# Geometry in Lean

An attempt to formalise geometry in Lean.

## Motivation

During the summer of 2018, I was part of the Xena project run by Kevin Buzzard. Due to my history as an Olympiad student, geometry has always been one of the four main chapters in mathematics. On the first day, as everyone was choosing their subjects, I realised geometry was missing. Real numbers had recently been added to lean, but using cartesian coordinates to do geometry loses the beauty and elegance. Also, synthetic geometry can be applied to other geometries.

## Tarski > Euclid

Everyone knows how Euclid is the father of geometry, and his postulates are the first example of an axiom system. So naturally I decided to take a look at Euclid’s work, and started from there. I was unpleasantly surprised when I learned that not only is Euclid’s language very vague, his axiom system is not as rigorous as I had thought. So, I searched for more recent axiom systems for geometry. I found Hilbert’s and Tarski’s, both with similarities. Hilbert has 3 primitive objects, points lines and planes, while Tarski only has points. They both have a betweenness relation, B x y z, which notes whether the point y lies on the x z segment. They also have a congruence relation, a b ≡ c d, which takes in 4 points and means the distance from a to b is same as the distance from c to d. Hilbert has a third relation which denotes if a point belongs to a line.

I chose Tarski’s system, since only having one type (Point) was going to make life a lot easier. Also, Tarski is a famous logician, so his methods are more suitable for a computer as well. I found this book: Schwabhäuser, W., Szmielew, W., and Alfred Tarski, 1983. Metamathematische Methoden in der Geometrie. Springer-Verlag. It’s in German, but they start from the axioms and prove every result one by one, so it is a perfect source. I started from the beginning, and went through the theorems one by one. Sometimes if I found an easier solution from theirs, I used my own proofs instead. And since we have tools such as simp, I was able to shorten quite a few of them. Some cases where they prove general results, I only proved in two dimensions, since that was all I was interested about. I tried to stay computable where possible, but the 11th axiom (regarding continuity of a segment) is of course noncomputable. I know I did not do this in the most elegant way, and I hope it can be improved in the future. I named my theorems the same as in the book, so theorem 13.1 is thirteen1 in my code. I went through most of the early chapters fairly quickly, but I struggled a lot when I got to chapter 13. After introducing the cosine function, the book uses it to prove theorems on similarity. I would have liked to have been able to identify my lengths with the positive reals, but that surpasses my lean knowledge.

## The Actual Code

From now on I will comment on my code, aptly named axioms and taski\_1 through 8. I decided to call the betweenness relation B, and the congruence one eqd (EQuiDistant). I added the 11 axioms, the first 10 of which are computable. I first started with existence, but later changed the axioms to be constructive.

### Tarski\_1

In chapter 2 I show basic results about eqd being an equivalence relation. Chapter three is basic results of B. three5 to three7b are theorems about the order of 4 points on a line.

In chapter 4 I define cong and col. Cong is a relation on 6 points, which says the distances between the first three are equal to the respective distances of the last three, i.e. cong a b c d e f means the triangle a b c is congruent to the triangle d e f (but we cannot prove that in lean yet). col a b c says that one point is between the other two, but it doesn’t matter which one.

Chapter 5 is very interesting. five1 is the first tough theorem, even Tarski thought it had to be an axiom, but Szmielew proved the result with the other axioms. I then define distle (distance less than or equal to), where distle a b c d means ab is shorter than cd. I also defined the strict version distlt.

In chapter 6 there is a strange relation, for which I couldn’t find a nice name so I just called sided. Sided a b c means the point c is on the ray starting from a and passing through b. Using this I define ray, our first new type, of type set point.

### Tarski\_2

Now I can define l a b, the line passing through a b. However, l a b is only of the class line if we give it a proof of a ≠ b. I was told this would be the most efficient way when I was having trouble in chapter 11. I then define is (intersection), where is x A B means A and B are distinct line which share the point x.

In chapter 7 I define M, the middle point of a segment. Note that we do not know that every segment has a middle point, I just define the notation for when we already have such point. After proving basic properties, I define S, the symmetry of a point i.e. the reflection of one point about another. After proving a few basic results, we get to our first substantial theorem. Seven25 states given a point equidistant to two points, we can find the middle point of those two points. In other words, given an isosceles triangle we can find the middle point of the base.

### Tarski\_3

I start chapter 8 by defining R, which implicitly means R a b c is a right angle. I use this to define perp, which says two lines are perpendicular. After the necessary lemmas, I prove in eight22 that every segment has a middle point.

### Tarski\_4

In chapter 9, I define Bl (Between Line). This means two points are separated by a line. This is going to be very important when we start talking about planes, and which side of a line a point is. I also define side, analogous to sided, when two points are on the same side of a line.

Now we can define a plane, using a line and a point. The second half of chapter nine contains a lot of theorems about higher dimensions, but since I was only interested in 2D, I ignored those.

### Tarski\_5

Now I define the mid function, which takes in two points and spits out their middle point. I use this to define Sl (Symmetry about a Line). Now in chapter 11 I define eqa, a relation that takes in 6 points and tells us if the two angles are equal. We say the angles are equal if we can construct two congruent triangles, one on each angle. Then I make three theorems which are different statements equivalent to eqa. Note that with our definition, reflex angles don’t exist; since according to eqa, the angle is equal to the angle .

Then I prove that symmetry preserves angles. Then we show vertical angles are equal, and that we can construct any angle on any segment. eleven16 states that all right angles are equal (Euclid would have been proud). eleven20 states there is a unique perpendicular from a point to a line. eleven21 states that all are equal and all angles are equal.

### Tarski\_6

eleven22 proves that we can add and subtract angles (we aren’t actually using angle measures, we just show that if two pairs of angles are equal, joining them together produces an equal result.

We define I (Interior), where I p a b c means the point p is in the interior of the angle abc; the interior being the angle less than (or a b c collinear). With I, we can now define less or equal than for angles.

After showing some basic properties of ang\_le, we show is the greatest, and is the smallest angle. Now we define ang\_lt, a strict version of ang\_le. Thanks to these definitions, we can now define acute and obtuse. After a whole lot of lemmas, we prove triangle similarities: ASA, SAS, AAS, SSS, SSA.

### Tarski\_7

We define Pl (the plane that contains our 3 given points in our axioms). We prove that every point is in this plane, and every plane is equal to this one, since axiom 9 doesn’t allow for three dimensions. We prove there’s a unique perpendicular from a point to a line.

Now we can start talking about parallel lines, which requires the 10th axiom. We define dpar (distinct parallel lines), and par (parallel lines not necessarily distinct). Then we prove lines intersect iff they are not parallel. As always, we show par is an equivalence relation (transitivity after twelve 11). twelve6 states a line always lies on the same side of a line it is parallel to. twelve9 states that if two lines are perpendicular to the same line, they are parallel to each other. twelve10 states that from any point, we can draw a line parallel to a given line. Note that this line is not necessarily unique.

In twelve11, we prove that the line in twelve10 is unique, but only by using axiom 10 (Euclid’s axiom). twelve17 through twelve20 prove various lemmas concerning parallelograms and how certain quadrilaterals with equal sides must be parallelograms.

twelve21 is a very familiar theorem: the Z rule for angles. twelve22 is the corresponding angles rule. twelve23 is the common proof that the angles in a triangle form a straight line. thirteen8 is a theorem from chapter 13, but it can be proved here, and it makes the following proofs quicker. twelve24 states a perpendicular drawn on two equal angles falls on the same side (i.e. if two angles are equal, they are both acute/obtuse).

Chapter 13 is where I really started struggling with how to go around my definitions. I made an order relation on point x point, which will be our distances. Then the same with angles. After proving loads of trivial things about angles and distances, we define cosine (by cheating). We define a function that takes in an angle and a distance and spits out , the adjacent side in a right triangle with angle and hypotenuse . We will be using cos for when we need to do ratios of distances, which is necessary to be able to talk about similarity. After proving things like cos 0 = 1 and cos 90 = 0, we also show other basic properties of cos.

### Tarski\_8

thirteen1 is basically the triangle midpoint theorem. thirteen2 is a lemma which is a weak version of two angles that span the same arc are equal. thirteen7 is a proof that cos is commutative, i.e. .

From here I wanted to fix distances and angles so they were easier to work with, but it proved too challenging at the time. But people have become much more capable in Lean, and newer additions to the library might make the task easier. I hope someone out there can make use of this project.