

Applied Logic

Tony Hoare

Lausanne

20 June 2011

Historical Survey

Discovery

- deductive logic
- constructive logic
- temporal logic
- algebraic logic

Application

philosophy

geometry

theology

calculus

Attribution

Aristotle

Euclid

Occam

Leibnitz

A flight of imagination.

- deductive logic
 - + constructive logic
 - + temporal logic
 - + calculational logic
 - + algebraic logic
- + intuitionistic logic
 - + relevance logic
 - + deontic logic
 - + types
 - + spatial logic ...

= programming logic

Programs as logic

- Programs describe time and space
 - including change and motion
 - and history and geography
- A programming language extends the range of logic to include these topics.
 - repaying the part of the debt that computing owes to logic.

Deductive logic

- Aristotle (384-322 bc, Athens)
- Father of classificatory Biology,
and deductive Logic,
- and grandfather of Computer Science.
- Applications : biology

Grammar

Let S stand for the subject of a sentence,

Let P stand for the predicate.

- 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

24 Syllogisms

Barbara

(Major premise)	All S are M	(a)
(Minor premise)	<u>All M are P.</u>	(a)
(Conclusion)	All S are P.	(a)

Celarent

	No M are P	(e)
	<u>All S are M</u>	(a)
	No S are P	(e)

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 24 syllogisms and the first two lines occur earlier in the proof.

- The definition is independent of the subject matter of the proof.

Example from Biology

Barbara

(Major premise)	All sharks are selachians	(a)
(Minor premise)	<u>All selachians inhabit the sea.</u>	(a)
(Conclusion)	All sharks inhabit the sea.	(a)

Celarent

No selachians are fish	(e)
<u>All rays are selachians</u>	(a)
No rays are fish	(e)

Application to biology

- Classificatory biology provided observational data for application of syllogisms.
- Today, far more observational data, from all branches of science, is held on computers
- Scientists' questions are answered automatically by deductive logic.

Program execution

- Programming languages are defined by syntax
- The steps in the execution of a program are defined by rules.
- The validity of an execution is independent of the application of the program

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.

Constructive logic

- Euclid (c. 300 bc, Alexandria)
- Father of the geometry of space,
- with applications to measurement of land, definition of boundaries, surveying, mapmaking, navigation, astronautics,...
- and now to graphic programming languages, computer displays, animated films, ...

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!
- They are written in a programming language
 - with assignments, sequencing,
 - subroutines, parameters,
 - preconditions, postconditions,...
- and proofs!

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

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 equal sides.

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

Five common notions

- include general purpose reasoning principles
 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. ...subtracted...
 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

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 calling proof

Subroutines

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

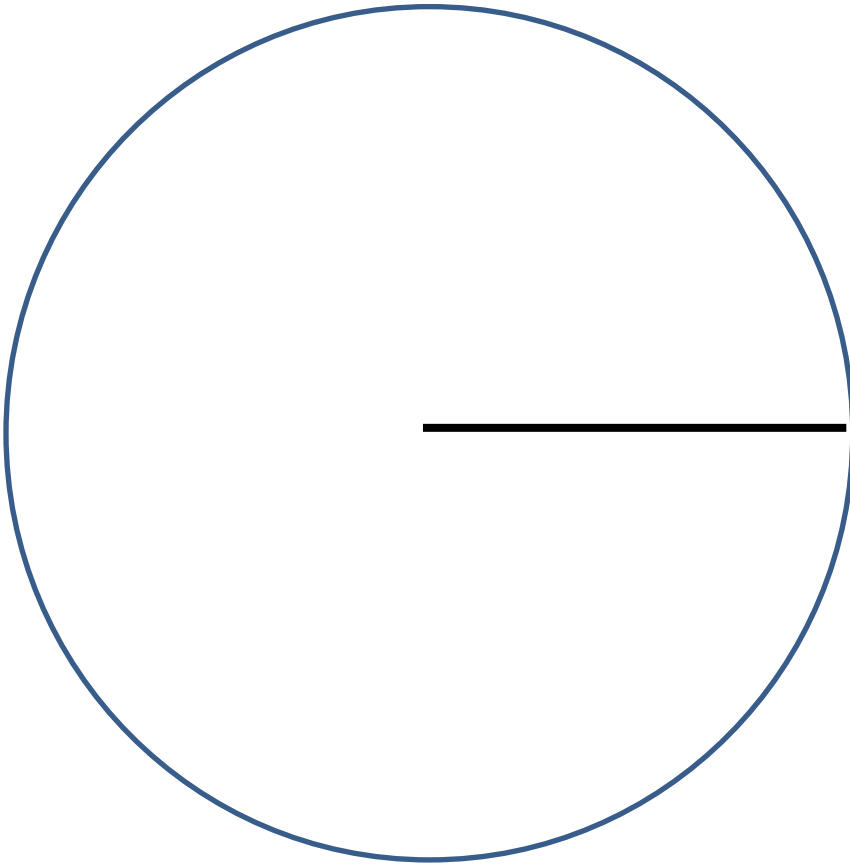
The proven properties of the construction can be used as assumptions of the calling proof

1. To construct an equilateral triangle

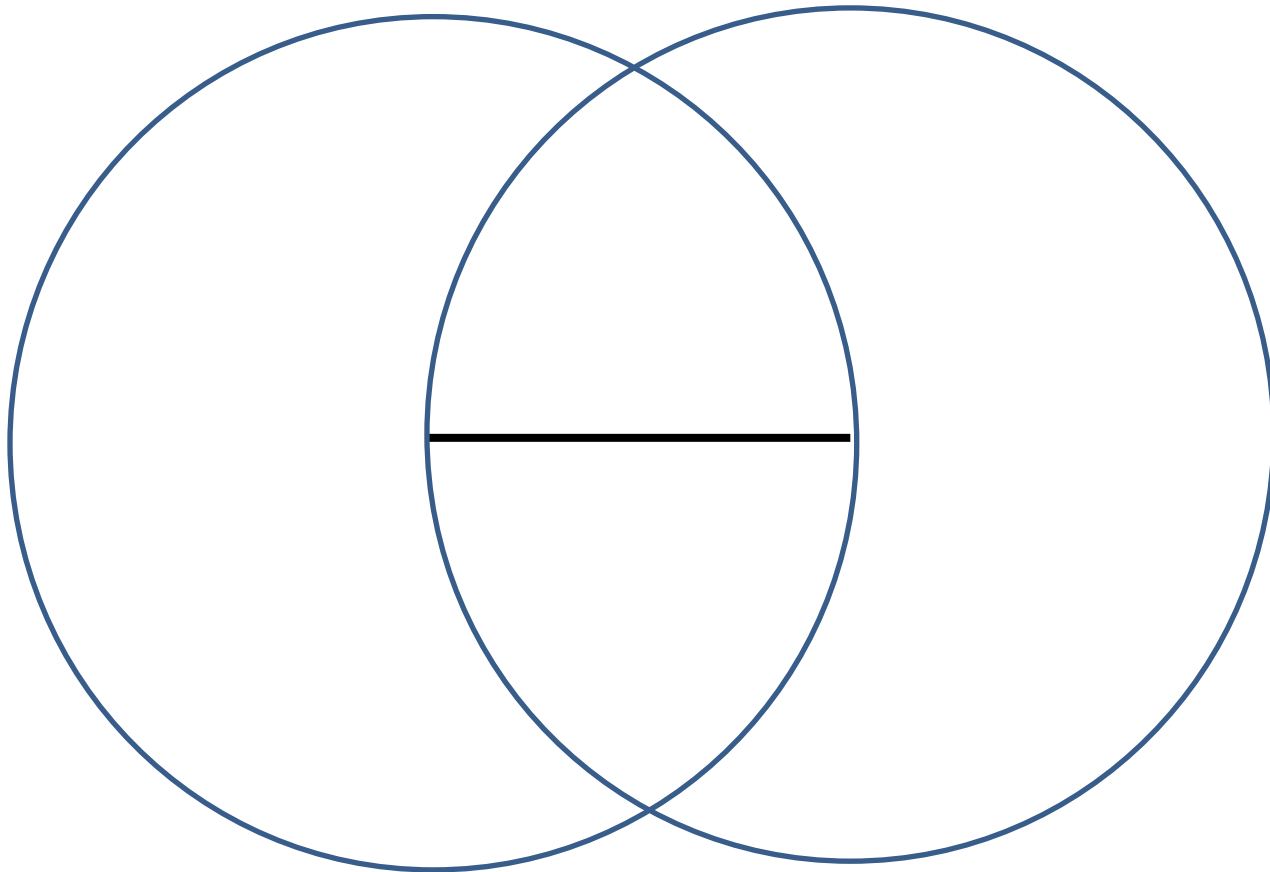
with a given side



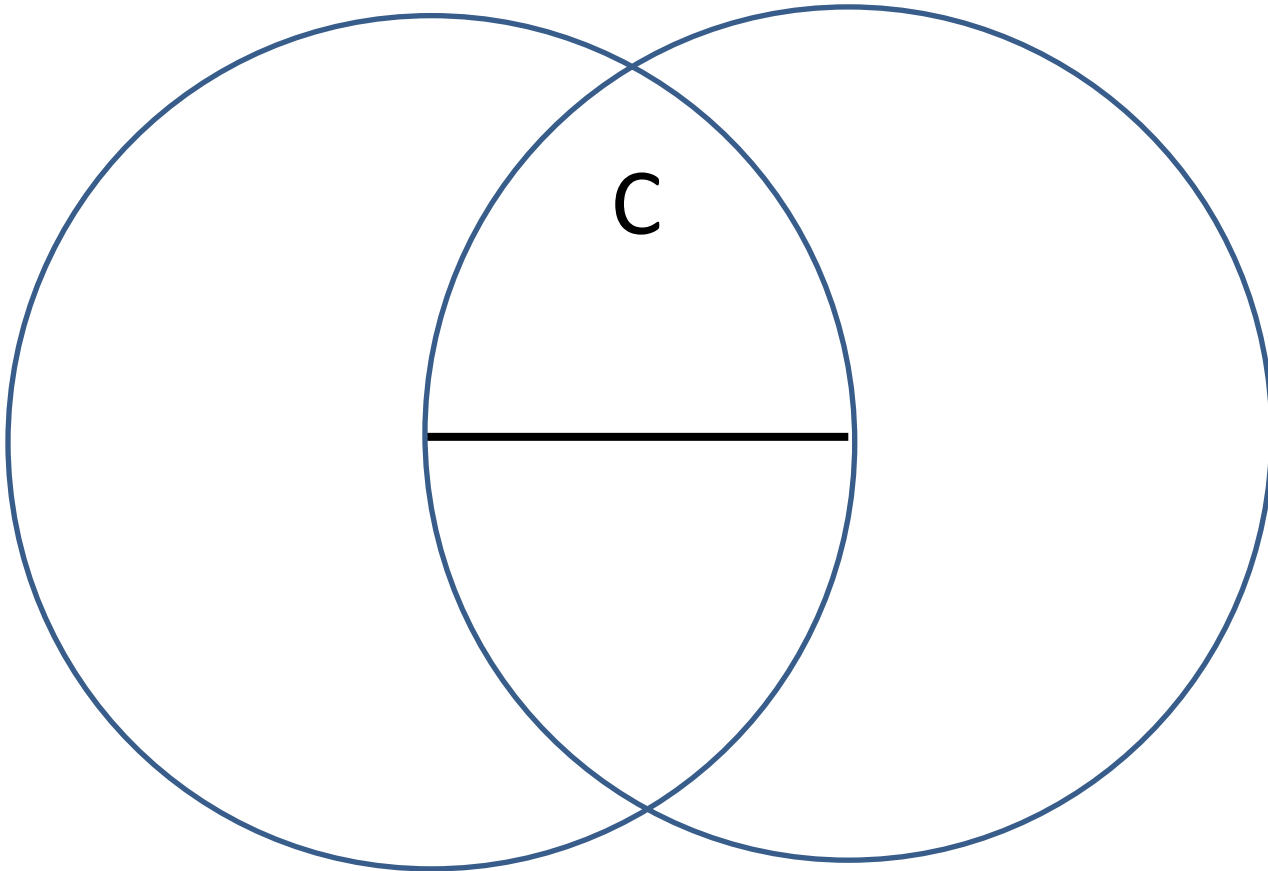
Draw a circle with the line as radius and centre at one end (postulate 3).



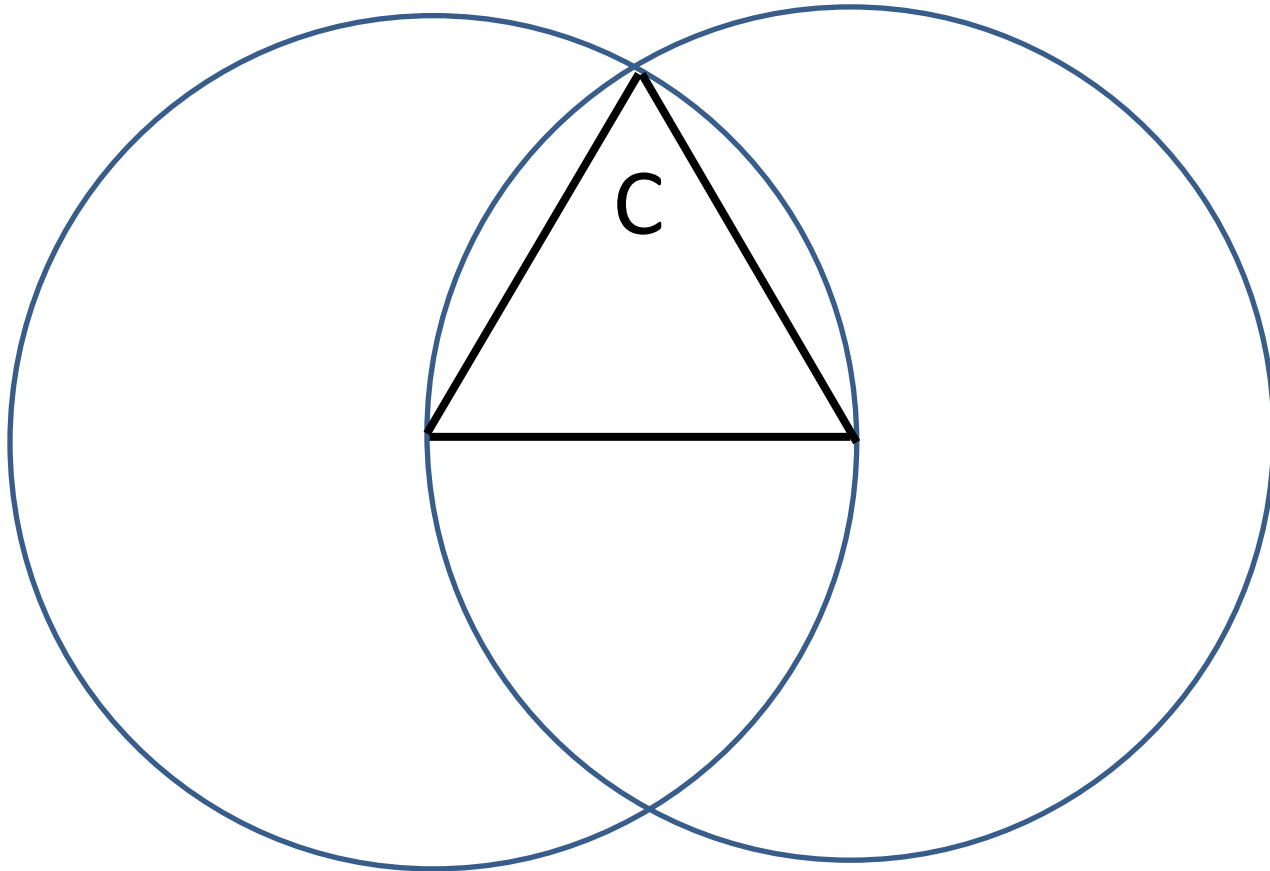
Then draw a circle with the line as radius and centre at the other end



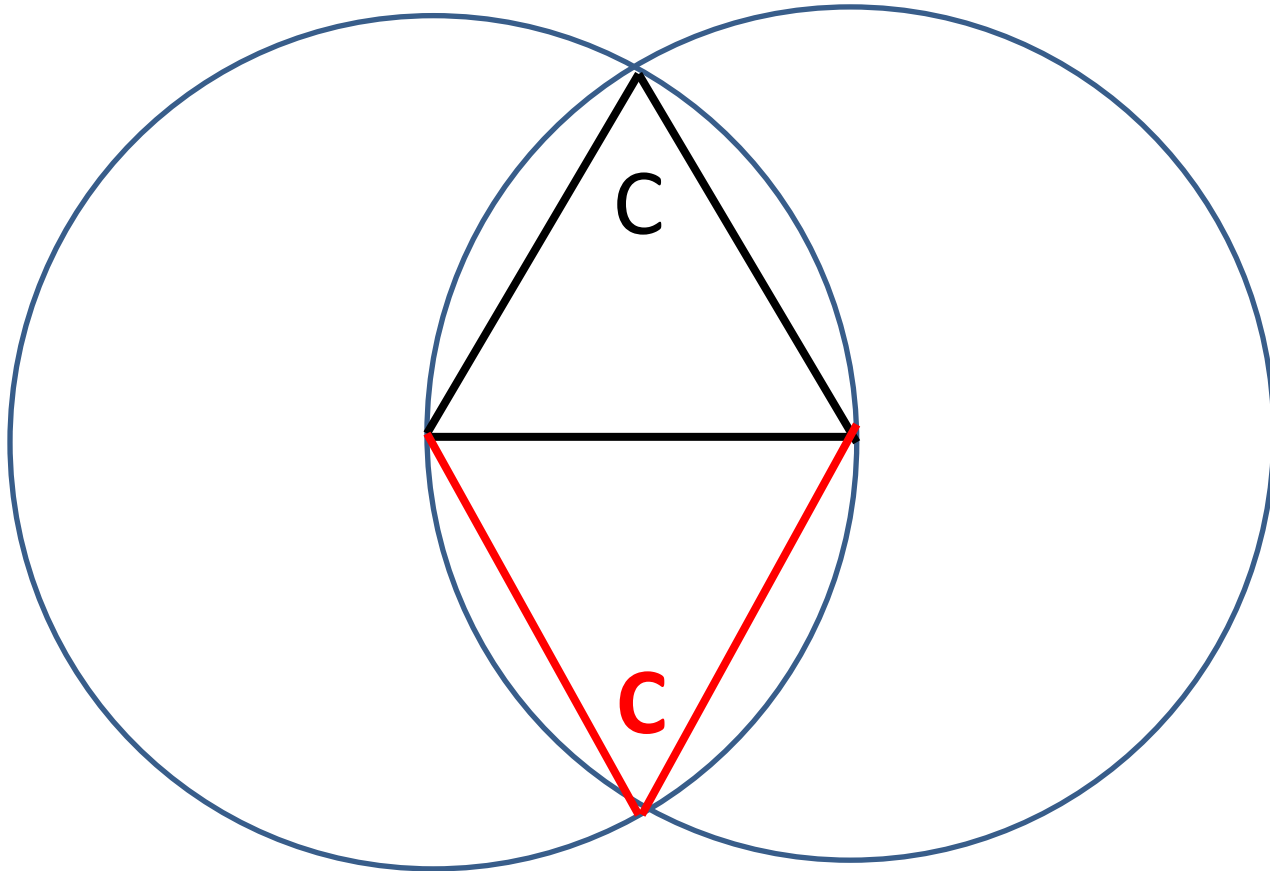
Then choose a point C where the two circles intersect each other




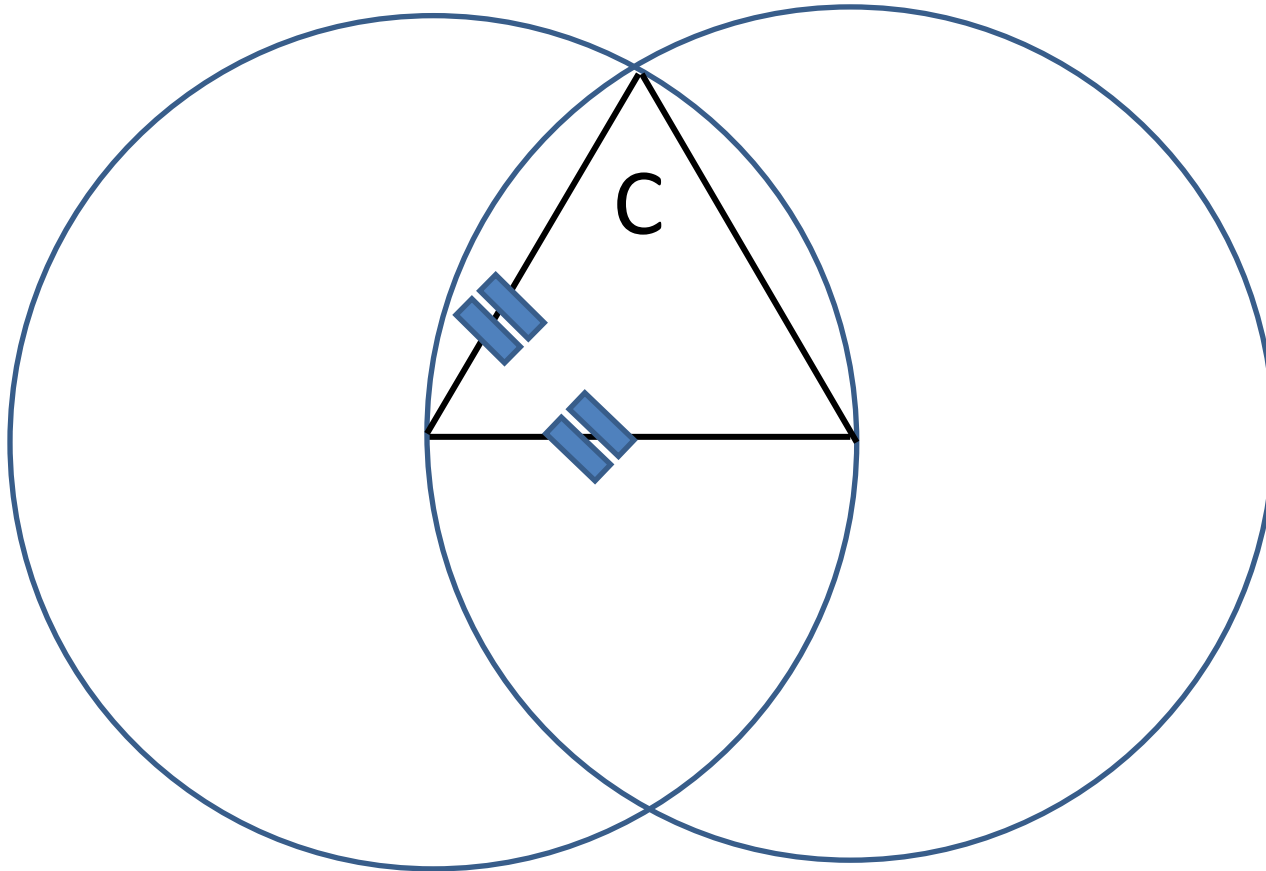
Then draw a line from C to each end of the given line (Postulate 1, twice)



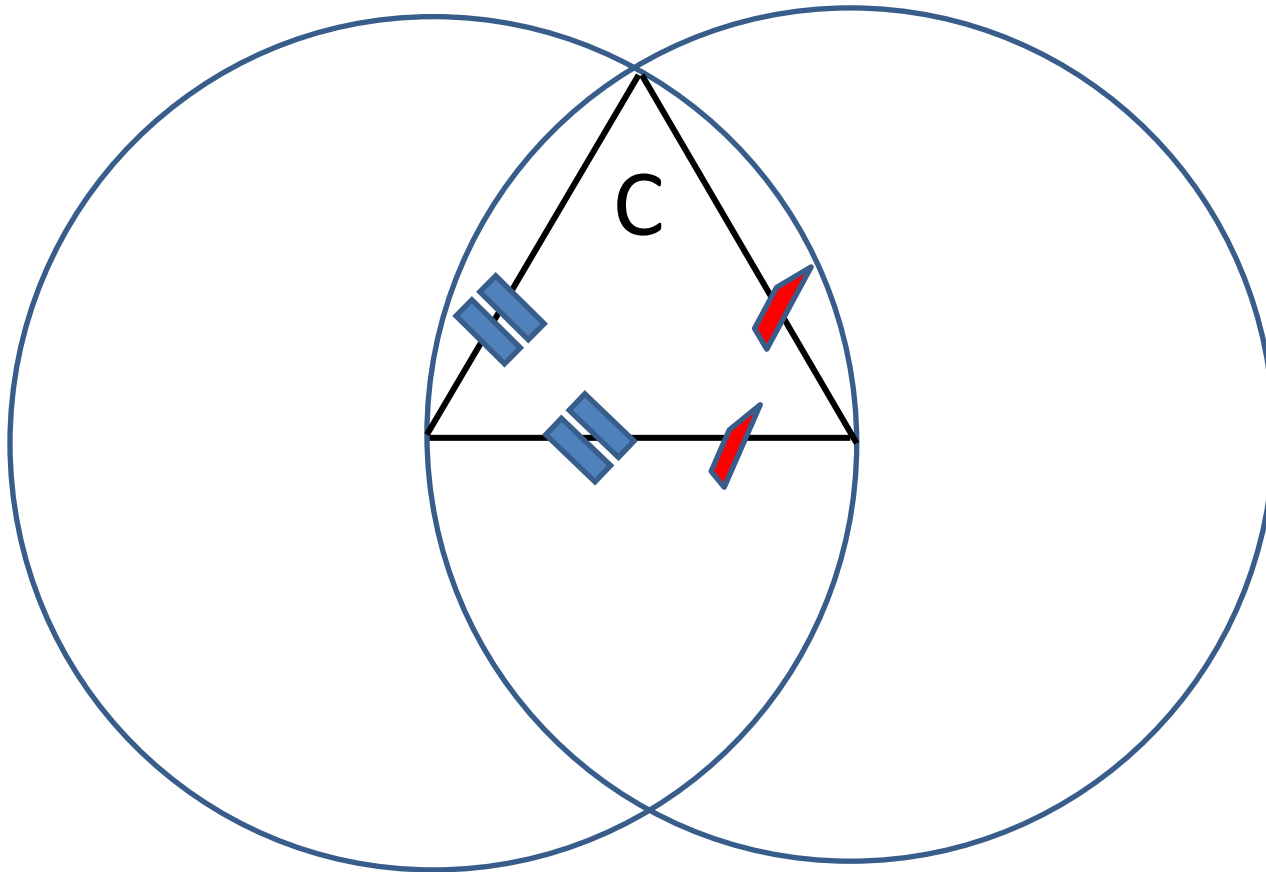
Non-determinism



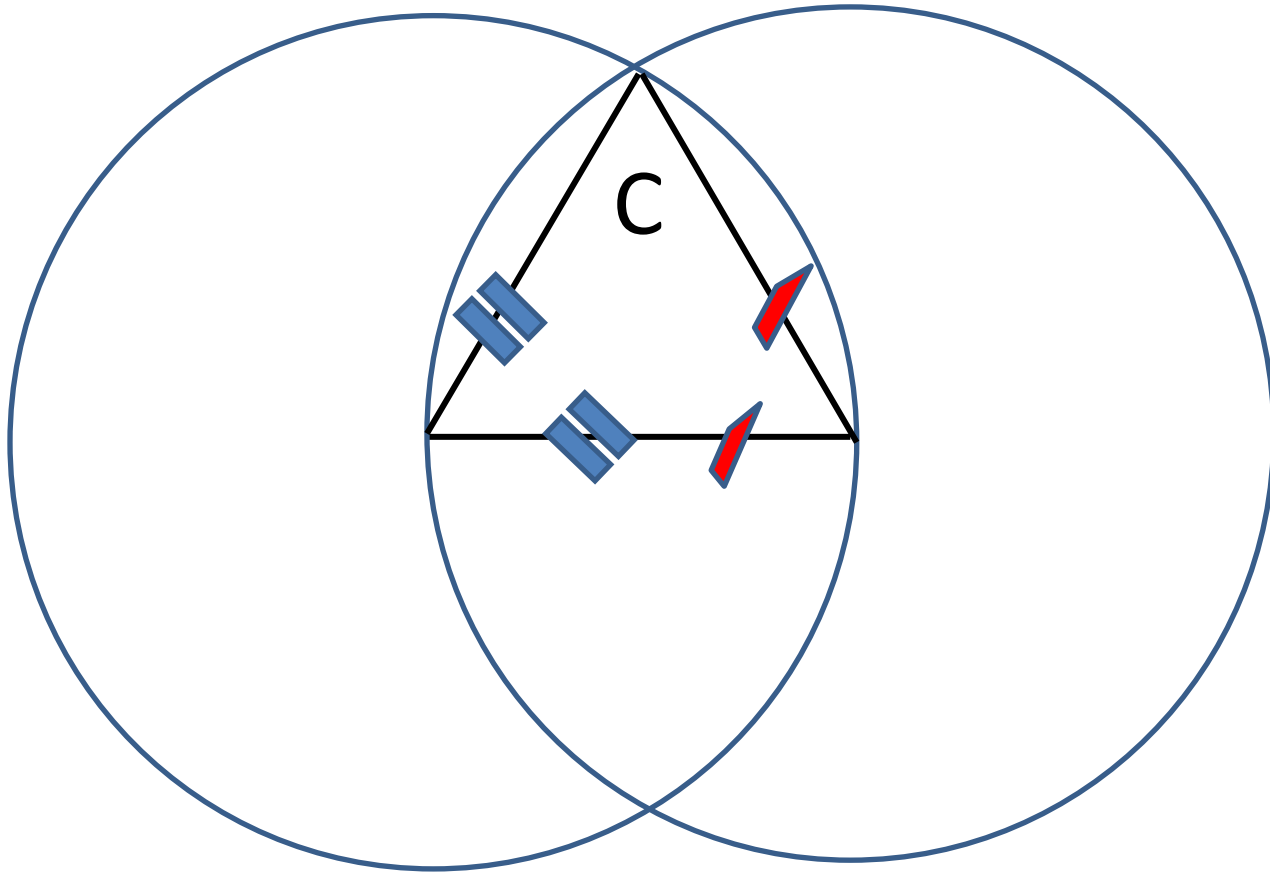
The lines marked  are equal,
being radii of the left circle (Def. 15)



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



The triangle is therefore equilateral
(Def 20, common notion 1) Q.E.D.



Summary

- primitive (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'

Temporal logic

- . William of Ockham (1287 – 1347)
- author of: Summa Logicae....
 De Praedestinatione et futuris contingentibus
- Application to theological paradoxes, e.g.
 Refutation of the argument:
 God knew, from the very beginning,
 what the future holds for man
 Therefore man has no free will.

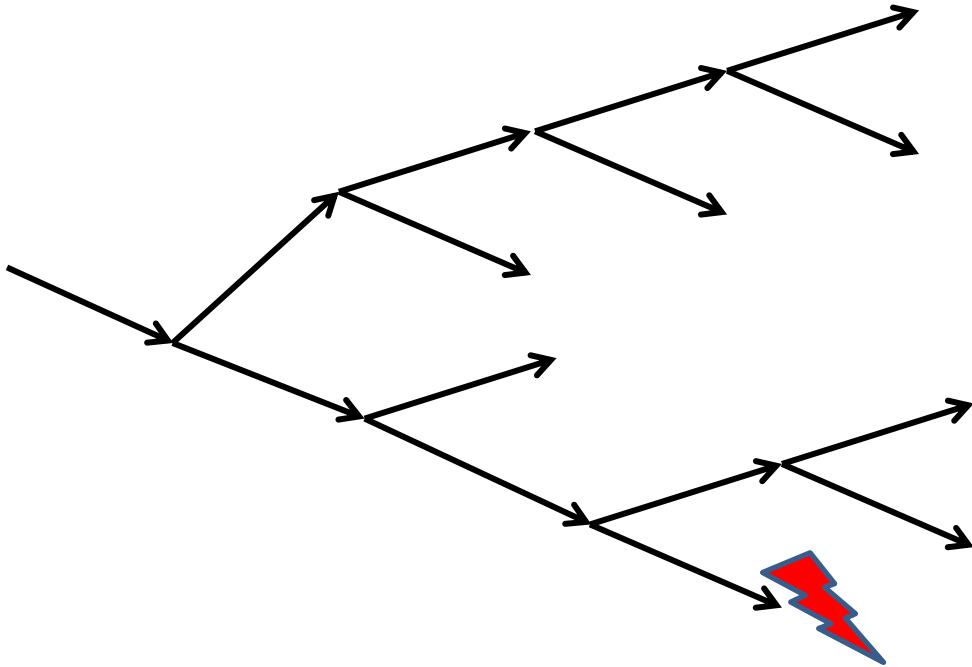
Analogy

- The programmer who wrote a program knows exactly what the program is going to do
- Therefore the implementation of the programming language has no free will
- But a program can be non-deterministic.

Non-deterministic programs

- The programmer knows the whole branching tree of possibilities of program execution.
- The implementer/user of the program has choice at branch points.
- Therefore the implementer/user has (limited) freewill.

Branching Time



Jonah 3 4-5, 10

- And Jonah began to enter into the city a day's journey, and he cried, and said, Yet forty days and Nineveh shall be overthrown
- So the people of Nineveh believed God, and proclaimed a fast, and put on sackcloth, from the greatest of them even to the least of them. ...
- And God saw their works, that they⁷ had turned from their evil way; and God repented of the evil, that he said that he would do unto them; and he did it not.

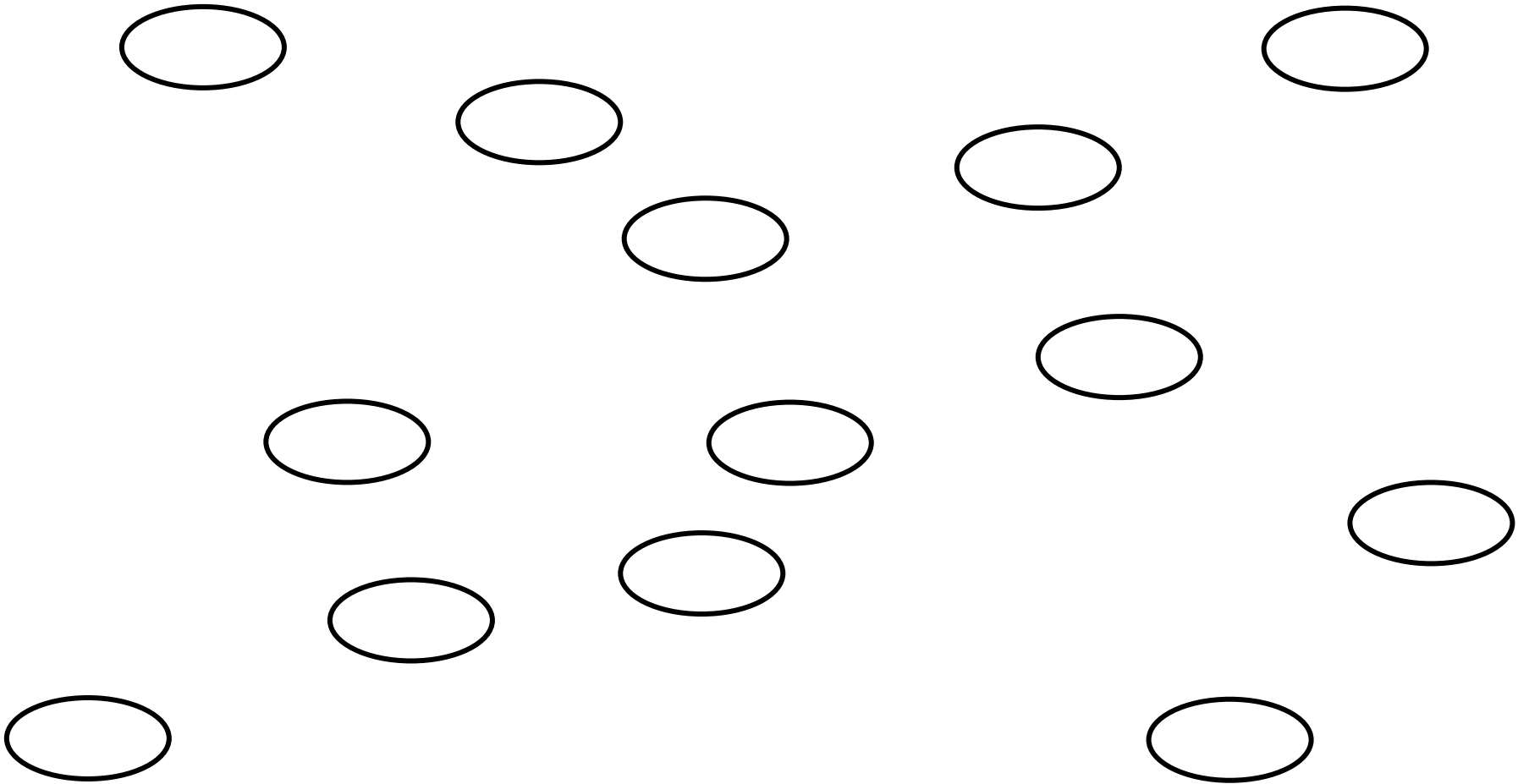
Ockham's logic.

- Let P and Q be clauses or sentences.

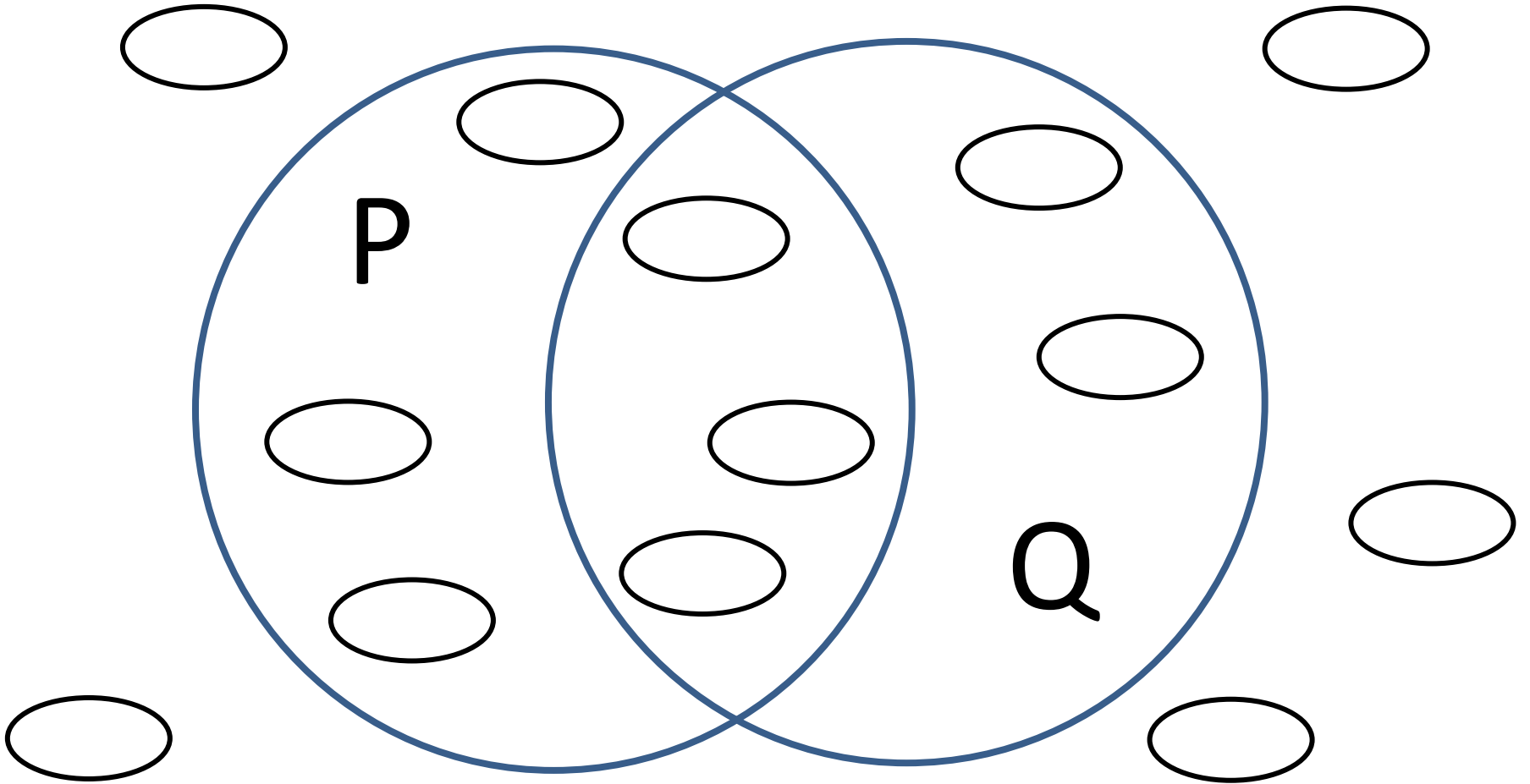
Then so are:

- P **if** Q conditionalis
- P **&** Q copulativa
- P **or** Q disiunctiva
- P **because** Q causalis
- P **with** Q temporalis

Program executions

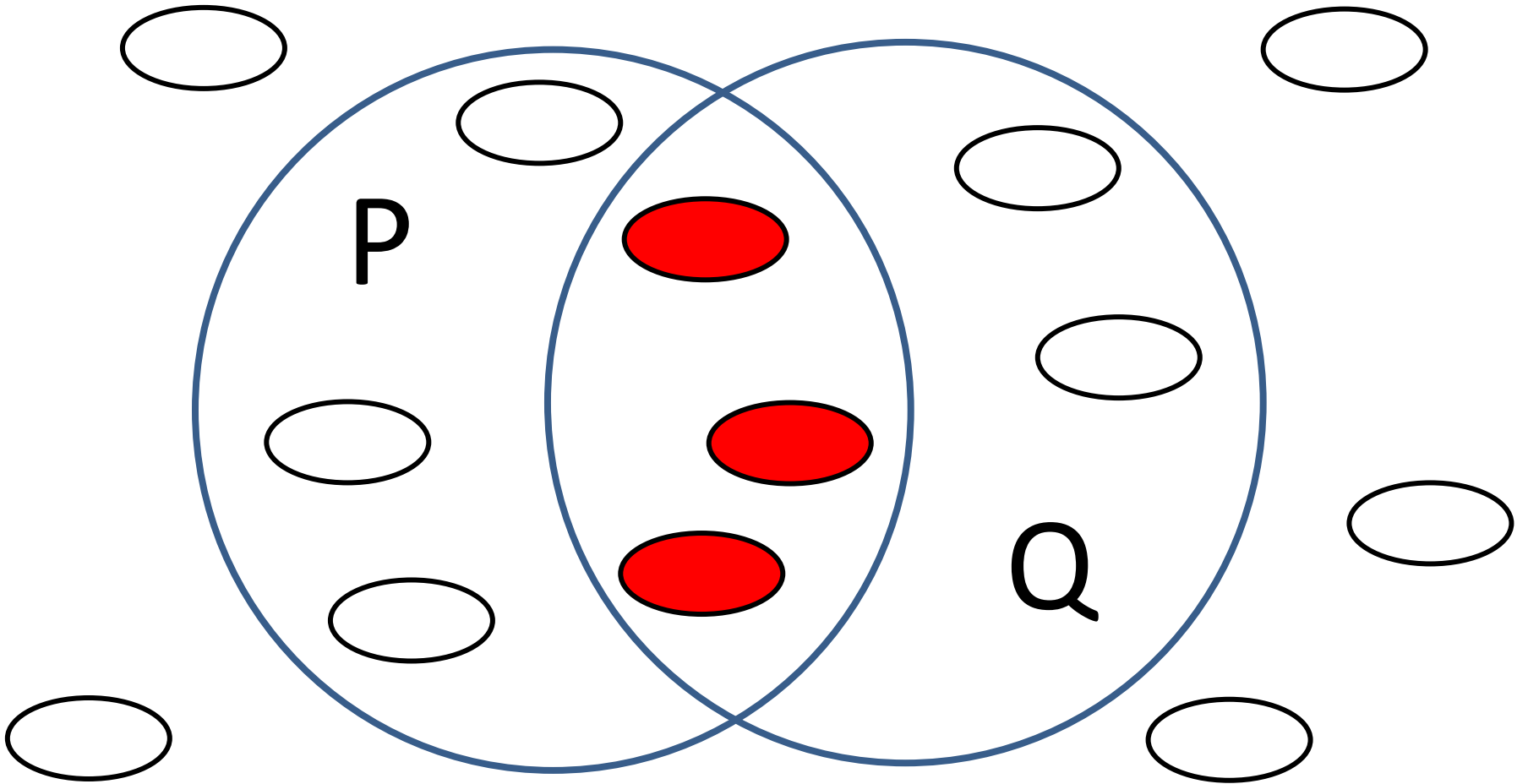


Programs P Q



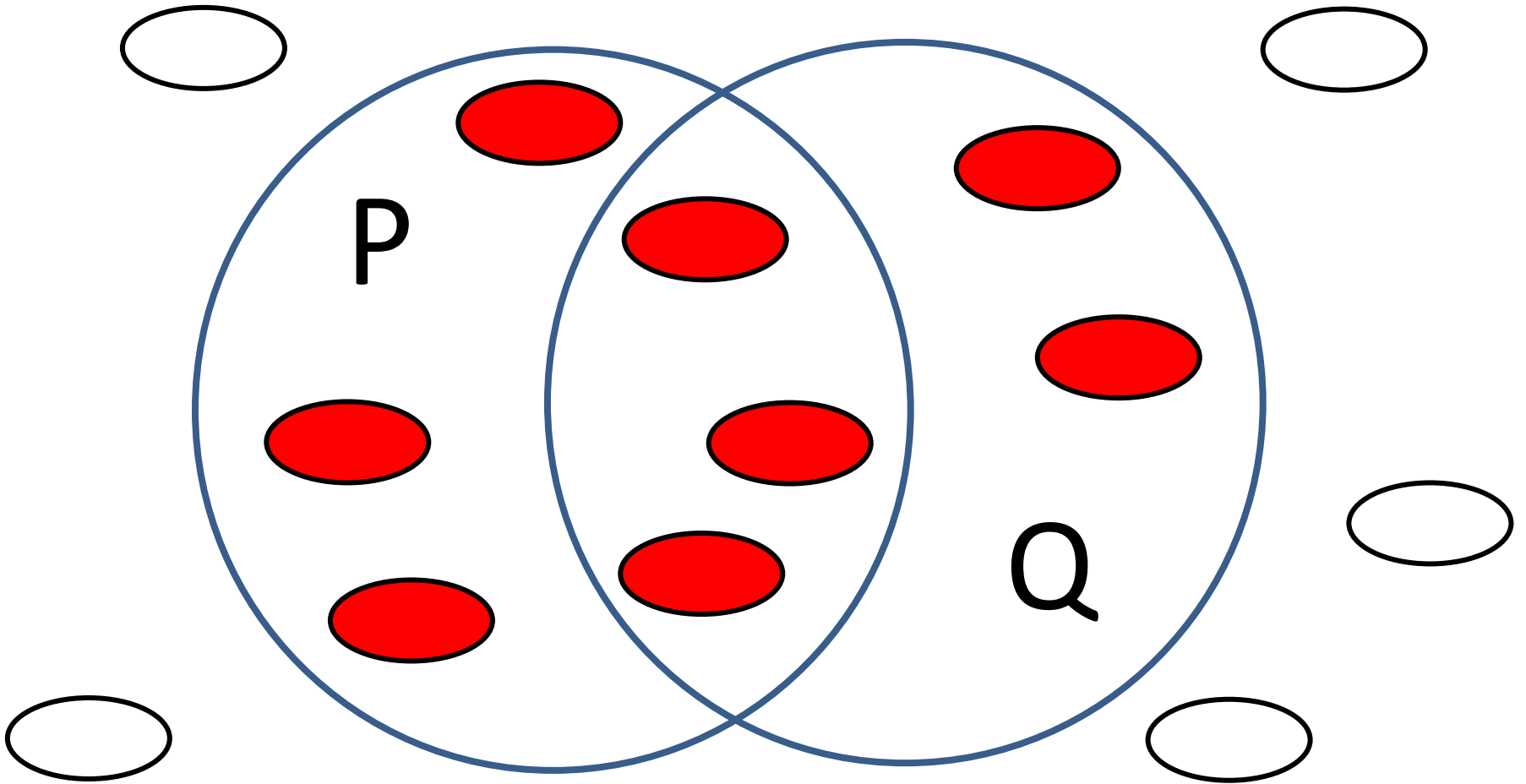
P & Q

(Conjunction, Ockham's copulativa)



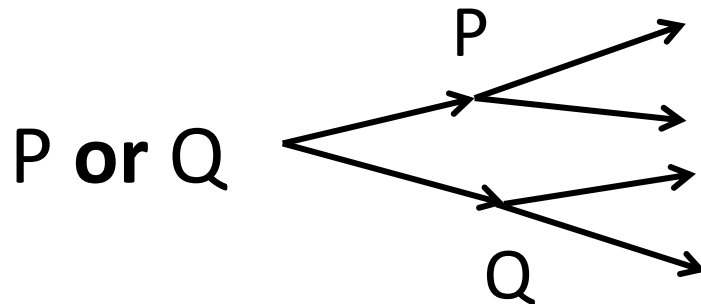
P or Q

(Disjunction, Ockham's disjunctiva)

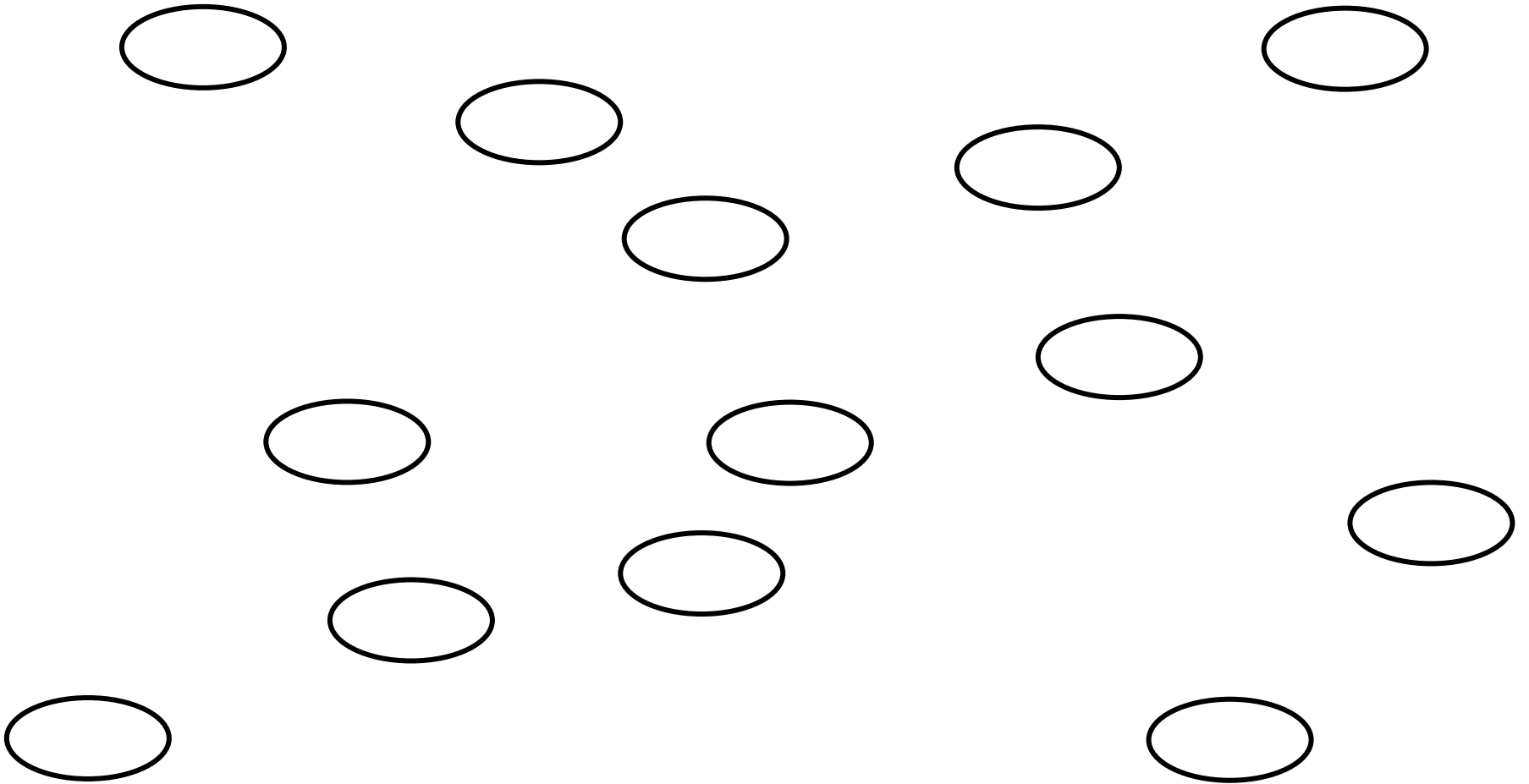


Non-determinism

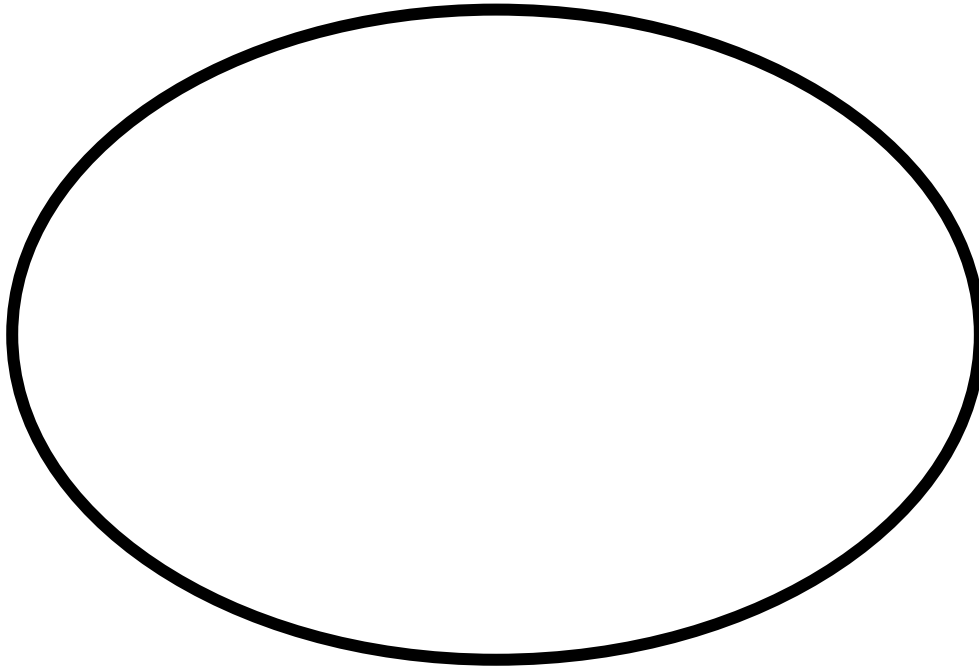
- **P or Q** is a non-deterministic program
 - behaving either like P or like Q
 - the programmer does not know which.



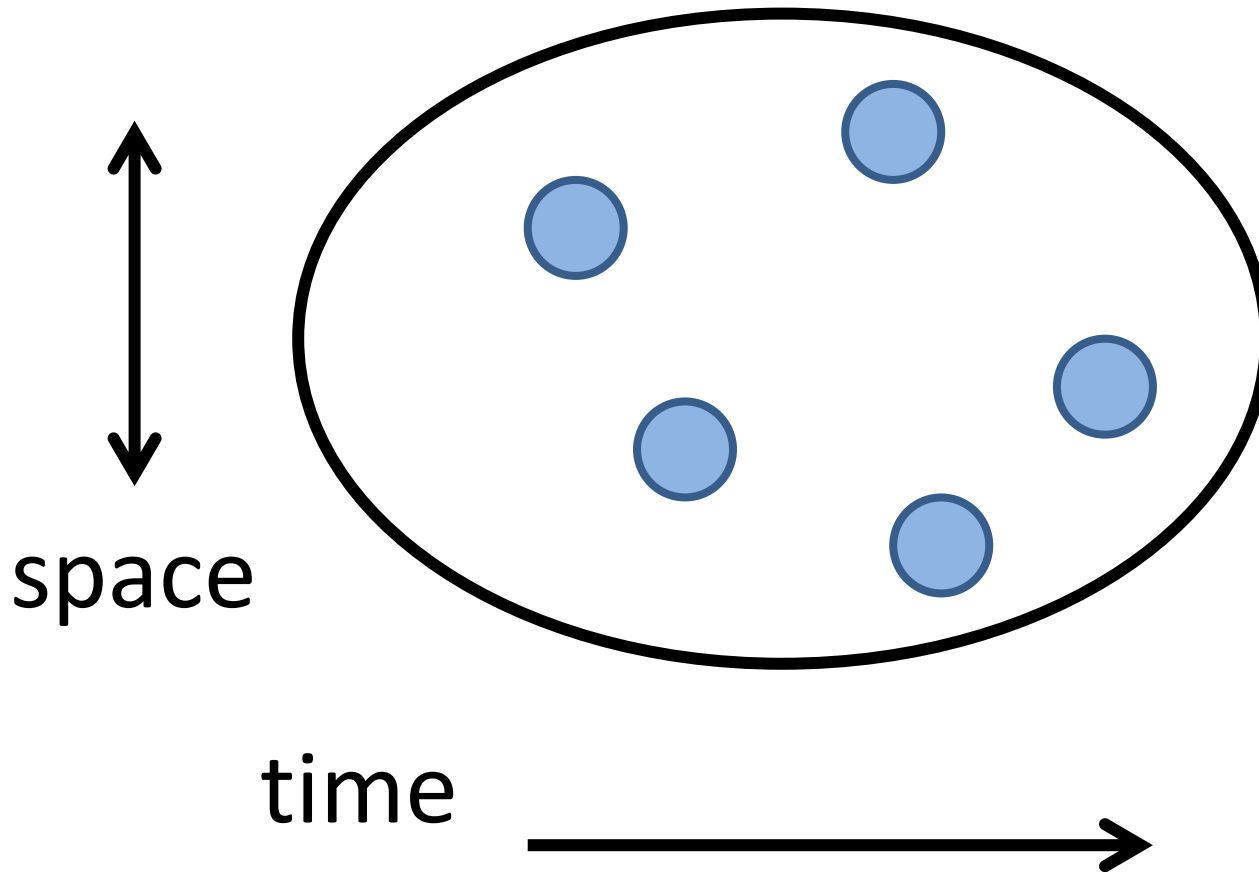
Executions



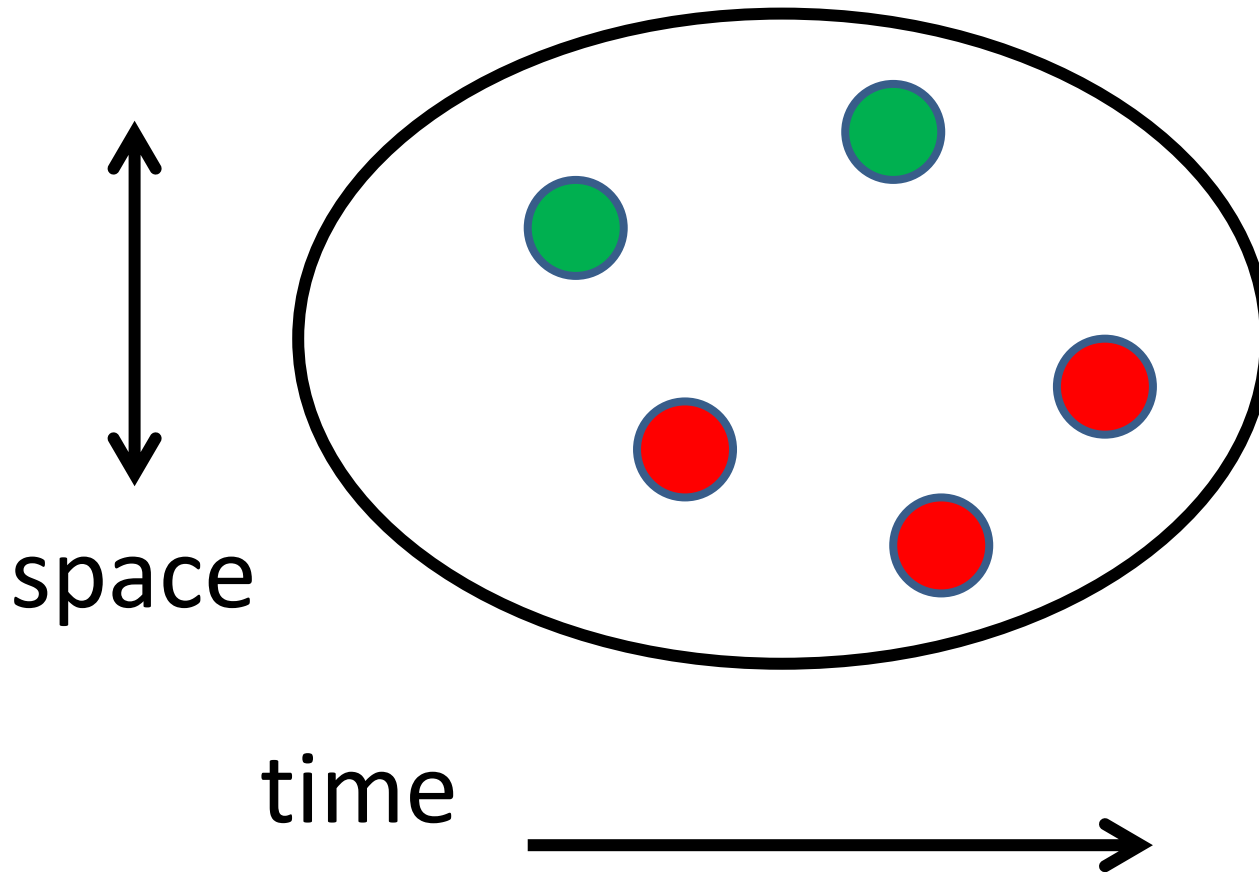
An execution



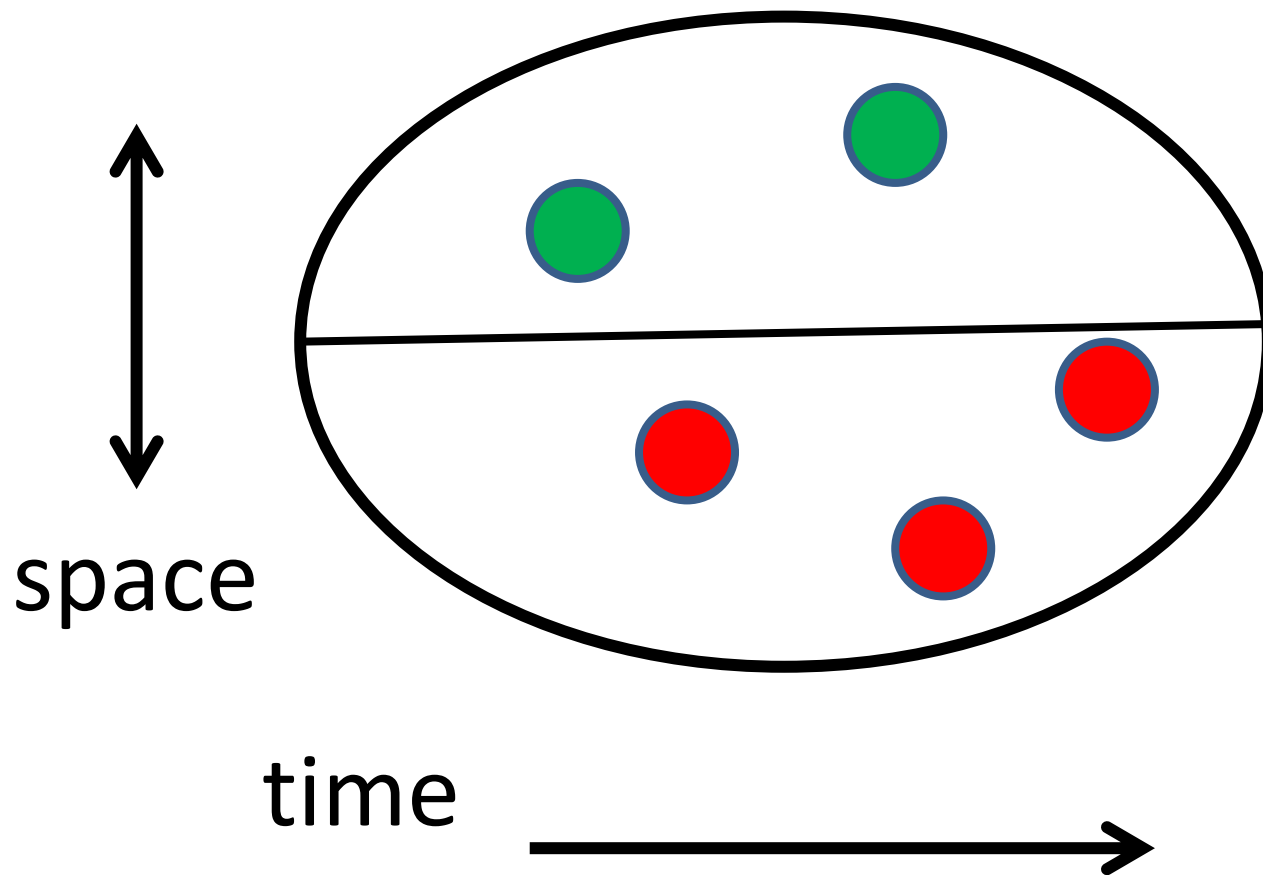
An execution with five events



which are either green or red

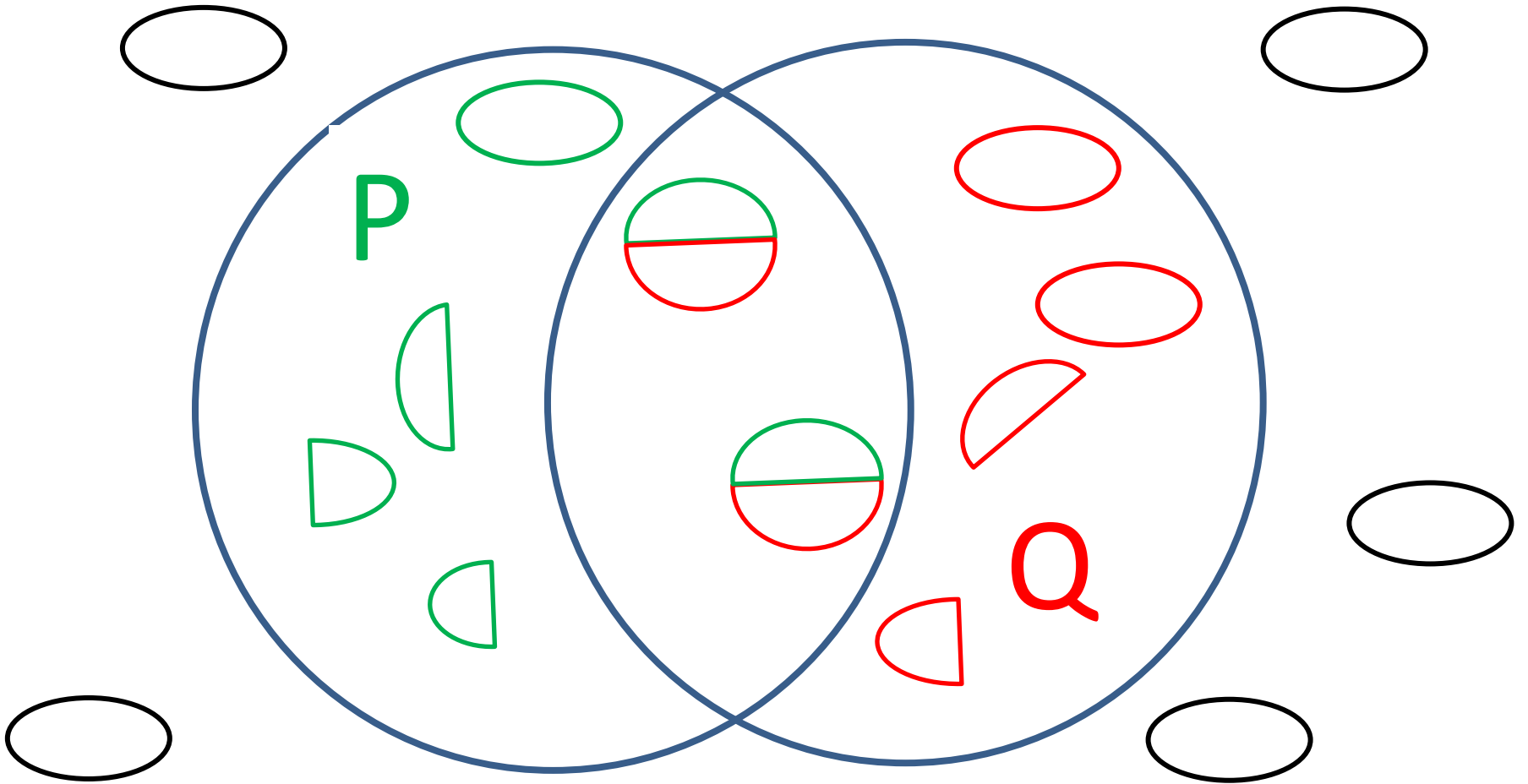


Horizontal split

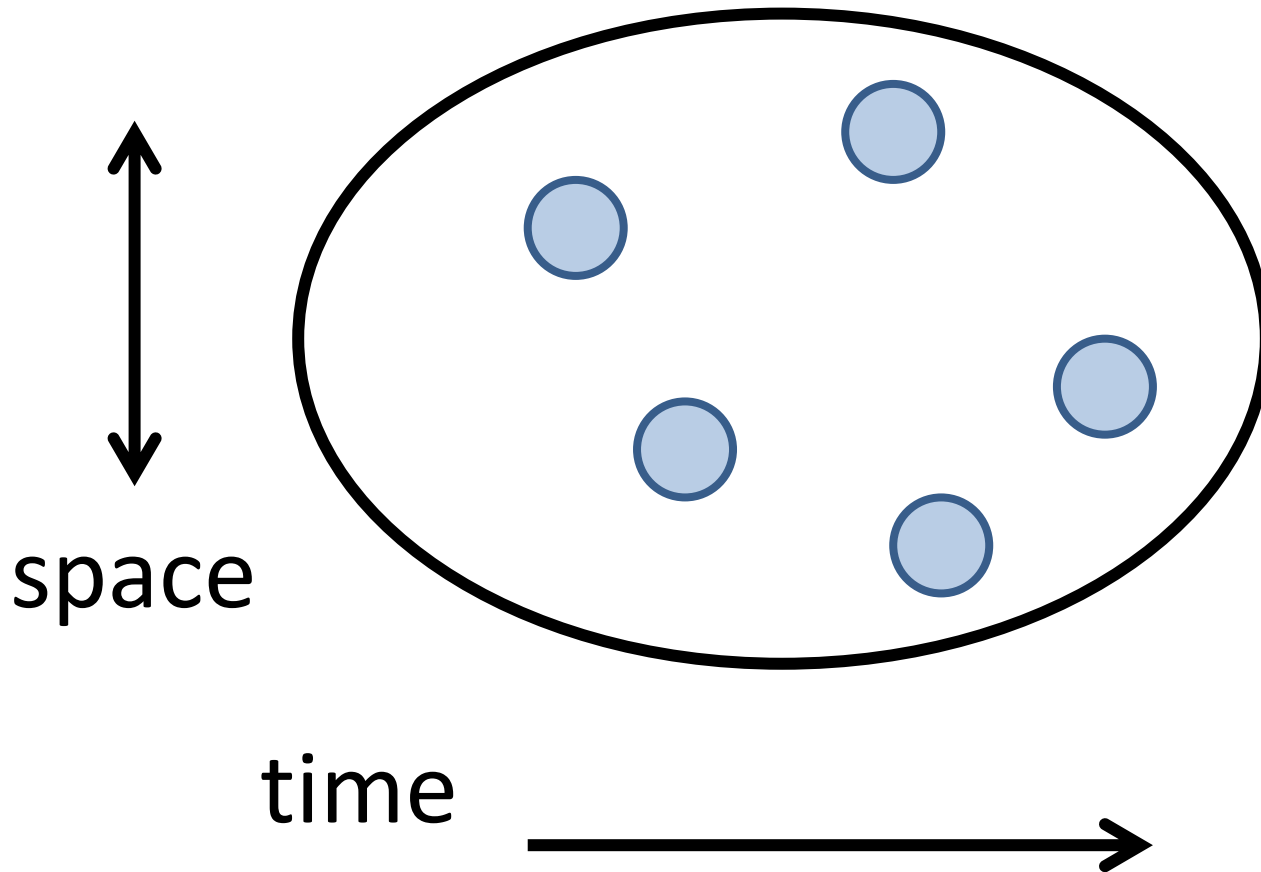


P with Q

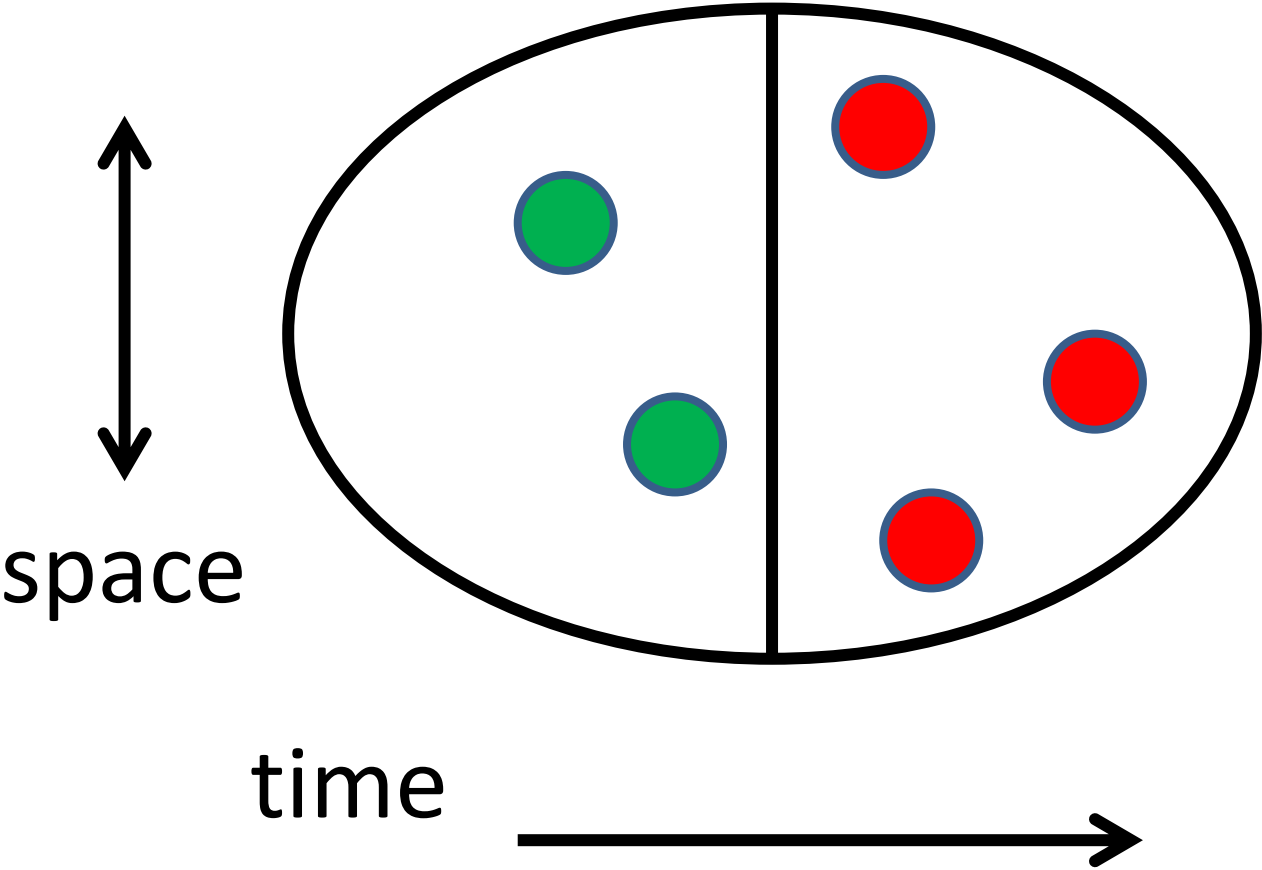
(Ockham's temporalis)



An execution with five events

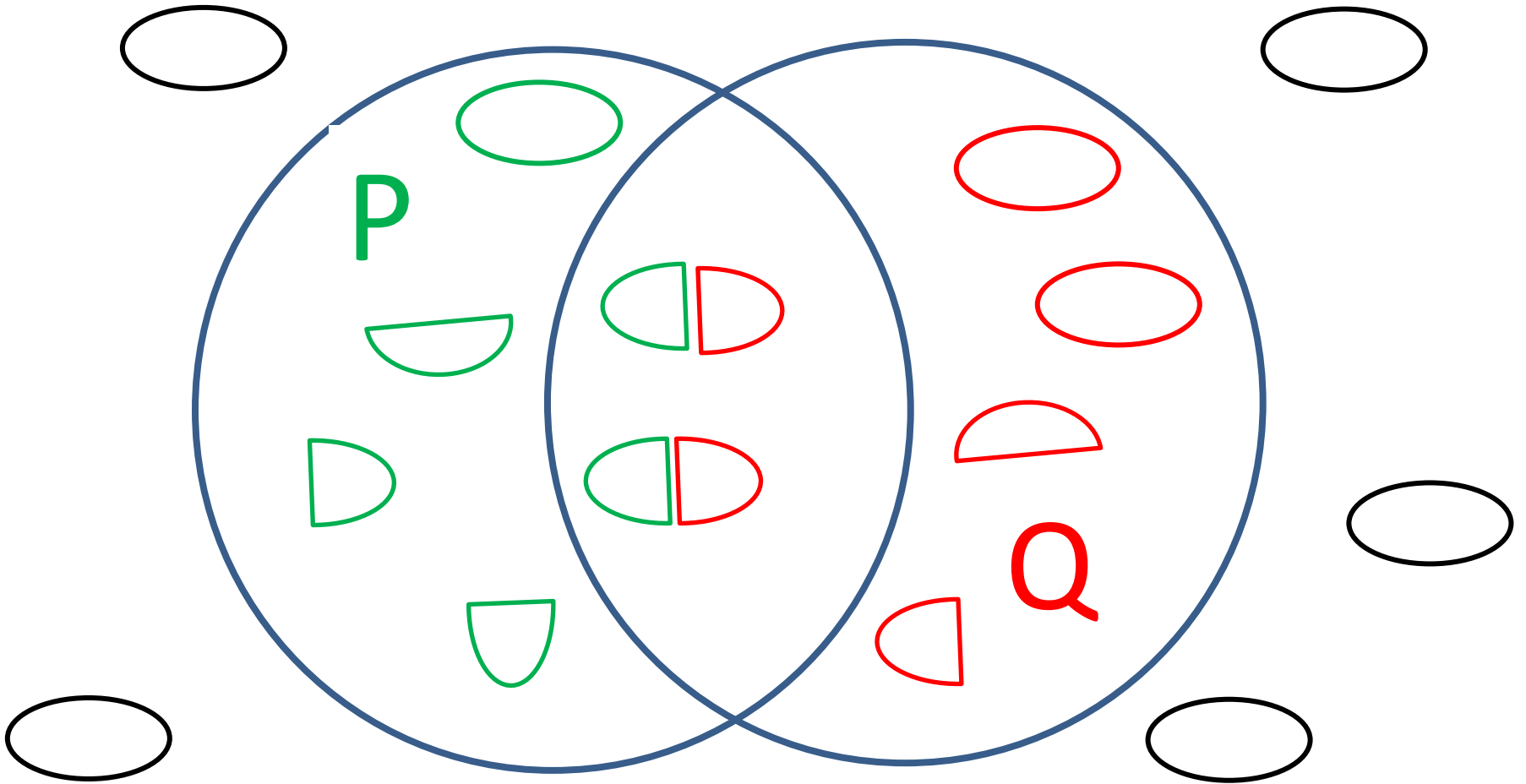


An execution with five events



$P ; Q$

(P then Q, sequential composition)



Every program tells a story

- about the internal events
 - occurring inside the computer
 - while it is executing the program.
- Simulation programs describe external events
 - occurring in the real world outside the computer,
 - e.g., inside the living cell.

Proof by calculation.

- Gottfried Leibnitz (1646-1716)
- Applications (then) to differential and integral calculus
- Applications (now) : mechanised proof of mathematical theorems and properties of computer programs

The calculus

$$\int (6x^2 + 4xy - 2y^2) dx$$
$$= 3x^3 + 2x^2y - 2y^2x + K$$

$$\frac{\partial(3x^3 + 2x^2y - 2y^2x + K)}{\partial x}$$

$$= (6x^2 + 4xy - 2y^2)$$

The calculus

$$\begin{aligned} & 3 \times 2^3 + 2 \times 2^2 \times 5 - 2 \times 5^2 \times 2 + 1 \\ &= 3 \times 8 + 2 \times 4 \times 5 - 2 \times 25 \times 2 + 1 \\ &= 24 + 40 - 100 + 1 \\ &= -37 \end{aligned}$$

$$\begin{aligned} & \frac{\partial(3x^3 + 2x^2y - 2y^2x + K)}{\partial x} \\ &= 3 \frac{\partial x^3}{\partial x} + 2 \frac{\partial x^2y}{\partial x} - 2 \frac{\partial y^2x}{\partial x} + \frac{\partial K}{\partial x} \\ &= (6x^2 + 4xy - 2y^2) \end{aligned}$$

Gottfried Leibniz (1646-1716)

- *characteristica universalis* - a notation that can express every truth.
- *calculus ratiocinator* – a formal method of reasoning by symbolic calculation.

A flight of imagination

- A programming language is a
characteristica universalis
- A programming logic is a
calculus ratiocinator.

Alan Turing

- I expect that digital computing will eventually stimulate a considerable interest in symbolic logic...
- The language in which one communicates with these machines...forms a sort of symbolic logic.

Acknowledgements

- Wikipedia (Aristotle)
- D.E.Joyce (Euclid)
- *Temporal Logic* (William of Occam)
- Martin Davis, *Engines of Logic* (Leibnitz)