Theory of Computation
Introduction

CSCI 341 - Bucknell University

Fall 2016

David Hilbert (1862-1943)

- In 1900, Hilbert challenged the community of mathematicians with **23 problems**.
- Some problems specifically related to the so called **Hilbert's program**.

Hilbert's program

Secure the foundations of mathematics.

Hilbert's program

Secure the foundations of mathematics.

- Formal language for mathematics (formal logic)

Hilbert's program

Secure the foundations of mathematics.

- Formal language for mathematics (formal logic)
- Completeness (all true statements should be provable)

Hilbert's program

Secure the foundations of mathematics.

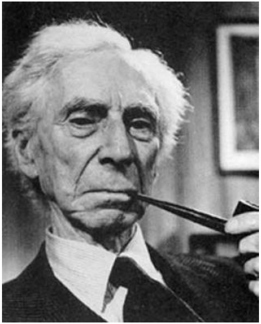
- Formal language for mathematics (formal logic)
- Completeness (all true statements should be provable)
- Consistency (no contradiction should be found)

Hilbert's program

Secure the foundations of mathematics.

- Formal language for mathematics (formal logic)
- Completeness (all true statements should be provable)
- Consistency (no contradiction should be found)
- Decidability (automatically prove/disprove statements)

Bertrand Russell (1872-1970)



Provide the first **formal logic** for mathematics in **Principia Mathematica** (joint work with Whitehead in 1910).

Principia Mathematical 1910

$\vdash . *90 \cdot 16 \cdot 152 . \supset \vdash . R_* | R \subset R_* . R_* | R \subset R_*$ (2)

$\vdash . *90 \cdot 151 . \quad \supset \vdash . R_* | \breve{R} \subset R_* | \breve{R}_* . \breve{R}_* | R \subset \breve{R}_* | R_*$ (3)

$\vdash . (3) . *92 \cdot 312 . \supset \vdash : \mathrm{Hp} . \supset . R_* | \breve{R} \subset R_* \cup \breve{R}_* . \breve{R}_* | R \subset R_* \cup \breve{R}_*$ (4)

$\vdash . (1) . (2) . (4) . \supset \vdash . \mathrm{Prop}$

***92·33.** $\vdash : R \epsilon 1 \to 1 . \supset . (R \cup \breve{R})_* = R_* \cup \breve{R}_*$

Dem.

$\vdash . *90 \cdot 18 . \quad \supset \vdash . R_* \subset (R \cup \breve{R})_* . \breve{R}_* \subset (R \cup \breve{R})_* .$

$[*23 \cdot 59] \quad \supset \vdash . R_* \cup \breve{R}_* \subset (R \cup \breve{R})_*$ (1)

$\vdash . *33 \cdot 272 . \quad \supset \vdash . I \upharpoonright C\text{‘}(R \cup \breve{R}) = I \upharpoonright C\text{‘}R .$

$[*90 \cdot 15 . *23 \cdot 58] \quad \supset \vdash . I \upharpoonright C\text{‘}(R \cup \breve{R}) \subset R_* \cup \breve{R}_*$ (2)

$\vdash . *92 \cdot 32 . *34 \cdot 34 . \supset \vdash :. \mathrm{Hp} . \supset : S \subset R_* \cup \breve{R}_* . \supset . S | (R \cup \breve{R}) \subset R_* \cup \breve{R}_*$ (3)

$\vdash . (2) . (3) . *91 \cdot 17 \dfrac{R \cup \breve{R}, \; S \subset R_* \cup \breve{R}_*}{R, \qquad \phi S} . \supset$

$\vdash :. \mathrm{Hp} . \supset : P \epsilon \mathrm{Potid}\text{‘}(R \cup \breve{R}) . \supset_P . P \subset R_* \cup \breve{R}_* :$

$[*41 \cdot 151] \quad \supset : \dot{s}\text{‘}\mathrm{Potid}\text{‘}(R \cup \breve{R}) \subset R_* \cup \breve{R}_* :$

$[*91 \cdot 55] \quad \supset : (R \cup \breve{R})_* \subset R_* \cup \breve{R}_*$ (4)

$\vdash . (1) . (4) . \supset \vdash . \mathrm{Prop}$

***92·34.** $\vdash : R \epsilon 1 \to 1 . \supset . (R \cup \breve{R})_{\mathrm{po}} = R_* \cup \breve{R}_*$

Dem.

$\vdash . *92 \cdot 33 . *91 \cdot 52 . \supset$

$\vdash : \mathrm{Hp} . \supset .$

Kurt Gödel (1906-1978)

- **Gödel's Incompleteness Theorem** (1^{st}):
  In an effective logic system, there exist statements that cannot be proved nor disproved. There is a true statement that cannot be proved.

Kurt Gödel (1906-1978)

- **Gödel's Incompleteness Theorem** (1^{st}):
  In an effective logic system, there exist statements that cannot be proved nor disproved. There is a true statement that cannot be proved.
- **Gödel's Incompleteness Theorem** (2^{nd}):
  In an effective logic system, the consistency of the system is not provable within the system.

Gödel is a VIP

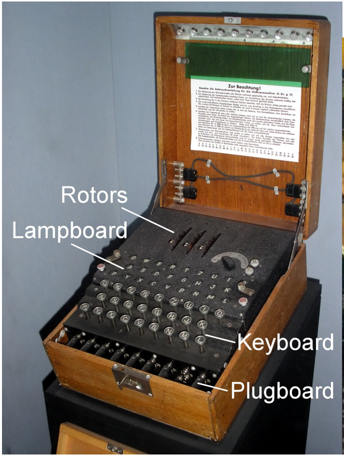
(the pizzeria slide)

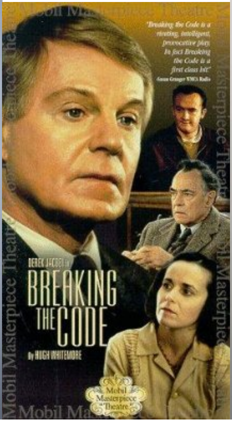
Alan Turing (1912-1954)

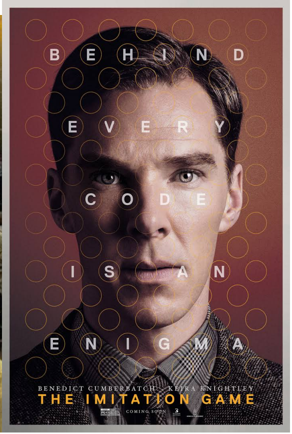
Turing machines, break the enigma code and a pioneer in computer design.

Undecidability: There are problems that cannot be solved automatically.

Breaking Enigma Code

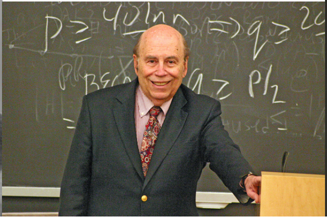
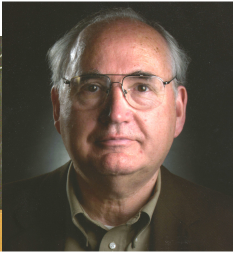






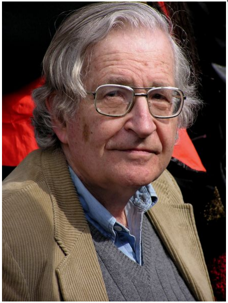
Machine used by Germany for encrypting information during WW2.

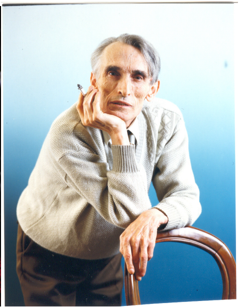
Kleene, Rabin & Scott

In the 1950's, pioneers of the **Theory of Automata** (Regular expressions).

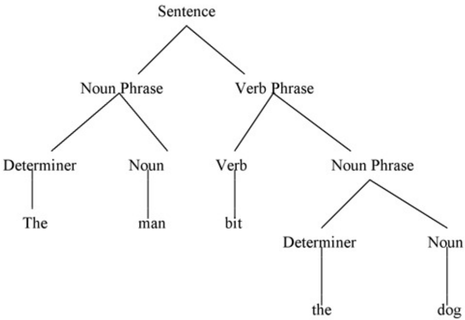
Chomsky & Schutzenberger



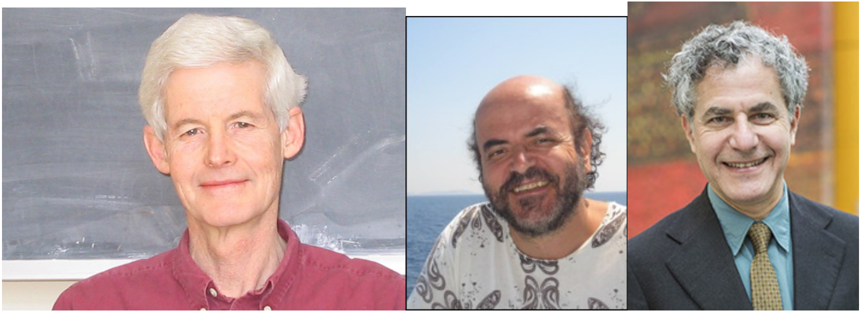


In the 1960's, pioneers in the study of **Formal Languages**.

Understanding language



Cook, Papadimitriou and Sipser



- Major contributors to the **Complexity Theory**, the theory of time-space resources necessay to solve problems.
- Cook: first stated the **P vs NP** problem in the 1970's.

Reward

The *Clay Mathematics Institute* is offering a

US$1 million reward

to anyone who has a formal proof that $P = NP$ or that $P \neq NP$.

Reward

The *Clay Mathematics Institute* is offering a

US$1 million reward

to anyone who has a formal proof that $P = NP$ or that $P \neq NP$.

There are already more than **70 different proofs** listed on
`http://www.win.tue.nl/~gwoegi/P-versus-NP.htm`

Reward

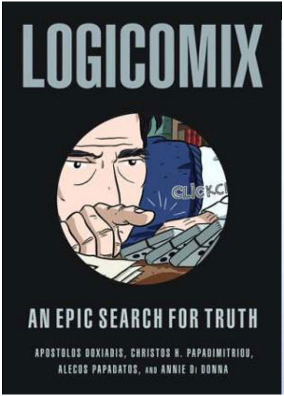
The *Clay Mathematics Institute* is offering a

US$1 million reward

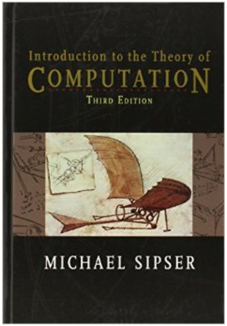
to anyone who has a formal proof that $P = NP$ or that $P \neq NP$.

There are already more than **115 different proofs** listed on
`http://www.win.tue.nl/~gwoegi/P-versus-NP.htm`

The life of Russell in a Comic Book



Introduction to the Theory of Computation
Sipser, 3rd Edition



- Ch 0. Mathematical background
- Ch 1-2. Automata and Languages
- Ch 3-4-5. Computability Theory
- Ch 7-8-9. Complexity Theory

CSCI 341
Theory of Computation

What are you going to learn during this semester?

CSCI 341
Theory of Computation

What are you going to learn during this semester?

- History

CSCI 341
Theory of Computation

What are you going to learn during this semester?

- History
- Foundations

CSCI 341
Theory of Computation

What are you going to learn during this semester?

- History
- Foundations
- Models of computation

CSCI 341
Theory of Computation

What are you going to learn during this semester?

- History
- Foundations
- Models of computation
- Practice formal reasoning

CSCI 341
Theory of Computation

What are you going to learn during this semester?

- History
- Foundations
- Models of computation
- Practice formal reasoning
- Implement the theory

See you tomorrow!