Lecture 20

CS 4860

November 3, 2016

1 Introduction

On November 22, the day before Thanksgiving break, we will have an in class discussion about the role of logic in computer science and in mathematics. This leaves 6 lectures devoted to constructive mathematics and type theory. Today we continue our discussion about what Edwards calls the "great divide" in mathematics that occurred in roughly 1880. We will see signs of this split even further back in history. On one side we have so called "classical" mathematics (or Platonic mathematics) and on the other side constructive (or computational) mathematics.

Classical (Platonic)	Constructive (Computational)
Dedekind	Kronecker
Cantor	Dirichlet
Weirstrauss	Gauss
Hilbert	Riemann, Euler
(Bourbaki group)	Poincaré, Brouwer, Bishop

We see this divide to some extent even in ancient times:

Plato	Aristotle
	Euclid

We see it being manifest in the nature of basic foundational theories:

set theory	constructive type theory

2 The Greeks

"logic eventually came to be independent of and prior to mathematics."

- Plato believed that mathematical truths preexist in a world independent of man - *Platonic reality*. We should say *Platonic math*, not classical math, because of Aristotle, Euclid, and Archimedes.
- Aristotle thought logic was preliminary to science and philosophy, and that deductive proof is the sole basis for facts in mathematics.

  Aristotle criticized Euclid's definition of a point ("a point is that which has no part"). He argued that points can't make up anything continuous like a line, that only by motion can points generate a line, and that arithmetic is prior to geometry (need 3 points to define a triangle).

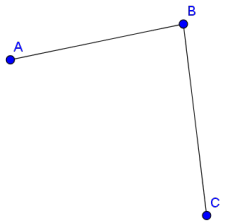
  - only the *potentially* infinite exists
  - space is potentially infinite, can be subdivided but is not infinite in extent.
  - time is potentially infinite in both senses
  - definitions must be in terms of prior concepts defined (is this strictly true for us? What about recursive definitions, e.s. streams?)
  - definitions tell us what a thing is, but not that it exists

3 Euclid's Proposition 2

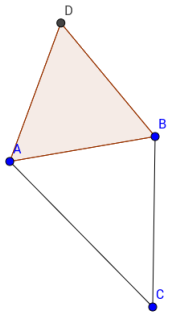
To place at a given point [as an extremity] a straight line equal to a given straight line.

Proof:

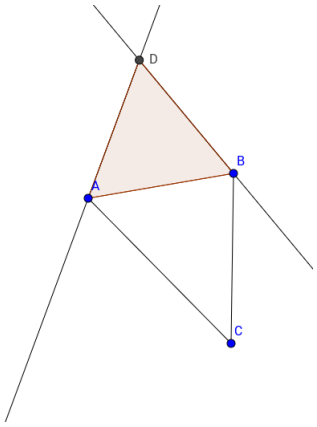
(1) Construct the line from A to B (**assumes** $A \neq B$).



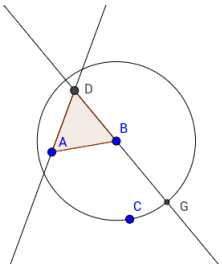
(2) Construct an equilateral triangle on AB, ΔDAB (**using proposition 1**): $AB = AD = DB$.



(3) Produce straight lines DA and DB.



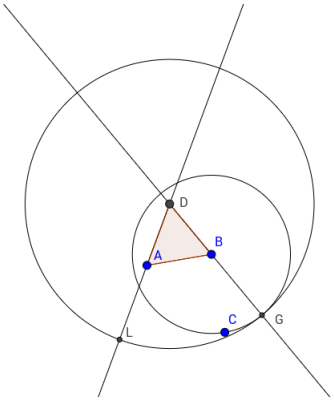
(4) Construct a circle with center B and radius C, call it $\bigcirc BC$.



(5) Point G is where $\bigcirc BC$ intersects DB.

(6) $BG = BC$ (**same radius**).

(7) Construct a circle with center D and radius G, call it $\bigcirc DG$.



(8) Point L is where $\bigcirc DG$ intersects DA.

(9) $DL = DG$ (**same radius**).

(10) $AL = BG$ is the result of subtracting equals from equals, noting $DA = DB$.

(11) Thus, $AL = BC$, by *transitivity of equality.*

(12) Hence, BC is "copied to" AL as required.

QED

4 A Modern Version of Proposition 2

To place at a given point [as an extremity] a straight line equal to a given straight line.

Euclid's proposition can be proven programmable but requires two axioms of modern geometry.

(I) Separate : *Two points are considered separate (indicated by $\neq$) if a point can be constructed between them.*

$$\alpha \neq \beta \Rightarrow \exists \mu : \alpha - \mu - \beta$$

(II) Extension: *A line segment of non-zero length can be extended.*

$$\forall \alpha, \beta, \gamma, \delta : \alpha \neq \beta \Rightarrow (\exists x : \beta x \equiv \gamma\delta)$$

Note, the segment $\overline{\alpha\beta}$ is non-zero if $\alpha \neq \beta$.

With these axioms, Euclid's second proposition can be re-written using propositional logic:

$$(1) \quad \forall a, b, c : a \neq b \Rightarrow \exists x : ax \equiv bc.$$

In (1), the "given point" from Euclid's statement of the proposition is a and the "given straight line" from Euclid's statement of the proposition is $\overline{bc}$.

The Programmable Proof

Where Euclid's proof of Proposition 2 is lengthy and not rigorous in the modern sense of a proof, the programmble proof is both simple and rigorous. One can see the the programmable proof of (1) follows from (II) above with $\alpha = b$, $\beta = a$, $\gamma = b$, and $\delta = c$:

$$\forall b, a, b, c : b \neq a \Rightarrow (\exists x : ax \equiv bc).$$

Or simply,

$$\forall a, b, c : b \neq a \Rightarrow (\exists x : ax \equiv bc).$$

The Construction

To construct a diagram of the programmable proof in the style of Euclid, one simply extends $\overline{ba}$ by the length $\overline{bc}$ to obtain X.