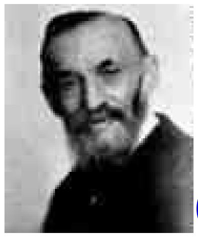
Mathematical Logic.

- From the XIXth century to the 1960s, logic was essentially mathematical.
- Development of first-order logic (1879-1928): Frege, Hilbert, Bernays, Ackermann.
- Development of the fundamental axiom systems for mathematics (1880s-1920s): Cantor, Peano, Zermelo, Fraenkel, Skolem, von Neumann.


Giuseppe Peano (1858-1932)

Mathematical Logic.

- From the XIXth century to the 1960s, logic was essentially mathematical.
- Development of first-order logic (1879-1928): Frege, Hilbert, Bernays, Ackermann.
- Development of the fundamental axiom systems for mathematics (1880s-1920s): Cantor, Peano, Zermelo, Fraenkel, Skolem, von Neumann.
- Traditional four areas of mathematical logic:
  - **Proof Theory**.
  - **Recursion Theory**.
  - **Model Theory**.
  - **Set Theory**.

Gödel (1).

Kurt Gödel (1906-1978)

- Studied at the University of Vienna; PhD supervisor Hans Hahn (1879-1934).
- Thesis (1929): Gödel Completeness Theorem.
- 1931: "*Über formal unentscheidbare Sätze der* Principia Mathematica *und verwandter Systeme I*". Gödel's First Incompleteness Theorem and a proof sketch of the Second Incompleteness Theorem.

Gödel (2).

- 1935-1940: Gödel proves the consistency of the Axiom of Choice and the Generalized Continuum Hypothesis with the axioms of set theory (solving one half of Hilbert's 1st Problem).
- 1940: Emigration to the USA: Princeton.
- Close friendship to Einstein, Morgenstern and von Neumann.
- Suffered from severe hypochondria and paranoia.
- Strong views on the philosophy of mathematics.

Gödel’s Incompleteness Theorem (1).

1928: At the ICM in Bologna, Hilbert claims that the work of Ackermann and von Neumann constitutes a proof of the consistency of arithmetic.

- 1930: Gödel announces his result (G1) in Königsberg in von Neumann’s presence.
- Von Neumann independently derives the Second Incompleteness Theorem (G2) as a corollary.
- Letter by Bernays to Gödel (January 1931): There may be finitary methods not formalizable in $\mathbf{PA}$.
- 1931: Hilbert suggests new rules to avoid Gödel’s result. Finitary versions of the ω-rule.
- By 1934, Hilbert’s programme in the original formulation has been declared dead.

Gödel's Incompleteness Theorem (2).

Theorem (Gödel's Second Incompleteness Theorem). If T is a consistent axiomatizable theory containing $\mathbf{PA}$, then $T \nvdash \mathrm{Cons}(T)$.

- "consistent": $T \nvdash \bot$.
- "axiomatizable": T can be listed by a computer ("computably enumerable", "recursively enumerable").
- "containing $\mathbf{PA}$": $T \vdash \mathbf{PA}$.
- "$\mathrm{Cons}(T)$": The formalized version (in the language of arithmetic) of the statement 'for all T-proofs P, $\bot$ doesn't occur in P'.

Gödel's Incompleteness Theorem (3).

- Thus: Either $\mathbf{PA}$ is inconsistent or the deductive closure of $\mathbf{PA}$ is not a complete theory.
- All three conditions are necessary:
  - **Theorem** (Presburger, 1929). There is a weak system of arithmetic that proves its own consistency ("Presburger arithmetic").
  - If T is inconsistent, then $T \vdash \varphi$ for all φ.
  - If $\mathbb{N}$ is the standard model of the natural numbers, then $\mathrm{Th}(\mathbb{N})$ is a complete extension of $\mathbf{PA}$ (but not axiomatizable).

Gentzen.

Gerhard Gentzen (1909-1945)

- Student of Hermann Weyl (1933).
- 1934: Hilbert's assistant in Göttingen.
- 1934: Introduction of the Sequent Calculus.
- 1936: Proof of the consistency of $\mathbf{PA}$ from a transfinite wellfoundedness principle.

Theorem (Gentzen). Let $T \supseteq \mathbf{PA}$ such that T proves the existence and wellfoundedness of (a code for) the ordinal ε_0. Then $T \vdash \mathrm{Cons}(\mathbf{PA})$.

Arithmetic and orderings (1).

Ordinals are not objects of arithmetic (neither first-order not second-order). So what should it mean that an arithmetical theory proves that "ε_0 is well-ordered"?

Let α be a countable ordinal. By definition, there is some bijection $f : \mathbb{N} \to \alpha$. Define

$$n <_f m :\leftrightarrow f(n) < f(m).$$

Clearly, f is an isomorphism between $\langle \mathbb{N}, <_f \rangle$ and α.

If $g : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ is an arbitrary function, we can interpret it as a binary relation on $\mathbb{N}$:

$$n <_g m :\leftrightarrow g(n, m) = 1.$$

Arithmetic and orderings (2).

Let us work in second-order arithmetic

$$\langle \mathbb{N}, \mathbb{N}^{\mathbb{N}}, 2^{\mathbb{N}\times\mathbb{N}}, +, \times, 0, 1, \mathrm{app} \rangle$$

$g : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ codes a wellfounded relation if and only if

$$\neg \exists F \in \mathbb{N}^{\mathbb{N}} \forall n \in \mathbb{N} (g(F(n+1), F(n)) = 1).$$

“Being a code for an ordinal $< \varepsilon_0$” is definable in the language of second-order arithmetic (ordinal notation systems).
$\mathrm{TI}(\varepsilon_0)$ is defined to be the formalization of “every code g for an ordinal $< \varepsilon_0$ codes a wellfounded relation”.

More proof theory (1).

$\mathrm{TI}(\varepsilon_0)$: "every code g for an ordinal $< \varepsilon_0$ codes a wellfounded relation"

Generalization: If "being a code for an ordinal $< \alpha$" can be defined in second-order arithmetic, then let $\mathrm{TI}(\alpha)$ mean "every code g for an ordinal $< \alpha$ codes a wellfounded relation".

The proof-theoretic ordinal of a theory T.

$$|T| := \sup\{\alpha\,;\, T \vdash \mathrm{TI}(\alpha)\}$$

Rephrasing Gentzen. $|\mathrm{PA}| = \varepsilon_0$.

More proof theory (2).

Results from Proof Theory.

- The proof-theoretic ordinal of primitive recursive arithmetic is ω^ω.
- (Jäger-Simpson) The proof-theoretic ordinal of arithmetic with arithmetical transfinite recursion is Γ_0 (the limit of the Veblen functions).

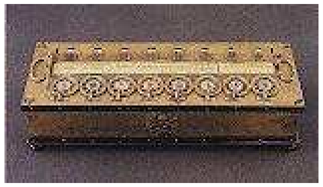
These ordinals are all smaller than ω_1^{CK}, the least noncomputable ordinal.

Early History of Computing.

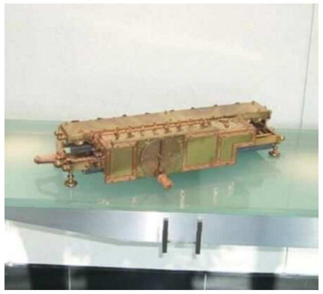
Wilhelm Schickard (1592-1635)

1623.

Blaise Pascal (1623-1662)

1642. 

Gottfried Wilhelm von Leibniz (1646-1716)

1671. 

Computation beyond numbers.

Charles Babbage (1791-1871)

- Difference Engine (1822)
- Analytical Engine

Ada King, Countess of Lovelace (1815-1852)

- Daughter of Lord Byron
- Collaborated with Babbage and de Morgan

“[The Analytical Engine] might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine ... Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientifi c pieces of music of any degree of complexity or extent.”

Turing.

Alan Turing (1912-1954)

- **1936**. *On computable numbers.* The Turing Machine.
- **1938**. PhD in Princeton.
- **1939-1942**. Government Code and Cypher School at Bletchley Park.
- *Enigma*.
- **1946**. Automatic Computing Engine (ACE).
- **1948**. Reader in Manchester.
- **1950**. *Computing machinery and intelligence.* The Turing Test.
- **1952**. Arrested for violation of British homosexuality statutes.

Turing Machines (1).

Entscheidungsproblem.

Is there an algorithm that decides whether a given formula of predicate logic is a tautology or not?

Positive answer simple; negative answer hard. Define “algorithm”.

Turing Machine. An idealized model of computation: an infinite tape, a finite alphabet Σ of symbols that can be on the tape, a read/write head, a finite set of actions A, a finite set S of states and a function (“programme”) $F : \Sigma \times S \to A$. One of the states is designated the HALT state. Write $T := \langle \Sigma, S, A, F \rangle$. There are only countably many Turing machines.

Turing Machines (2).

Turing Machine. An idealized model of computation: an infinite tape, a finite alphabet Σ of symbols that can be on the tape, a read/write head, a finite set of actions A, a finite set S of states and a function ("programme") $F : \Sigma \times S \to A$. One of the states is designated the HALT state. Write $T := \langle \Sigma, S, A, F \rangle$. There are only countably many Turing machines.

- Given some finite string $s \in \Sigma^*$ as input, the machine starts its computation according to F.
- There is a unique defined sequence of states that the computation runs through.
- If one of them is HALT, we say that the machine halts and write $T(s) \downarrow$.
- Otherwise, we say that the machine loops (diverges) and write $T(s) \uparrow$.
- If $T(s) \downarrow$, then the machine outputs the content of the tape. We write $T(s)$ for the output.
- We say that T accepts s if $T(s) \downarrow$ and $T(s) = 1$.
- We say that T rejects s if $T(s) \downarrow$ and $T(s) = 0$.
- A set $X \subseteq \Sigma^*$ is decidable if there is a Turing machine T such that $s \in X$ if and only if T accepts s and $s \notin X$ if and only if T rejects s.

The Universal Turing Machine (1).

Fixing a finite alphabet $\Sigma := \{\sigma_0, ..., \sigma_s\}$ and a finite set of actions $A := \{\alpha_0, ..., \alpha_a\}$, we can list all Turing machines:

If $F : \Sigma \times S \to A$ is a Turing machine programme, we can view it as a partial function $\Phi_F : \{0, ..., s\} \times \{0, ..., n\} \to \{0, ..., a\}$ for some natural number n.

If now $\Phi : \{0, ..., s\} \times \{0, ..., n\} \to \{0, ..., a\}$ is a partial function, we assign a natural number (the “Gödel number of Φ”):

$$\mathrm{G}(\Phi) := \prod_{i \leq s, j \leq n} \mathrm{prime}_{ij}^{\Phi(i,j)+1}.$$

The Universal Turing Machine (2).

$$\mathrm{G}(\Phi) := \prod_{i\leq s, j\leq n} \mathrm{prime}_{ij}^{\Phi(i,j)+1}.$$

Let

$$T \subseteq \mathbb{N} := \{n\,;\,\exists F\,(\mathrm{G}(\Phi_F) = n\,)\,\}$$

be the set of numbers that are Gödel numbers of some Turing machine. Let t_n be the nth number in T and let T_n be the Turing machine such that $\mathrm{G}(\Phi_{T_n}) = t_n$.

> “It can be shown that a single special machine of that type can be made to do the work of all. It could in fact be made to work as a model of any other machine. The special machine may be called the universal machine. (Turing 1947).”

The Universal Turing Machine (3).

Let T be the set of numbers that are Gödel numbers of some Turing machine. Let t_n be the nth number in T and let T_n be the (a) Turing machine such that $\mathrm{G}(\Phi_{T_n}) = t_n$.

A universal Turing machine is a Turing machine U with alphabet $\{0,1\}$ such that at input $\langle n, m\rangle$ such that $n \in T$ the following happens:

- If $T_n(m)\uparrow$, then $U(n,m)\uparrow$.
- If $T_n(m)\downarrow = k$, then $U(n,m)\downarrow = k$.

The Halting Problem K is the set

$$K := \{n\,;\, U(n,n)\downarrow\}.$$

The Halting Problem.

Theorem (Turing). The Halting Problem is not decidable.

Proof. Suppose it is decidable. Then there is a Turing machine T such that

$$\begin{array}{ccccc} T(n)\downarrow = 0 & \leftrightarrow & n \in K & \leftrightarrow & U(n,n)\downarrow \\ T(n)\downarrow = 1 & \leftrightarrow & n \notin K & \leftrightarrow & U(n,n)\uparrow \end{array}$$

By universality, there is some $e \in T$ such that $T = T_e$, *i.e.*,

$$\begin{array}{ccccc} T(n)\downarrow = 0 & \leftrightarrow & T_e(n)\downarrow = 0 & \leftrightarrow & U(e,n)\downarrow = 0 \\ T(n)\downarrow = 1 & \leftrightarrow & T_e(n)\downarrow = 1 & \leftrightarrow & U(e,n)\downarrow = 1 \end{array}$$

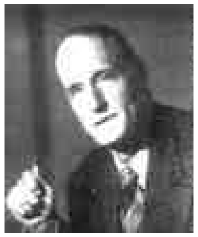
Substitute $n = e$ in the above equivalences and get:

$$U(e,e)\downarrow = 1 \;\leftrightarrow\; U(e,e)\uparrow .$$

Contradiction! q.e.d.

Computability (1).

Alonzo Church
1903-1995


Stephen Kleene
1909-1994

"Both Turing and Gödel preferred the terminology 'computable' for this class of functions. When Turing's 1939 paper appeared, he had already been recruited as a cryptanalyst three days after Britain was plunged into World War II. Gödel moved to set theory. Neither Turing nor Gödel had much influence on the terminology of the subject after 1939.
The present terminology came from Church and Kleene. They had both committed themselves to the new 'recursive' terminology before they had ever heard of Turing or his results. (Soare 1996)"

Robert I. **Soare**, Computability and recursion, **Bulletin of Symbolic Logic** 2 (1996), p.284-321

Computability (2).

computable	recursive
computably enumerable	recursively enumerable
Computability Theory	Recursion Theory

The class of Church-recursive functions is the smallest class containing projections and the successor function closed under primitive recursion, substitution and μ-recursion.

Theorem. A function is Turing-computable if and only if it is Church-recursive.

Church-Turing Thesis. Every algorithm is represented by a Turing machine.

The *Entscheidungsproblem*.

Theorem (Church). The set of all (codes for) tautologies in predicate logic is undecidable, *i.e.*, there is no Turing machine T such that

$$\begin{aligned} T(n)\downarrow= 0 \quad &\leftrightarrow \quad \varphi_n \text{ is a tautology} \\ T(n)\downarrow= 1 \quad &\leftrightarrow \quad \varphi_n \text{ is not a tautology.} \end{aligned}$$

Alonzo **Church**, An Unsolvable Problem of Elementary Number Theory, **American Journal of Mathematics** 58 (1936), p. 345-363

Oracle Machines.

- An oracle machine is a regular Turing machine with an extra tape on which it cannot write but only read.
- If $f : \mathbb{N} \to \mathbb{N}$ and T is an oracle machine, we say that T halts at input x with oracle f if the computation with f written on the extra tape halts. We write $T^f(x)\downarrow$.
- A function f is Turing-computable in g if for all x, we have
$$f(x) = y \leftrightarrow T^g(x) \downarrow y.$$
- **Theorem.** A function is Turing-computable in g if and only if it is in the smallest class containing projections, the successor function, and g closed under primitive recursion, substitution and μ-recursion.
- Let us write C_g for that class.

Relative Computability.

- We write $f \leq_{\mathrm{T}} g$ if and only if $\mathbf{C}_f \subseteq \mathbf{C}_g$.
- $\leq_{\mathrm{T}}$ is a partial preorder, *i.e.*, a transitive and reflexive relation.
- It is not antisymmetric: If f and g are computable, then $\mathbf{C}_f = \mathbf{C}_g$ is the class of computable sets.
- If f is computable, then $f \leq_{\mathrm{T}} K$ and $K \not\leq_{\mathrm{T}} f$.
- Define $f \equiv_{\mathrm{T}} g$ if and only if $f \leq_{\mathrm{T}} g$ and $g \leq_{\mathrm{T}} f$.
- $\mathcal{D} := \mathbb{N}^{\mathbb{N}}/{\equiv_{\mathrm{T}}}$ is a partial order called the Turing degrees.

Two questions.

- **Is $\mathcal{D}$ a linear order?**
  Are there f and g such that $f \not\leq_{\mathrm{T}} g$ and $g \not\leq_{\mathrm{T}} f$?

  No!

- A set A is called computably enumerable (c.e.) if there is a Turing machine T such that

$$x \in A \leftrightarrow T(x) \downarrow .$$

- **Post's Problem**: Is there a non-computable c.e. A such that $\chi_A \not\equiv_{\mathrm{T}} K$.

  Yes! (Friedberg-Muchnik 1956/1957).

Hilbert’s Problems Once Again.

- **Hilbert’s First Problem.** *The Continuum Hypothesis.* “What is the cardinality of the real numbers?”
- **Hilbert’s Second Problem.** *Consistency of Arithmetic.* “Is there a finitistic proof of the consistency of the arithmetical axioms?”
- **Hilbert’s Tenth Problem.** *Solvability of Diophantine Equations.* “Is there an algorithm that determines whether a given Diophantine equation has a solution or not?”

Hilbert's Tenth Problem (1).

A diophantine equation is an equation of the form

$$a_n x^n + a_{n-1} x^{n-1} + ... + a_0 = 0.$$

Hilbert's Tenth Problem. Is there an algorithm that determines given $\langle a_n, ..., a_0 \rangle$ as an input whether the Diophantine equation $a_n x^n + a_{n-1} x^{n-1} + ... + a_0 = 0$ has an integer solution?

Answer (Davis-Putnam-Robinson-Matiyasevich; 1950-1970). **No!**

Hilbert's Tenth Problem (2).

Davis Robinson Matiyasevich Putnam