introduction to Automata Theory, Languages, and Computation*, Reading, Massachusetts: Addison-Wesley, 1979.

[12] Joseph, George Gheverghese, *The Crest of the Peacock: Non-European Roots of Mathematics*, London: I. B. Tauris & Co., 1991.

[13] Kline, Morris, *Mathematics for the Nonmathematician*, New York: Dover Publications, 1967.

[14] Kline, Morris, *Mathematics: The Loss of Certainty*, New York: Oxford University Press, 1980.

[15] Luengo, Isabel,"A Diagrammatic Subsystem of Hilbert's Geometry", in *Logical Reasoning with Diagrams*, Gerard Allwein and Jon Barwise, eds., New York: Oxford University Press, 1996.

[16] Luengo, Isabel, *Diagrams In Geometry*, Ph.D. Thesis, Indiana University, 1995.

[17] Peirce, Charles Saunders, *Collected Papers*, Vol. IV, Charles Hartshorne and Paul Weiss,eds., Cambridge: Belknap Press, 1960.

[18] Plutarch, *Convivialium disputationum.*

[19] Plutarch, *Parallel Lives: Marcellus.*

[20] Neugebauer, Otto, and A. Sachs, *Mathematical Cuneiform Texts*, Lancaster, PA: Lancaster Press, 1945.

[21] Renegar, James, "Computational complexity of solving real algebraic formulae," *Proceedings, International Congress of Mathematicians*, Tokyo: Springer-Verlag, 1991.

[22] Shin, Sun-Joo, *The Logical Status of Diagrams*, Cambridge: Cambridge University Press, 1994.

[23] Simmons, George F., *Calculus with Analytic Geometry*, New York, McGraw Hill, 1985.

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

[25] Whitehead, Alfred North, *An Introduction to Mathematics*, London: Williams and Norgate, 1911.