Can Computers Understand Their Own Programs?

Prof Sir Tony Hoare

Auxiliary question 1.

- Can people understand their own rational thought processes?

- [1] Aristotle, Prior Analytics. (~350 bce)
  - classics.mit.edu/Aristotle/prior.html
- [2] Euclid, Elements Book 1 (~300bce)
  - aleph0.clarku.edu/~djoyce/java/elements.html

Auxiliary question 2 .

- How can a program be understood?

[3] A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc, Ser 2 Vol 42 (1) 230-265 (1936).

[4] A.M. Turing, Checking a large routine, Report on a Conference on High Speed Automatic Calculating Machines, Cambridge Univ. Math. Lab 67-69 (1949)

Auxiliary question 3.

- How do we know whether the computer understands something?

- [5] A.M. Turing, Computing machinery and intelligence, MIND Vol 59 No. 236 (Oct 1950)

Question 1

- Can people understand their own rational thought processes

Aristotle 384-322 BC.

Founded the Lyceum in Athens, teaching

sciences: physics, biology, zoology;
aesthetics: poetry, theatre, music,
ethics: politics, government, rhetoric,
philosophy: metaphysics, logic, linguistics

Logic is defined by

- a notation for each relevant concept
- a grammar for each line of the proof
- a grammar for each valid deductive step

Grammar

- Let S stand for the subject of a sentence,
- Let P stand for the predicate of the sentence
  - e,g, Socrates, sharks, fishes, men.
- The four permitted forms of sentence are:
  - (a) All S are P (e) No S are P
  - (i) Some S are P (o) Some S are not P

Barbara

(Major premise)	All S are M	(a)
(Minor premise)	All M are P .	(a)
(Conclusion)	All S are P.	(a)

Celarent

No M are P (e)

All S are M (a)

No S are P (e)

Darii

- All M are P (a)
- Some S are M. (i)
- Some S are P (i)

24 syllogisms

- Barbara, Barbari, Barnalip, Baroco, Bocardo, Camestres, Camestros, Calemes, Calemos, Celarent, Celaront, Cesare, Cesaro, Darapti, Darii, Datisi, Dimatis, Disamis, Felapton, Ferio, Ferison, Fesapo, Festino, Fresison.
- Modern logic has just one rule of inference, and a couple of axioms,
- and is much more powerful.

Examples from Biology

Barbara		Celarent	
All sharks are selachians	(a)	No selachians are fish	(e)
<u>All selachians inhabit the sea</u>	(a)	<u>All rays are selachians</u>	(a)
All sharks inhabit the sea	(a)	No rays are fish	(e)

A five-line proof

All sharks are selachians (a)

<u>All selachians inhabit the sea</u> (a)

All sharks inhabit the sea (a)

<u>Some sharks are carnivores</u> (i)

Some inhabitants of the sea are carnivores (i)

Grammar of proofs

A **proof** is a sequences of sentences
in which each sentence is either a premise
or it is the last line of one of the 24 syllogisms,
and the first two lines of that syllogism
occur earlier in the **proof**.

Principles

- Validity of a proof does not depend on
  - Its subject matter
  - The desirability of the proven result
  - The person who is presenting the proof
  - The truth of the premises
- It depends only on the syntactic form

Computer reasoning

- Computers easily check conformity of a proof to the given deductive rules.
- Computers discover proofs by exploring all the possible deductions from lines proved so far.
- Computers were essential to proof of Four-colour Theorem and the Kepler Conjecture.

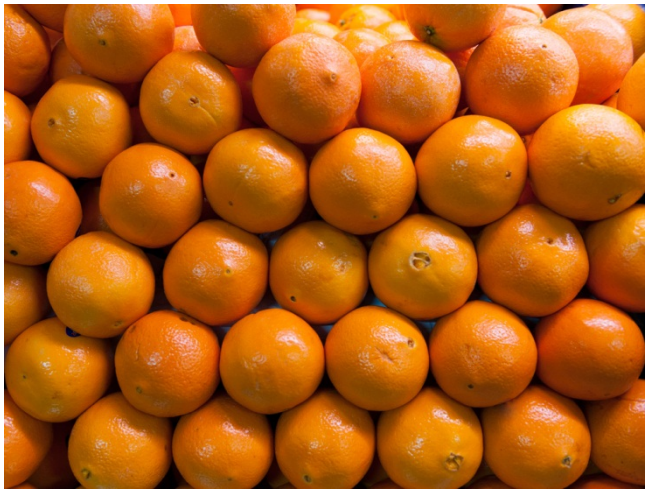
The four-colour theorem

- To colour each side of all borders differently, no map needs more than four colours.
- The left diagram needs all four.
- The right diagram needs only three.

The four-colour theorem

- The proof in Coq by Georges Gonthier
  examines 633 cases
- Each case requires round a million proof steps

The Kepler Conjecture



This is the way to pack the most oranges into a large container.

Question 2

- How can a program be understood?

Euclid

worked in Alexandria, round 300 BCE.

His *Elements* was the main textbook for teaching mathematics (especially geometry) until the late 19th or early 20th century.

Euclid also wrote on perspective, conic sections, spherical geometry, number theory and rigor, and the greatest common divisor.

Constructions

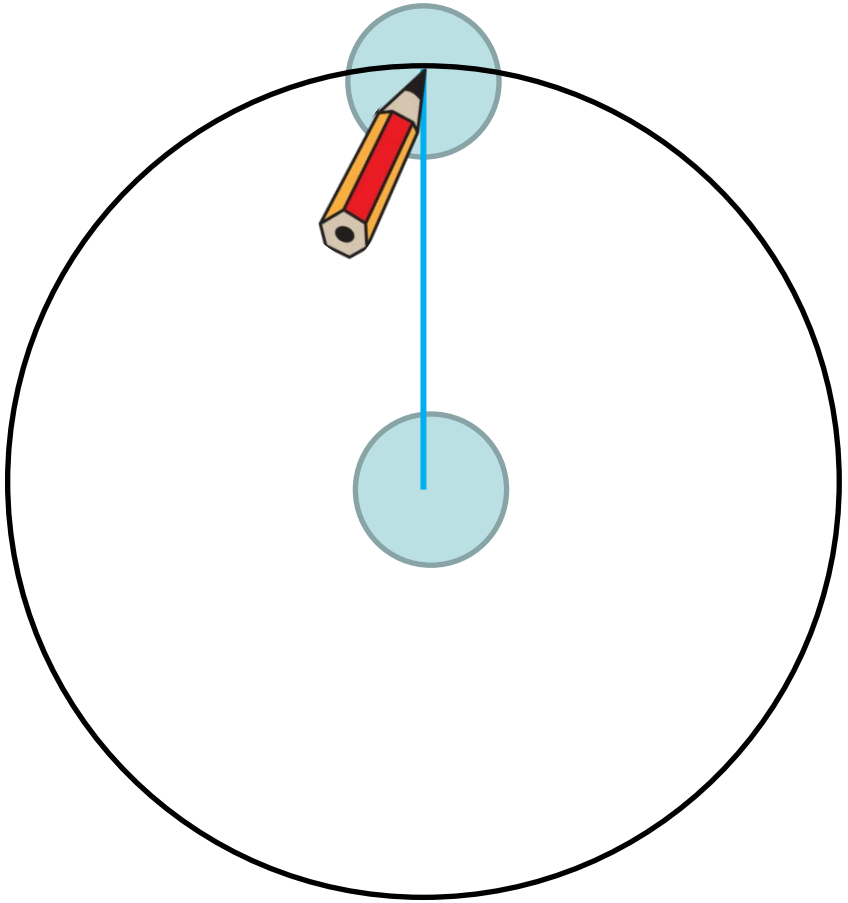
- A geometric construction describes how to draw
  - a figure, line, point,..
  - which satisfies some desired property,
  - together with a proof that it does so!
- It is written in a programming language
  - with assignments, sequencing,
  - subroutines, parameters,
  - preconditions, postconditions,…
- and proof of correctness of the program!

Five postulates

1. To draw a straight line between two points.
2. …
3. To draw a circle with any centre and radius.
4. …
5. … parallel postulate…

These are the five basic actions of the language

To draw a circle with any centre and radius. (postulate 3).

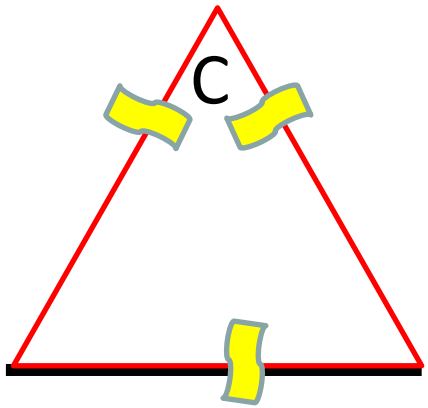


23 Definitions

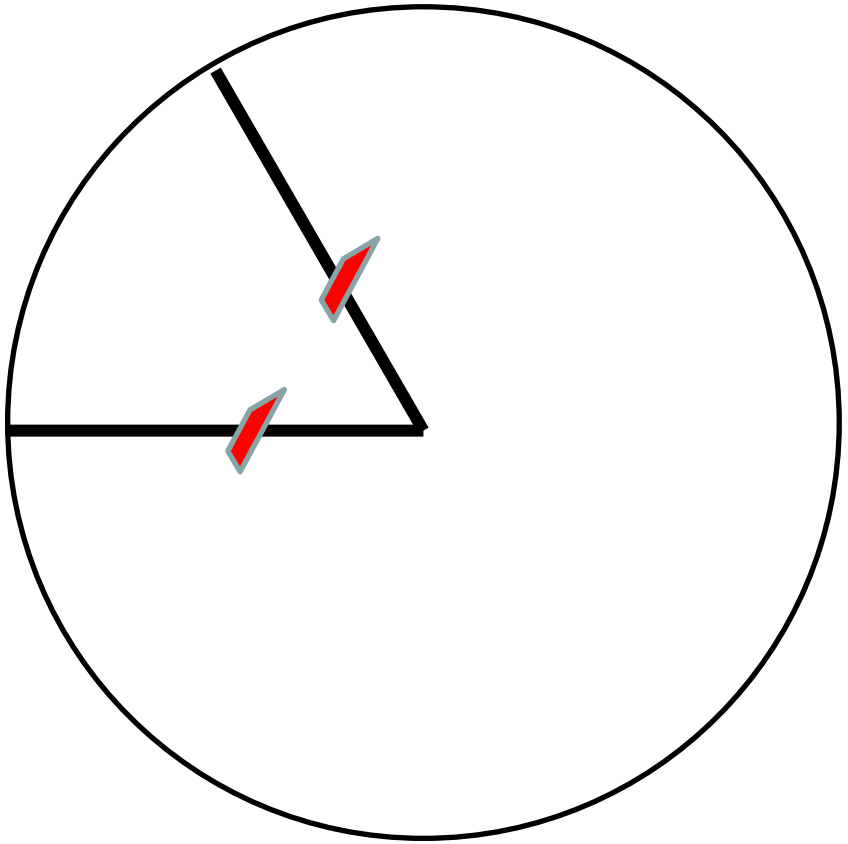
1. A *point* is that which has no part
2. ….
15. A *circle* is …
16. Its *centre* is ….
20. An *equilateral* triangle has three equal sides.

Words of the language are related to each other and to their meaning in the real world.

An equilateral triangle has equal sides (Def 20)



All straight lines from the boundary to the centre of a circle are equal (Def 14, 15, 16)



Five common notions

1. Two things that are both equal to a third thing
   are equal to each other.
2. If equals are added to equals, the wholes are equal
3. If equals are subtracted from equals…
4. Things which coincide are equal
5. The whole is greater than the part

48 Propositions of Book 1

1. To construct an equilateral triangle with given side.
2. …

47/8 Pythagoras' theorem

Subroutines

Propositions are subroutines that can be called by name repeatedly in later proofs, to perform useful constructions.

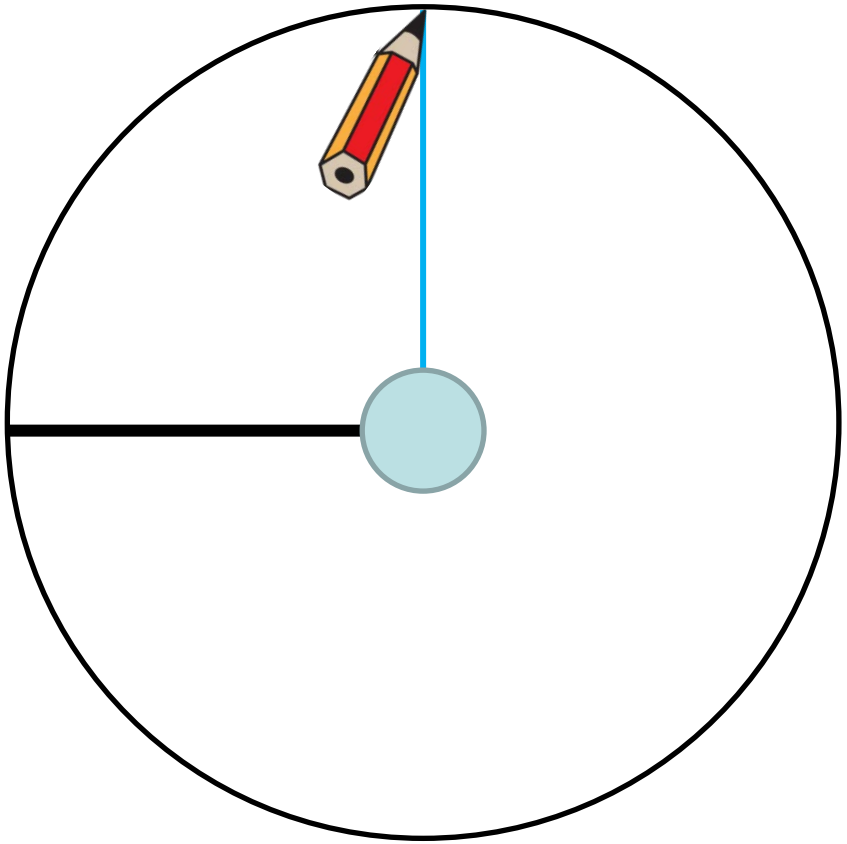
The proven properties of the construction can be used as assumptions of the proof of any proposition that calls it.

1. To construct an equilateral triangle

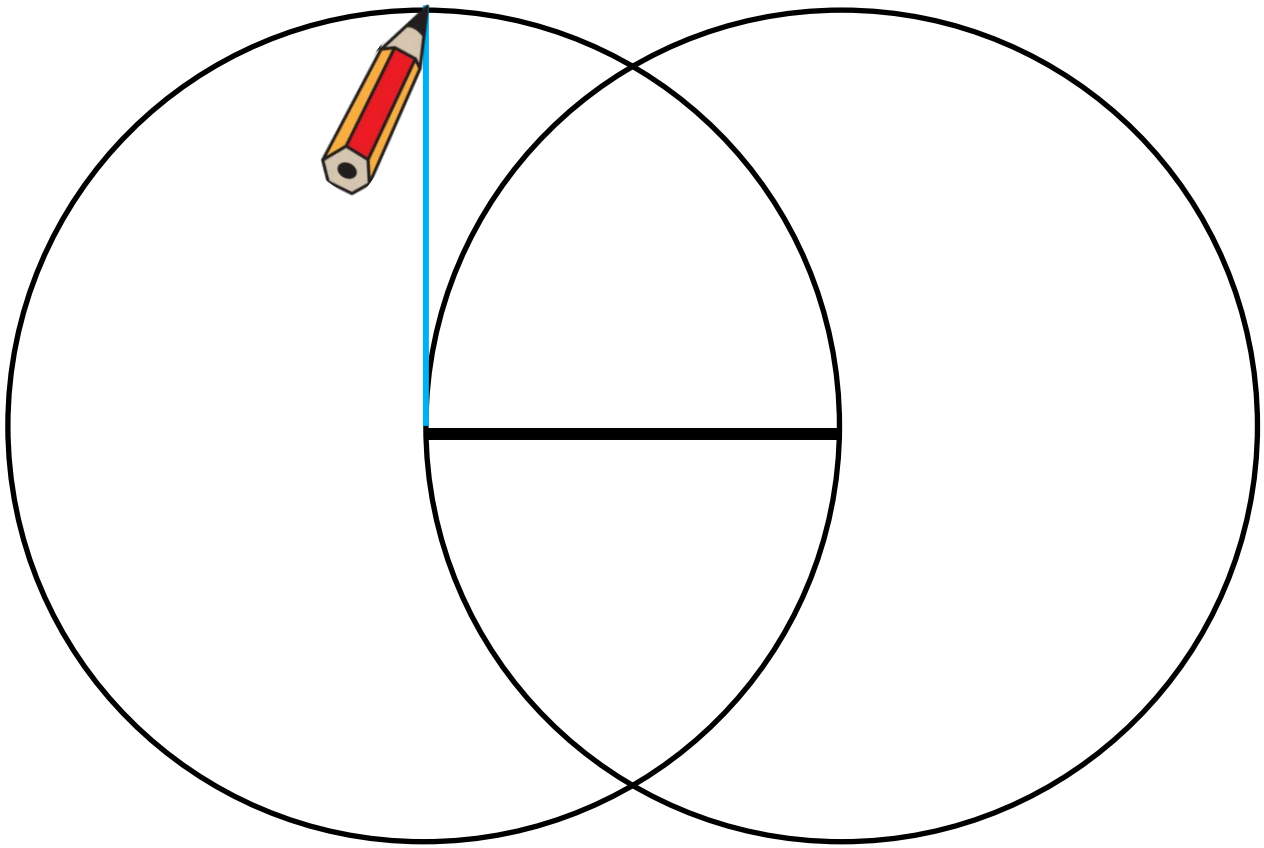
with a given line as one of its sides

choose one end of the given line

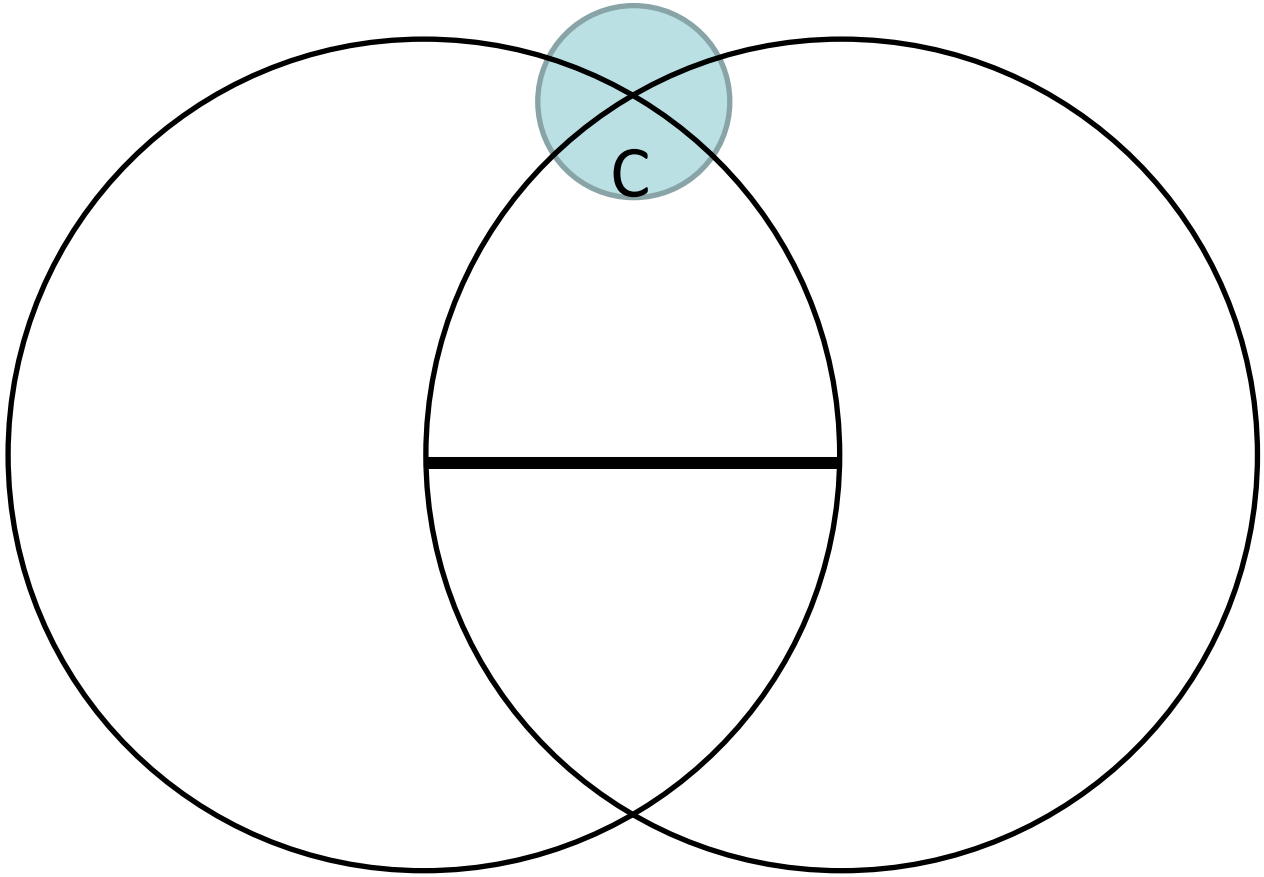
Draw a circle with the line as radius and its centre at one end (postulate 3).



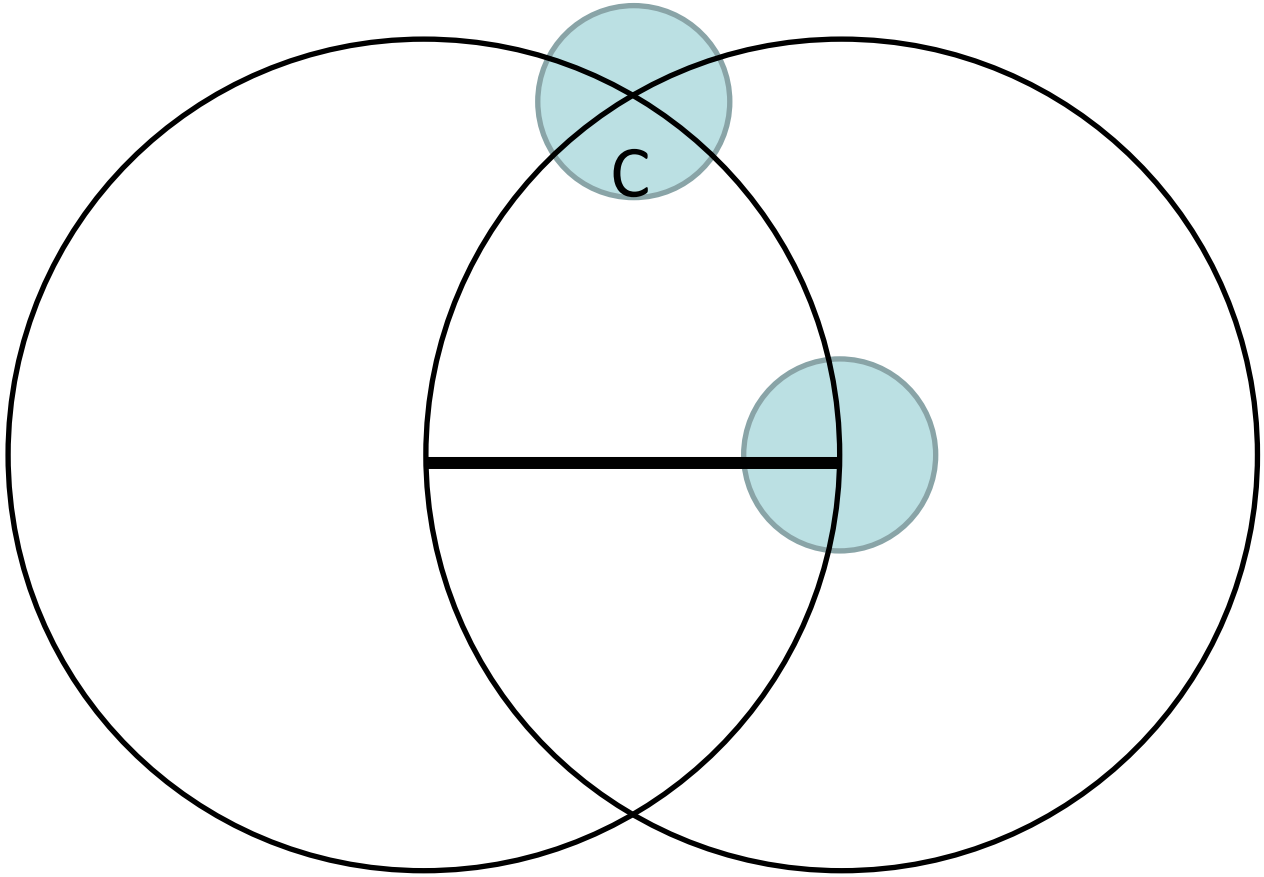
Then draw a circle with the line as radius and centre at the other end



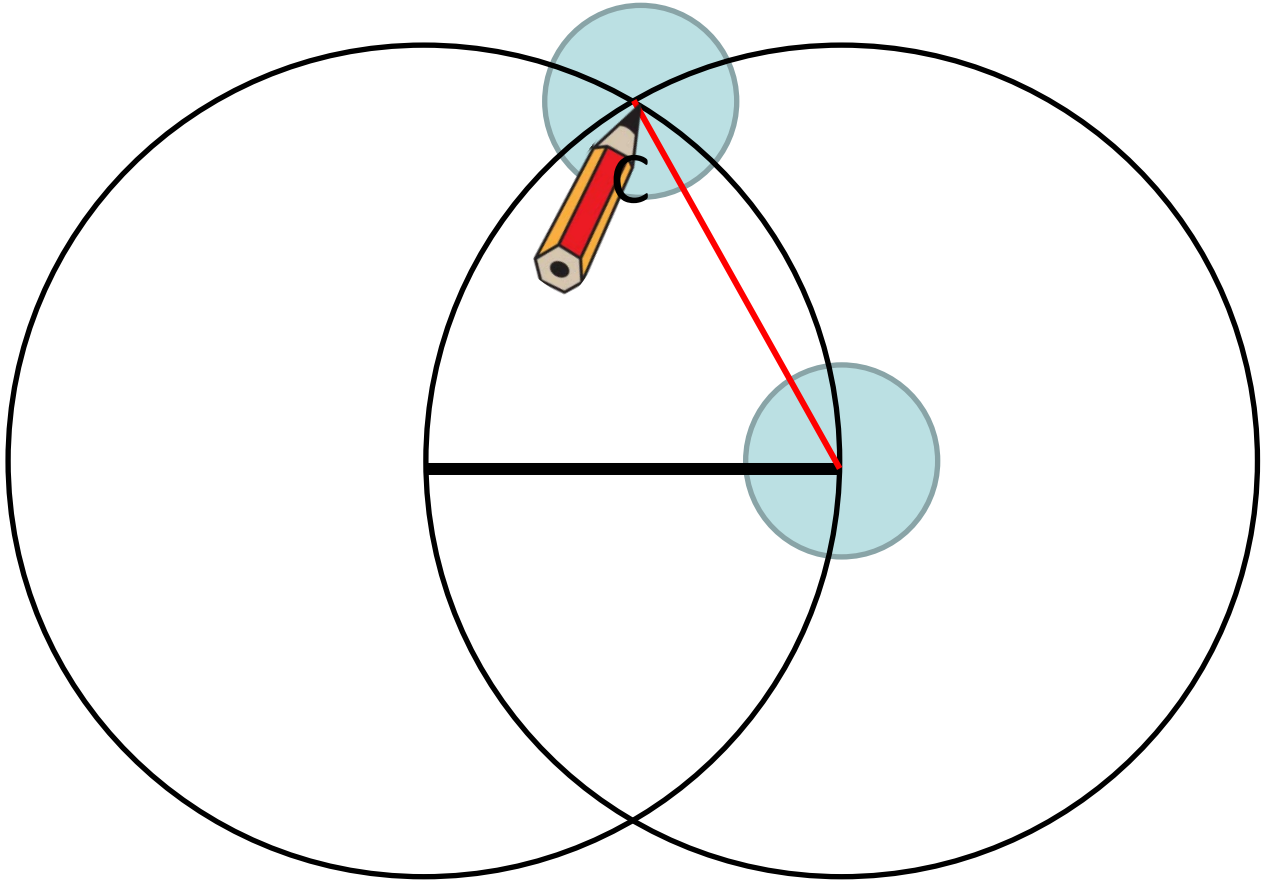
Now choose a point where the two circles intersect each other



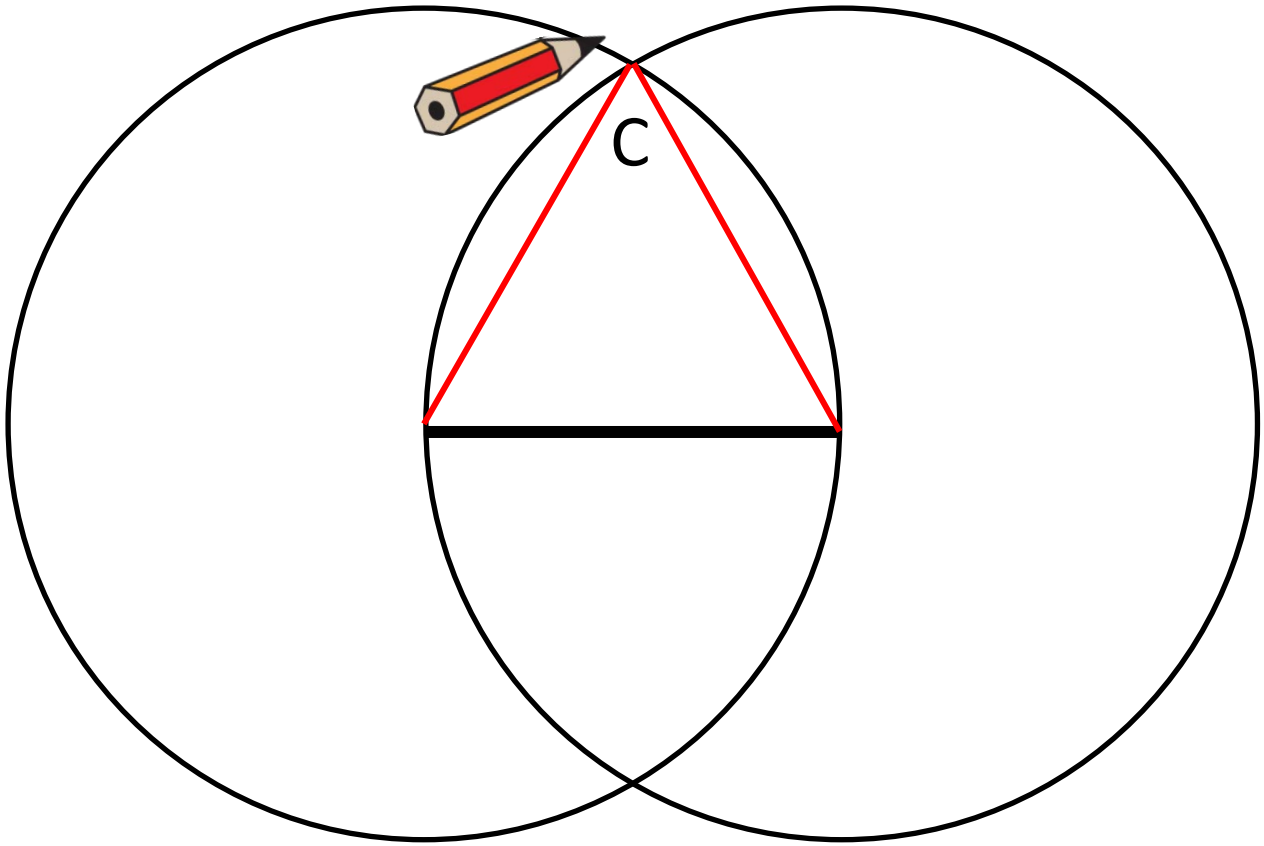
..and one end of the given line



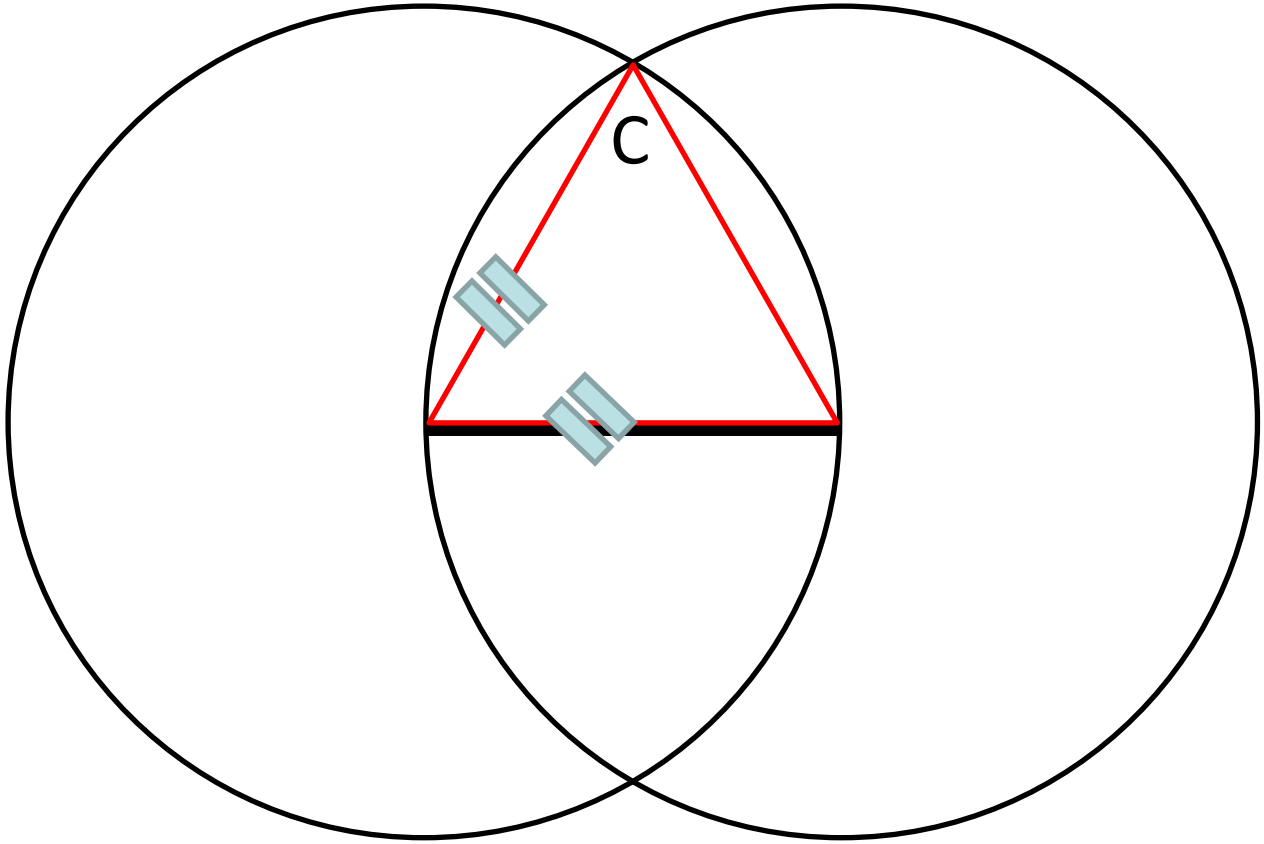
Then draw a line between them (Postulate 1)



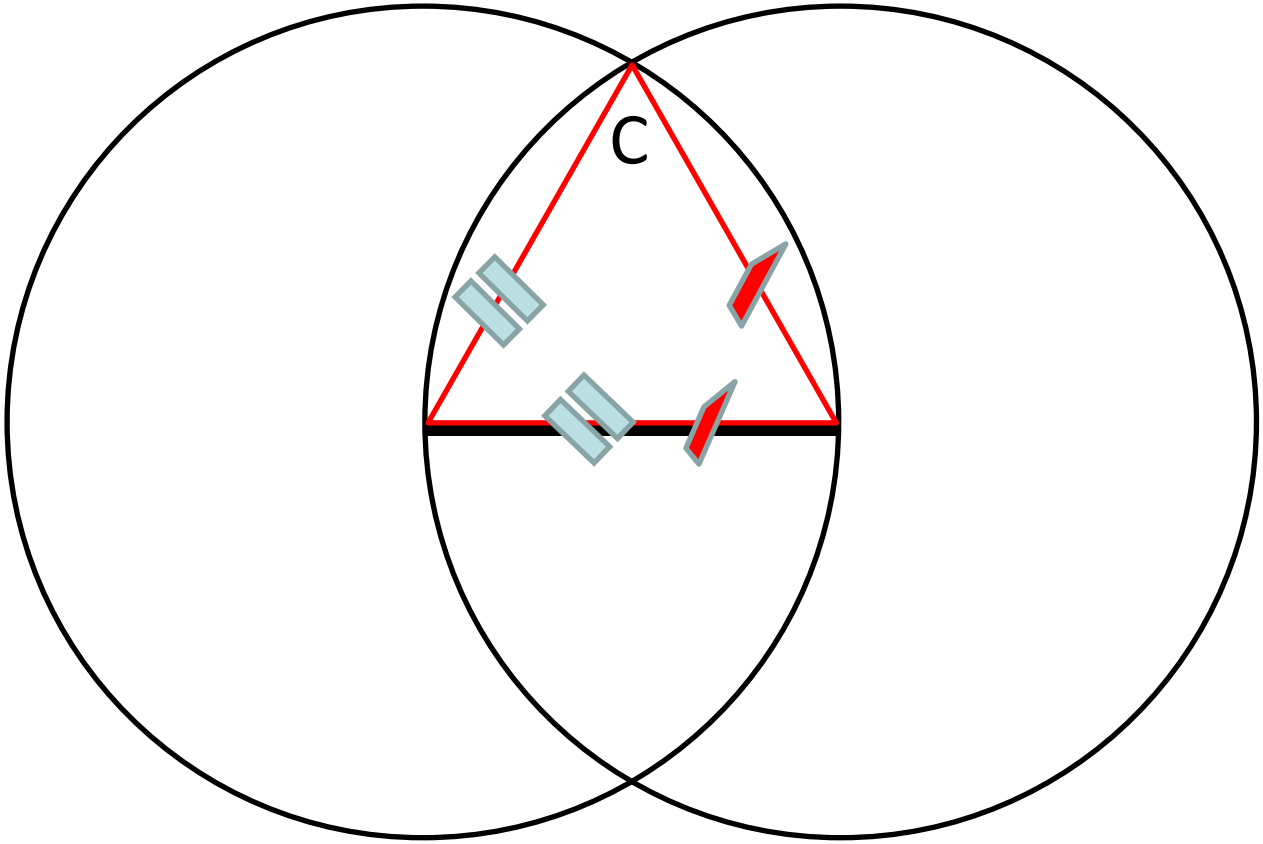
Do the same on the other side (Postulate 1)



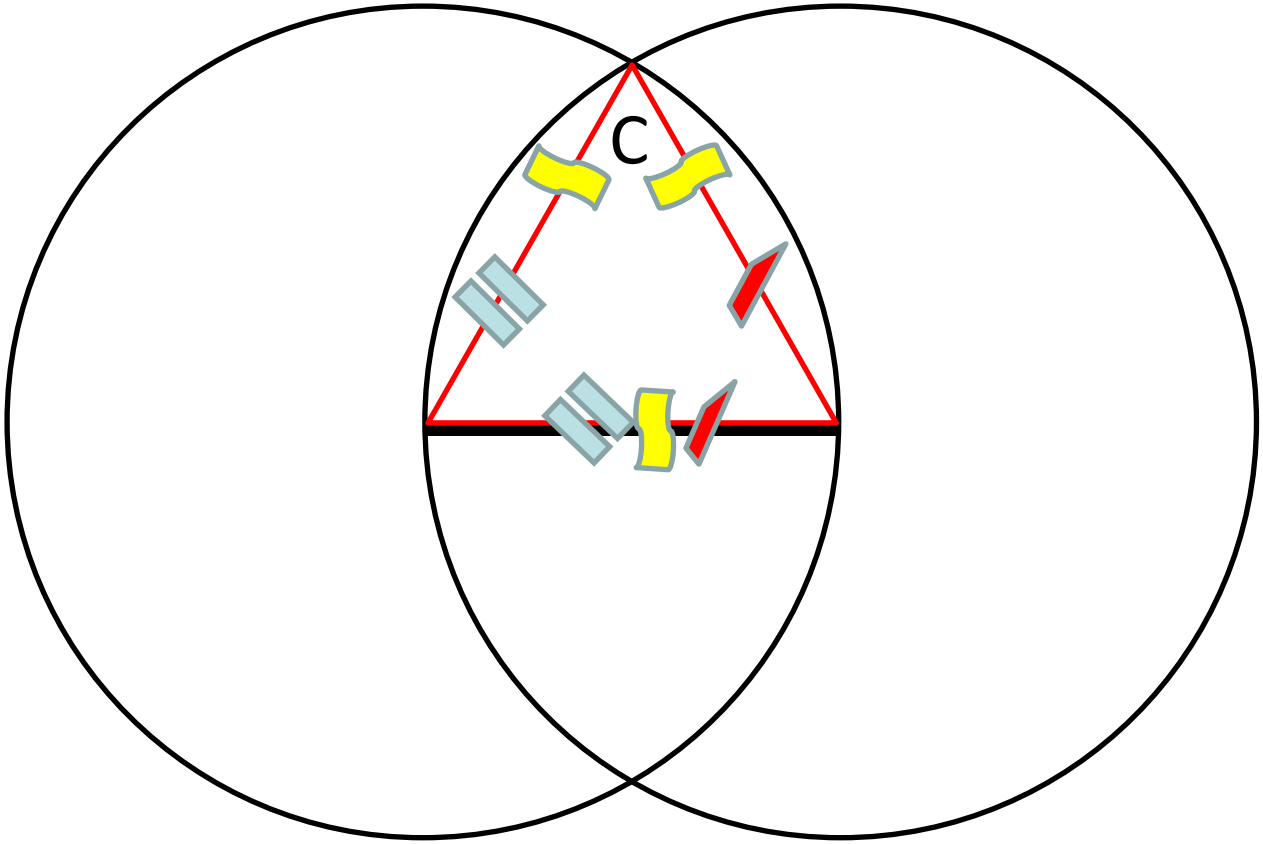
The lines marked // are equal, being radii of the left circle (Def. 15)



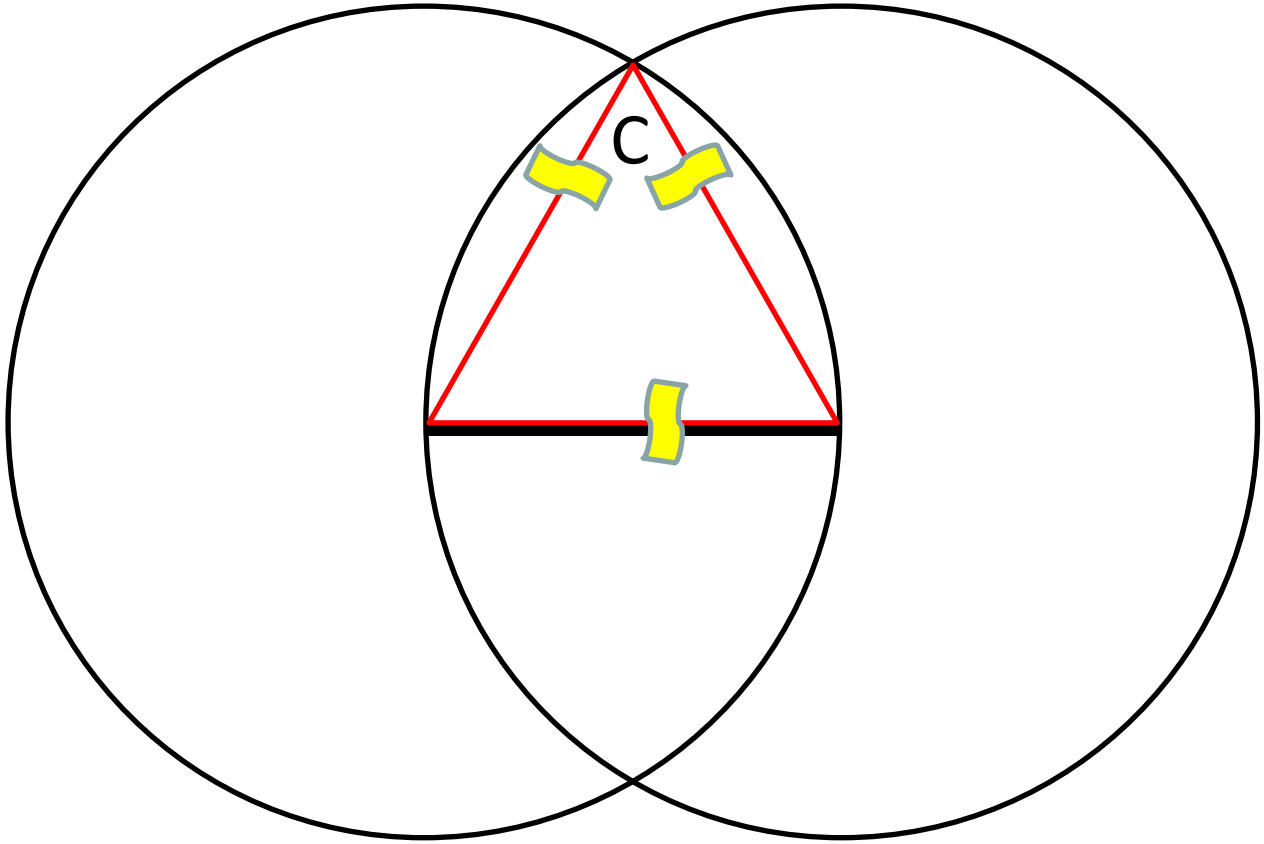
The lines marked / are equal, being radii of the right circle (Def. 15)



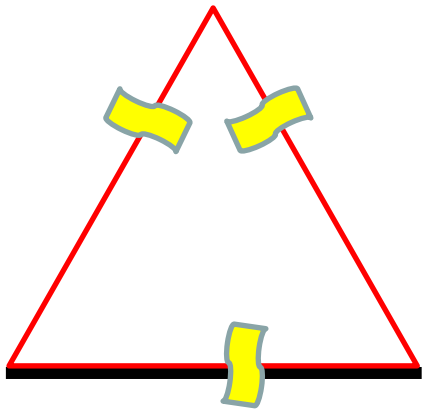
The lines marked are equal, (common notion 1)



The triangle is therefore equilateral (Def 20) Q.E.D.



The caller of this proposition does not need to reproduce or even know the construction.



Summary of Euclid's method

- primitives (postulates), 'Draw a circle with centre …'
- definition of new names 'Choose a point C on …'
- sequencing of commands 'Draw … and then draw …'
- subroutines (propositions) 'Draw an equilateral triangle'
- preconditions (Data) 'Given a straight line…'
- postconditions (QED) '…the triangle is equilateral'

Alan Turing

- invented a logic for reasoning about programs.
- The Turing machine can use this logic to answer questions about its own programs.
- And to justify its answers by proof.

The Hoare triple: P{V}R

means: When P , action V ensures R

where V changes (part of) the world

P describes an (initial) state of that part

R describes the resulting (final) state

Examples of triples

- When the radio whistles,

turning the suppression knob clockwise
ensures the absence of the whistle.

- When two distinct points are closer than r,

drawing a circle of radius r around each point,
ensures that their circumferences intersect at two places.

- When the value of x is greater than three

increasing the value of x
ensures that x is greater than 4.

Sequential execution

- When P, V1 ensures S

  When S , V2 ensures R

When P , (V1 then V2) ensures R .

Conditional tests.

- When P and B , V1 ensures R

  When P and not B , V2 ensures R

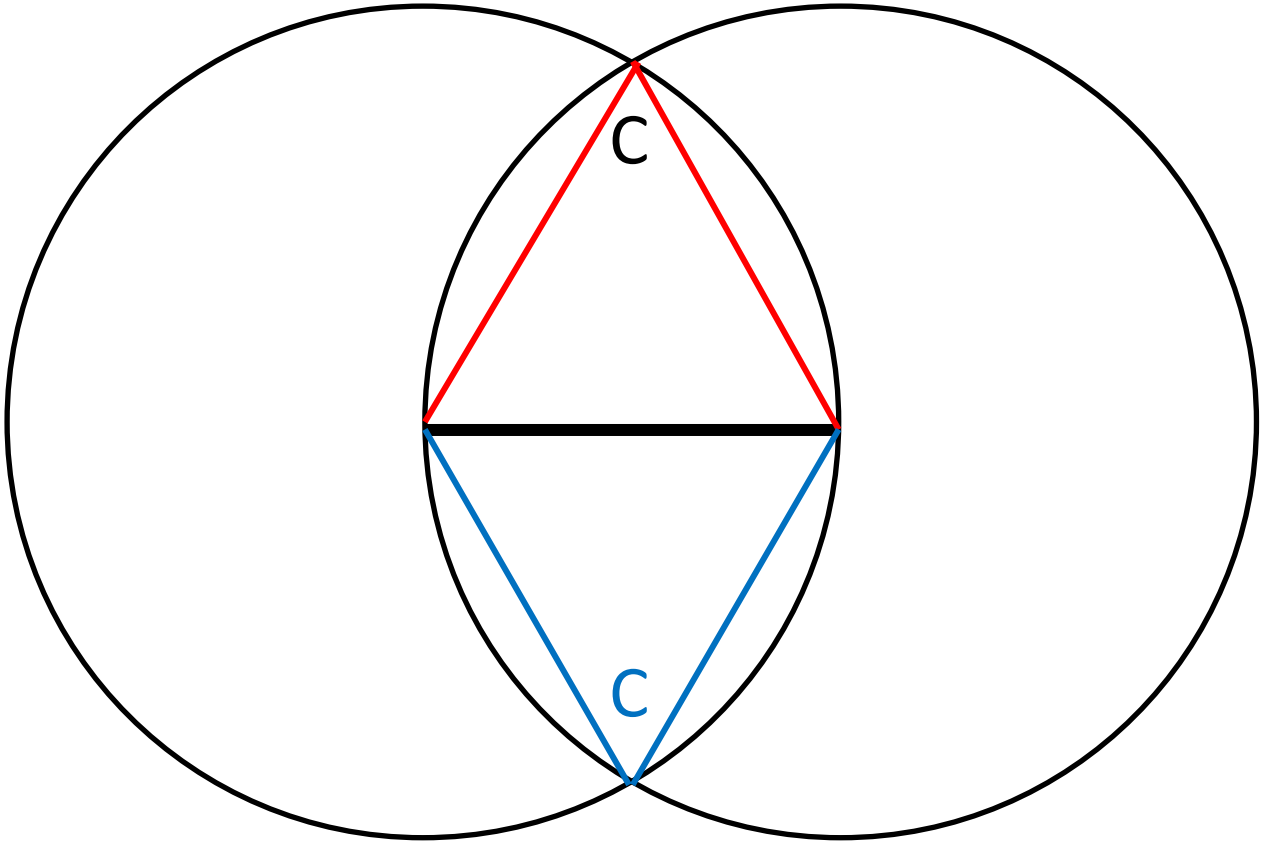
When P , (if B then V1 else V2) ensures R

Repetition

- When P and notB , V ensures P

When P , (repeat V until B) ensures (P and B)

Non-determinism



Syllogism for non-determinism.

- When P , V1 ensures R

  When P , V2 ensures R

When P , (V1 orelse V2) ensures R

Question 3

- How do we know whether the computer understands something?

The Turing test

- by conversation between tester and machine
- on arbitrary topics of human interest
- for five minutes.

- The machine fools the tester that it is human
- on 30% of the tests

An Engineering Version of the Test

- by interactive examination of the machine
- on the single topic of its own program
- for as long as desired.

Criterion of understanding

- The machine gives an answer to most questions
- and explains the reasoning behind it.
- All the answers are correct.
- most of them are relevant and useful
- to the software engineer writing the program.

Typical questions

- Can the program overflow a buffer?
- If so, give an execution that reveals the error.
- Generate test cases that exercise all the changes recently made to the program.
- Can this change make the program slower?
- Are all assertions made in the program valid?
- Could the airplanes under control by the program ever collide?

The Intelligent Programmer's Assistant

- Within 50 years, a design automation system for software engineering
- will be widely used by programmers
- in the analysis, design, programming, testing, delivery, and subsequent improvements
- of the ubiquitous software of tomorrow.

J Moore

Collaborative program development

- Human understands
  - the real world
  - the needs of program users
  - the commercial opportunities
- Computer understands
  - the consequences of human decisions

  in the context of a large and complex program.

The programmer will complain

- ‘The computer doesn’t understand what I want my program to do.
- It only understands the easy part of my job,
- and sometimes not even that’

Analogies

- Computers understand
  - by analogy with human understanding
- Airplanes fly
  - by analogy with the flight of birds
- Why don’t submarines swim
  - by analogy with fishes?

Alan Turing

- The language in which one communicates with these machines…forms a sort of symbolic logic….

Speculation

- The logic of programming is the logic of action in space and time.

Speculation

- Why did it take so long for a logic of change to be developed by philosophers and logicians?

Isaac Newton

Communication with Richard Gregory (1694)

“Our specious algebra [of fluxions] is fit enough to find out, but entirely unfit to consign to writing and commit to posterity.”

Sample questions

- What are its properties?
- How does it behave in its environment?
- What is its internal structure?
- What is its purpose?
- Why do we believe the answers to the preceding questions?

low level of understanding

- [3] A.M. Turing, On computable numbers, with an application to the Entscheidungs-problem, Proc. London Math. Soc, Ser 2 Vol 42 (1) 230-265 (Nov 1936).

Summary

- Computer Science is part of the enduring cultural and intellectual history of mankind.
- It has developed and exploited the ideas of the greatest thinkers of the past.
- It is developing new ideas that will influence the thinking of future generations
- And empower them to solve the many problems which we know that they will face.