MAA Meeting
Computable Mathematics

November 8, 2014

Valentina Harizanov
Department of Mathematics, GWU
harizanv@gwu.edu
http://home.gwu.edu/~harizanv/

Al-Khwarizmi

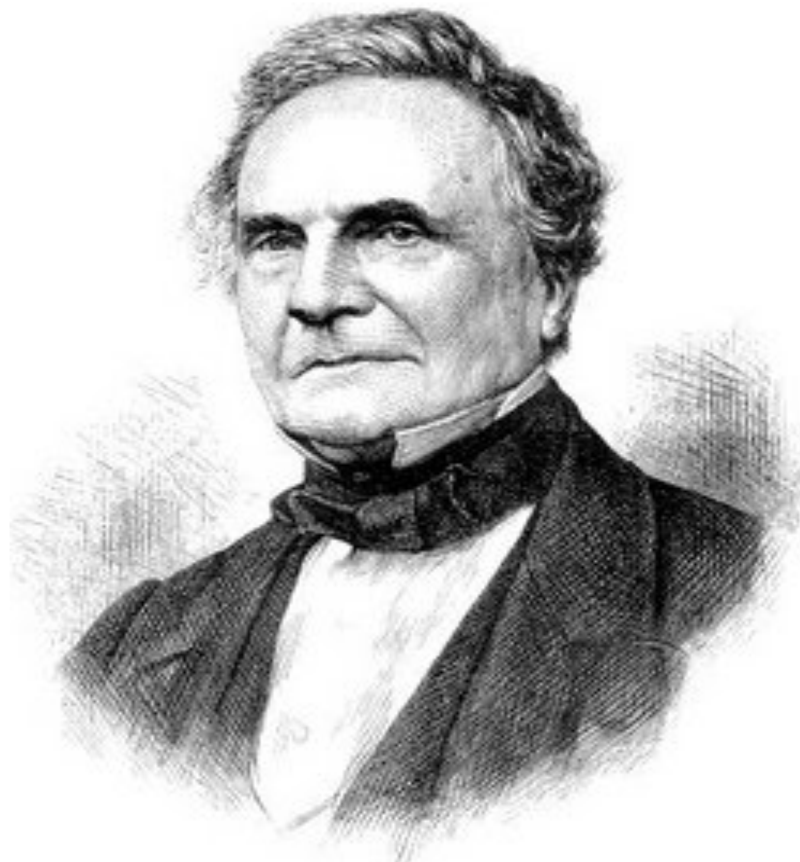
- Muhammad bin Musa **al-Khwarizmi** (early 9th century) from Khwarizm (Khiva)
- Further developed the concept of a mechanical procedure
- Systematic approach to soving linear and quadratic equations: "Hisab **al-jabr** w'al-muqabala"

An algorism/algorithm

- An ***algorism*** refers to the well-defined step-by-step rules performing arithmetic using Arabic numerals (positional notation).
- By the 18th century evolves into an ***algorithm***—any *finite* systematic well-defined procedure for solving many instances of a general problem.

Charles Babbage (1791– 1871) Implementing algorithms

- The idea of a programmable computer
- ***Difference engine***, 1822 (computing tables in math and astronomy)
- Construction not completed



Analytical engine

- Babbage: ***analytical engine***, 1833–1842 (arbitrary calculation)
- Ada Byron, Lady Lovelace (1815–1852), wrote several programs for the analytical engine
- Construction not completed (programs never executed)

Mathematical rigor missing

- Lack of the mathematical ***rigor*** in the definition of an algorithm (finite process, effective procedure), even in the 20th century.

- A. Church: "The notion of an effective process occurs frequently in connection with mathematical problems, where it is apparently taken to have a clear meaning, but this meaning is commonly taken for granted without explanation."

Mathematical Problems of David Hilbert (1862–1943)

- Lecture at the International Congress of Mathematricians in Paris in 1900
- “What new methods and new facts in the wide and rich field of mathematical thought will the new century disclose?”

Hilbert's problem on Diophantine equations

- Determination of solvability of a ***Diophantine equation***
- Diophantine equations are polynomials (involve only addition and multiplication) with integer coefficients.

- Example: $9(u^2+7v^2)^2-7(r^2+7s^2)^2=2$
- Solution: u=1, v=0, r=1, s=0
- Solution:

  u=525692038369576

  v=1556327039191013

  r=2484616164142152

  s=1381783865776981

Hilbert’s Tenth Problem

- Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients,

  devise a ***process*** *according to which it can be determined by a* ***finite*** *number of operations whether the equation is solvable in rational* ***integers****.*

The possibility of a negative solution

- Hilbert: “Every definite mathematical problem must necessarily be susceptible of an exact settlement, either in the form of an actual answer to the question asked, or by the ***proof of the impossibility*** of its solution and therewith the necessary failure of all attempts.”

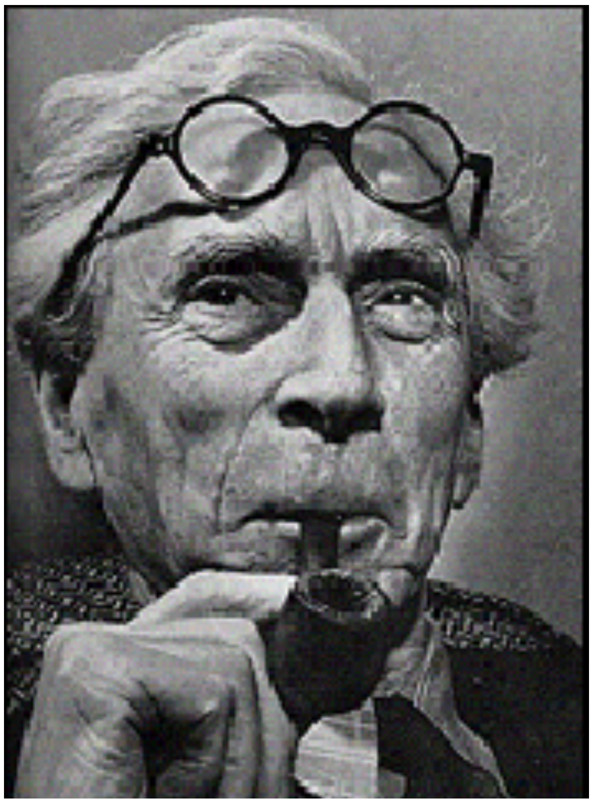
- Emil Post: “Hilbert’s Tenth Problem begs for an ***unsolvability*** proof.”

Hilbert's Program

Proposal for the foundation of mathematics

- Formalize a theory by giving a set of axioms (formal system). The axioms should be consistent.
- Prove the consistency of more complicated mathematical systems in terms of simpler systems.
- Hilbert's Second Problem: Establish the ***consistency*** (compatibility) of the axioms for arithmetic (the numbers 0,1,2,3,4,… with addition and multiplication).

Principia Mathematica, 1910–13 Bertrand Russell (1872–1970) and Alfred Whitehead (1861–1947)



Is Peano Arithmetic complete?

International Congress of Mathematicians in Bologna in 1928

- Hilbert asked for a proof that the formal system of arithmetic (known as ***Peano Arithmetic***) is complete.
- ***Complete***: Every well-formed sentence is either provable or disprovable.
- Peano Arithmetic has an ***algorithmic*** set of axioms.

The Incompleteness Theorem

Kurt Gödel (1906–1978)

- “On formally undecidable propositions of *Principia Mathematica* and related systems I,” 1931.
- ***Undecidable propositions***: neither provable nor disprovable

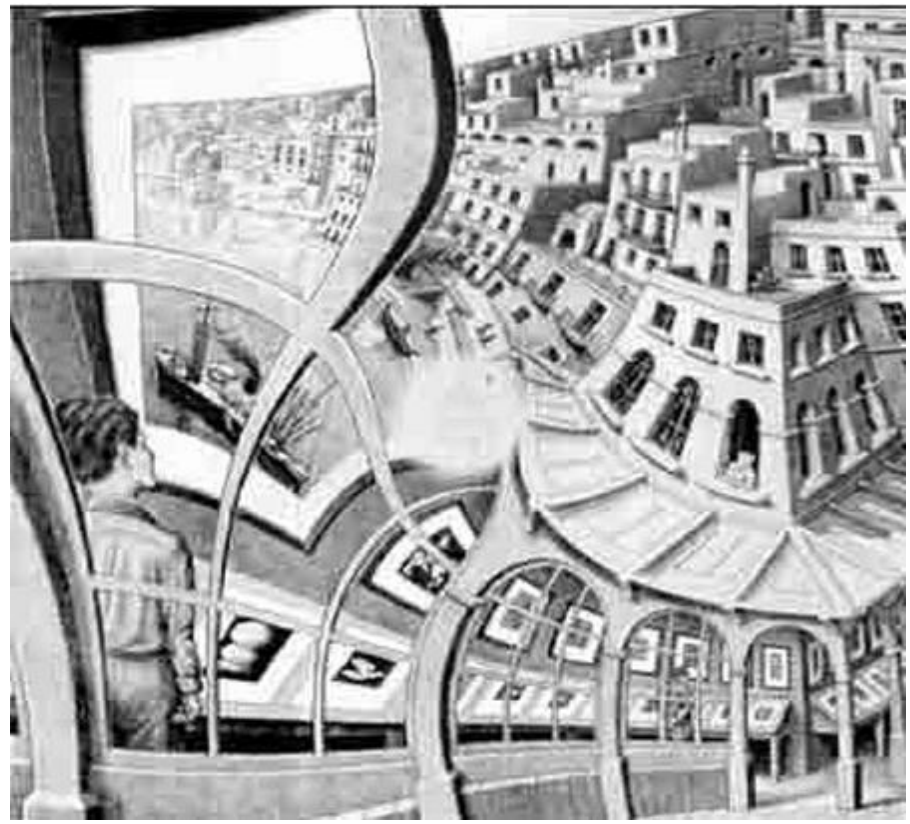
Incompleteness

- In any ***consistent*** formalization of mathematics with an ***algorithmic*** set of axioms, which is sufficiently strong to contain ***arithmetic*****,** one can obtain a ***true*** statement that ***cannot be proved*** within that system.

- Given an algorithm that produces true statements about numbers, one after another, we can always obtain another true statement that is not generated by that algorithm.

Gödel's sentences

- True sentences that are not provable.
- "I am not provable."
- "I am lying."
- Statements about numbers can be ***self-referential***.



Gödel’s arithmetization

- Assigning unique numbers (Gödel numbers) to:
  - symbols
  - formulae (strings of symbols)
  - proofs (strings of formulae)

- A consistent formal mathematical system cannot prove its own ***consistency***.

Mathematical formalism of an algorithm: Turing machine, 1936

- Alan Mathison Turing (1912–1954)
- Several other formalisms followed
- Gödel: “That this really is the correct definition of mechanical computability was established beyond any doubt by Turing.”

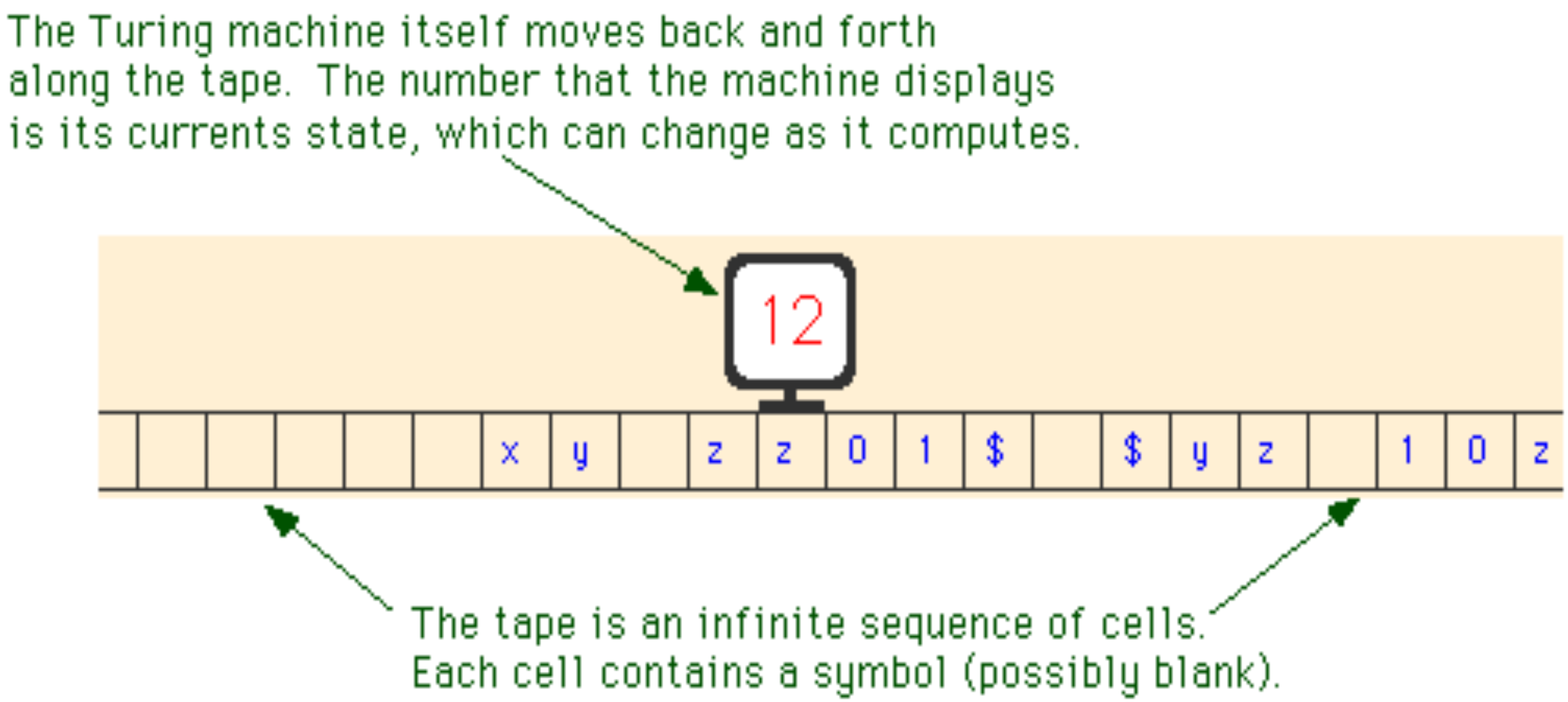
Turing's 1936 paper

"On computable numbers, with an application to the Entscheidungsproblem"

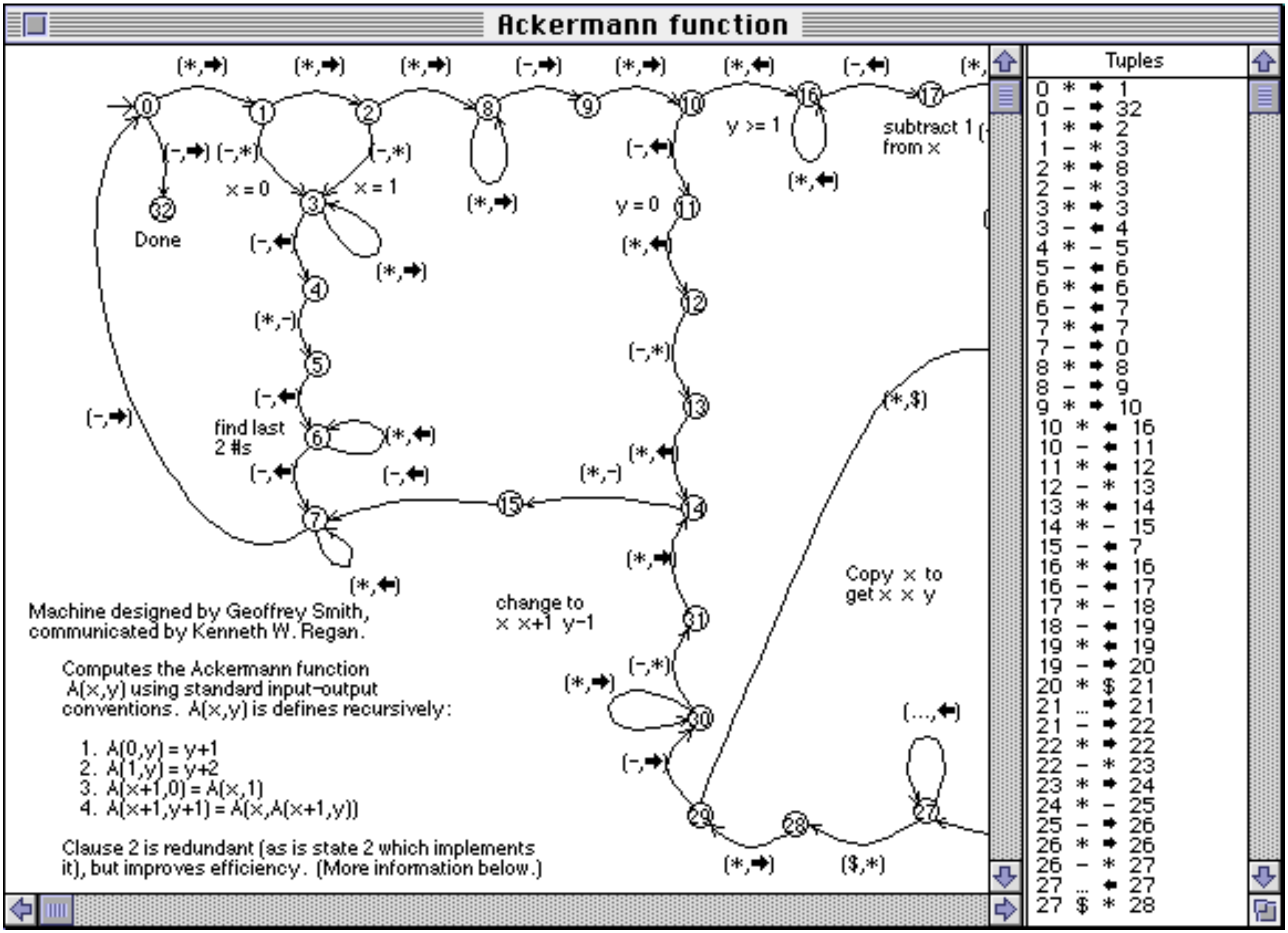
- Introduces an ***abstract machine***, which moves from one state to another, using a precise finite set of rules (given by a finite table), and depends on single symbols it reads from a tape.
- Negative answer to the Entscheidungsproblem

Turing Machine

"The behaviour of the computer at any moment is determined by the ***symbols*** which he is observing and its ***state of mind*** at that moment."



Finite state diagram



Recursion

- Addition

  $a(x,y)=x+y$

- $x+0=x$
- $x+(y+1)=(x+y)+1$

- $a(x,0)=x$
- $a(x,y+1)=a(x,y)+1$

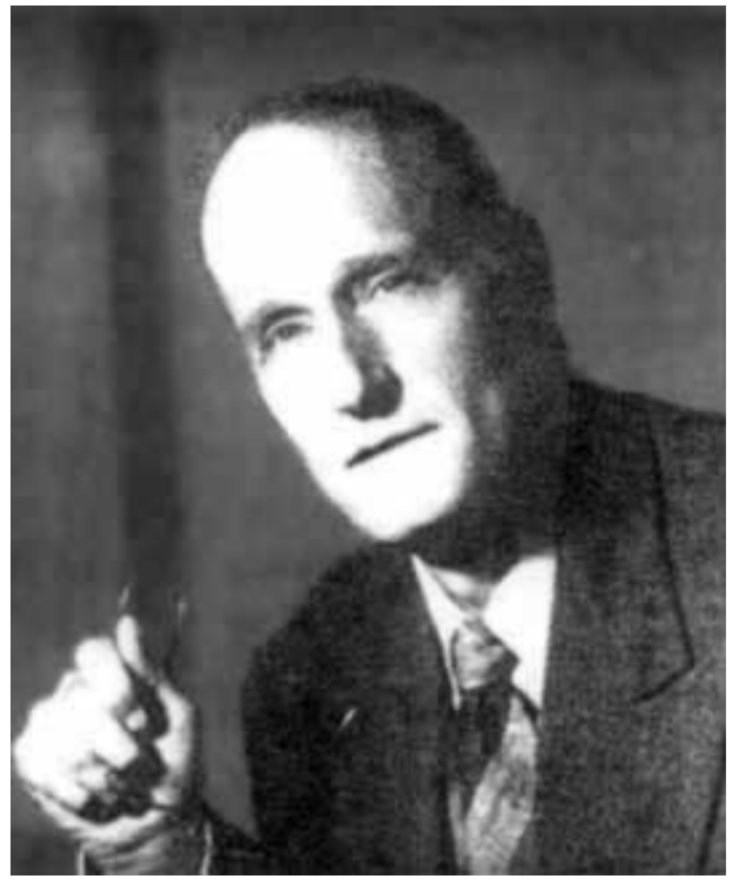
- Multiplication

  $m(x,y)=x \cdot y$

- $x \cdot 0=0$
- $x \cdot (y+1)=x \cdot y+x$

- $m(x,0)=0$
- $m(x,y+1)= a(m(x,y),x)$

Recursive functions
Gödel, 1934; Stephen Kleene, 1938 (1909–1994)



Basic functions

- zero
- adding 1
- projections

  input (x, y) → output x

Operations on functions

- composition
- recursion
- search for the least element that annihilates a function

The Church-Turing Thesis
Alonzo Church (1903–1995)

- Different formalisms (Turing, Gödel, Post, Church, Kleene, Markov) of an algorithm turned out to be ***equivalent***.
- The intuitive notion of an effectively computable function is identified with that of a Turing computable function, since other plausible definitions of effective computability yield notions that are equivalent to Turing computability.

Gödel, 1946:

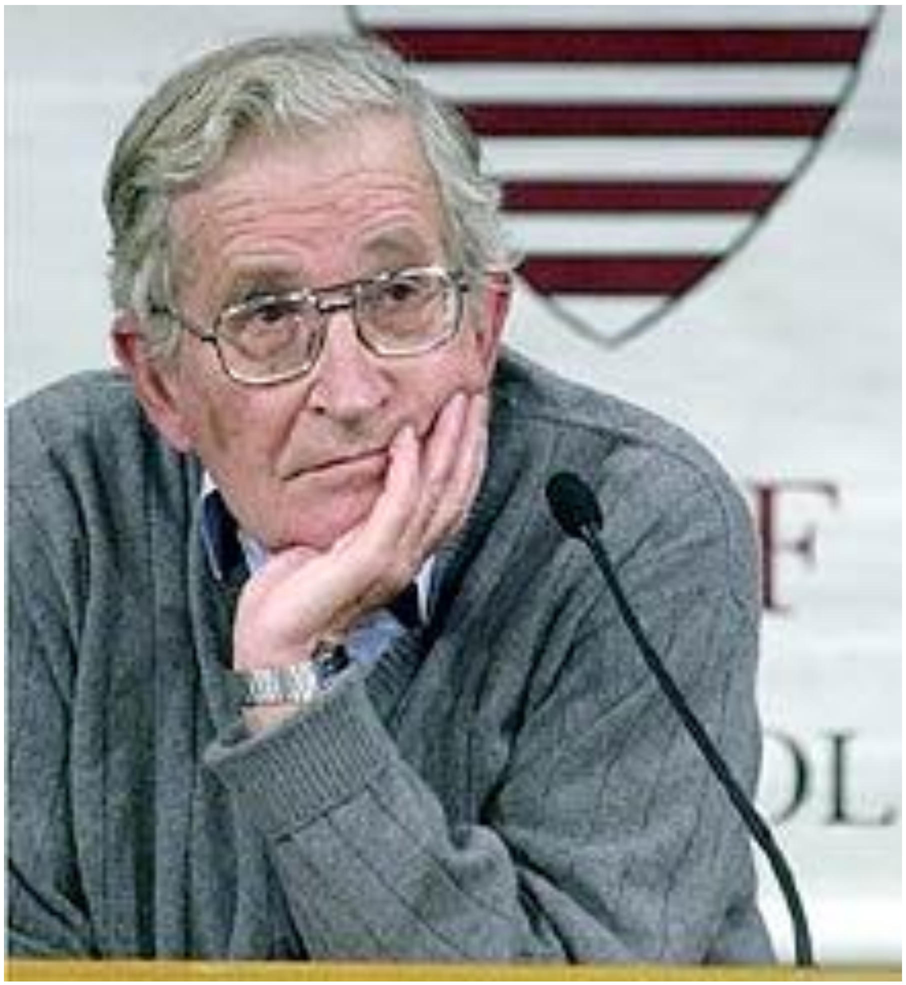
- “It seems to me that great importance of general recursiveness, or Turing computability, is largely due to the fact that with this concept one has for the first time succeeded in giving an ***absolute*** definition of an interesting epistemological notion, i.e., one not depending on the formalism chosen.”

Algorithmically enumerable sets

Church, 1936: Sets the elements of which are enumerated, listed, generated by algorithms

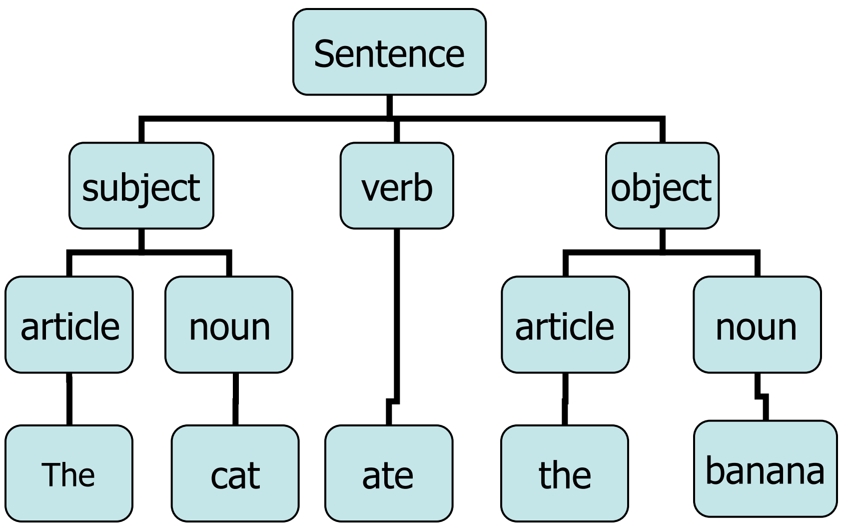
- The outputs of an algorithmic function
- Provable statements in formal systems with algorithmic sets of axioms
- The inputs on which a Turing machine halts

Noam Chomsky's languages (1956)



- Chomsky's languages with unrestricted grammars exactly correspond to algorithmically enumerable ones.

Chomsky’s grammar



- Finite alphabet
- Variable symbols (S=sentence, verb, noun, …)
- Initial symbol=S
- Terminal symbols (cat, ate,…)

- Productions (transformational rules)

  S → (subject)(verb)(object)

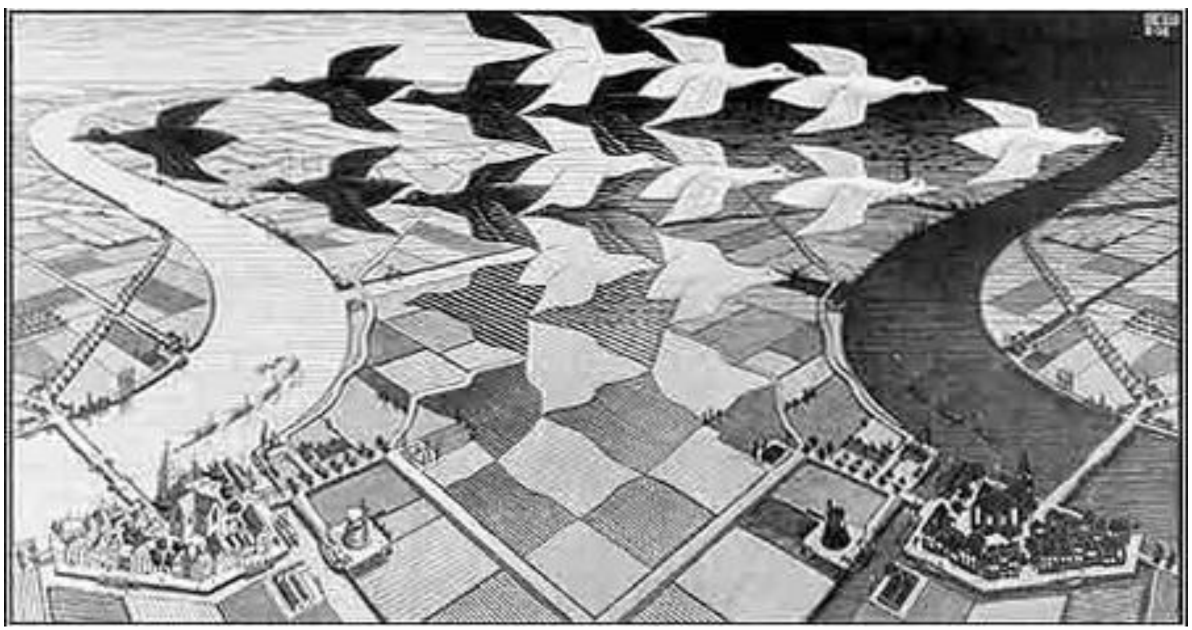
  (subject) → (article)(noun)

  (noun) → (cat)

  …

Not all algorithmically enumerable sets are algorithmic (computable)

- Algorithmically enumerable sets that are not computable are exactly those the complements of which are not algorithmically enumerable.



Algorithmic proof of the Incompleteness Theorem

- The set of all provable sentences in an algorithmic formal system of arithmetic ***is*** an algorithmically enumerable set.

- The set of all true arithmetical sentences ***is not*** algorithmically enumerable.

- Thus, TRUE does not equal PROVABLE.
  (TRUE strictly contains PROVABLE.)

Truth is non-algorithmically enumerable in a strong way

There is an ***algorithm***, which
for any adequate algorithmic formal system
(that is, its code),
outputs
a true sentence that is not provable
(a Gödel sentence).

Undecidability of Hilbert’s Tenth Problem,1970

- Established by Yuri Matiyasevich;

  Martin Davis, Hilary Putnam, Julia Robinson

- Being a Diophantine set is equivalent to being an algorithmically enumerable set.

Davis, Robinson, Putnam, Matiyasevich

Algorithms with external information

- Turing machines augmented with the so-called ***oracles*** introduced by Turing in his 1938 dissertation at Princeton under A. Church.
- An oracle supplies ***additional information*** on demand (performs finitely many non-mechanical steps).
- Allow fine classification of mathematical objects and problems

Turing degrees of complexity

Introduced in 1948 by
Emil Post (1897–1954)

•*degA*<*degB*

if *A* can be computed by a Turing machine with oracle *B*,

but not *vice versa*

(otherwise, *A* and *B* have the same Turing degree)

Rich structure of Turing degrees

- Uncountably many Turing degrees
- Partially ordered
- Algorithmic sets have Turing degree **0**
- Countably infinitely many Turing degrees of algorithmically generated sets
- Uncountably many Turing degrees of complete extensions of Peano Arithmetic