# CDI

## COMPUTABILITY

## DECIDABILITY

## INCOMPLETENESS

SOMMER-SEMESTER 2023

Lecture I

4 April 2023

1 Virtually identical to "Recursion Theory" WS 21/22

2 Webpage with these notes in PDF.

3 Oral examinations (August / September)

4 Lectures I, II, III : 4 April, 11 April, 18 April → H3
no lecture on 25 April
MAY / JUNE lectures : delivered online
July : fully in person again

5 VERTIEFUNGSVERANSTALTUNG :
assumes MLML Mathe-Logik & Metagerüchte

# LIMITATIVE THEOREMS

Negative theorems showing that a method DOESN'T WORK

Asymmetry between positive theorems & negative theorems:

Example GALOIS THEORY:

It is impossible to construct with ruler & compass the number $\sqrt[3]{2}$.

POSITIVE: Provide a construction.

NEGATIVE: Need to prove that NO construction works.

Galois Theory: Identify the informal concept of "construction w/ ruler & comp" with an algebraic operation & prove the negative result.

# HILBERT'S PROBLEMS

A positive answer to H10 does not require to define "procedure".

A negative answer needs a definition of "VERFAHREN".

INTERPRETED as "algorithm".

David HILBERT
1862-1943

H10

D. Hilbert, Mathematische Probleme, Kgl. Ges. Wiss. Göttingen 1900

10. Entscheidung der Lösbarkeit einer Diophantischen Gleichung.

Eine Diophantische Gleichung mit irgend welchen Unbekannten und mit ganzen rationalen Zahlencoefficienten sei vorgelegt: *man soll ein Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden läßt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.*

# Das ENTSCHEIDUNGSPROBLEM


DIE GRUNDLEHREN DER MATHEMATISCHEN WISSENSCHAFTEN IN EINZELDARSTELLUNGEN

BAND XXVII

D. HILBERT UND W. ACKERMANN

GRUNDZÜGE DER THEORETISCHEN LOGIK

ZWEITE AUFLAGE

SPRINGER-VERLAG BERLIN HEIDELBERG GMBH

Hilbert - Ackermann

1928

**§ 12. Das Entscheidungsproblem.**

Aus den Überlegungen des vorigen Paragraphen ergibt sich die grundsätzliche Wichtigkeit des Problems, bei einer vorgelegten Formel des Prädikatenkalküls zu erkennen, ob es sich um eine identische Formel handelt oder nicht. Nach der in § 5 gegebenen Definition bedeutet die Identität einer Formel dasselbe wie die Allgemeingültigkeit der Formel für jeden Individuenbereich. Man pflegt deswegen auch von dem *Problem der Allgemeingültigkeit* einer Formel zu sprechen. Genauer müßte man statt Allgemeingültigkeit Allgemeingültigkeit für jeden Individuenbereich sagen. Die identischen Formeln des Prädikatenkalküls sind nach den Ausführungen des § 10 gerade die Formeln, die aus dem Axiomensystem des § 5 sich ableiten lassen. Zu einer Lösung des Problems der Allgemeingültigkeit vermag uns diese Tatsache nicht zu helfen, da wir kein allgemeines Kriterium für die Ableitbarkeit einer Formel haben.

Q: Given a formula $\varphi$ in predicate logic, is there an algorithm that determines whether $\varphi$ is valid?

CONTRAST For propositional logic, the method of truth tables provides an algorithm that checks validity.

# COMPUTABILITY

Informal notion: COMPUTATION.

Limitative theorems: Limits of computation.

In this lecture:

Define mathematically precisely the notion of computation & computability.

Then show:

There is a set that is not computable.


Alan Turing 1912–1954



# ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

*By* A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

Defined notion of computation based on known computation, defined computability and proved that the set

$K := \{n\}$ $n$ is a code of a machine that halts in finitely many steps on input $n$

THE HALTING PROBLEM

is not computable.

# DECIDABILITY

Hilbert – Ackermann

**§ 12. Das Entscheidungsproblem.**

Aus den Überlegungen des vorigen Paragraphen ergibt sich die grundsätzliche Wichtigkeit des Problems, bei einer vorgelegten Formel des Prädikatenkalküls zu erkennen, ob es sich um eine identische Formel handelt oder nicht. Nach der in § 5 gegebenen Definition bedeutet die Identität einer Formel dasselbe wie die Allgemeingültigkeit der Formel für jeden Individuenbereich. Man pflegt deswegen auch von dem *Problem der Allgemeingültigkeit* einer Formel zu sprechen. Genauer müßte man statt Allgemeingültigkeit Allgemeingültigkeit für jeden Individuenbereich sagen. Die identischen Formeln des Prädikatenkalküls sind nach den Ausführungen des § 10 gerade die Formeln, die aus dem Axiomensystem des § 5 sich ableiten lassen. Zu einer Lösung des Problems der Allgemeingültigkeit vermag uns diese Tatsache nicht zu helfen, da wir kein allgemeines Kriterium für die Ableitbarkeit einer Formel haben.

Decision problem is a set D of objects & a decision algorithm is a procedure that gets any object x and decides in finitely many steps whether

$x \in D$.

Informal notion: "decision algorithm / procedure"

Limitative theorem: Show that some decision problem cannot be solved by an algorithm.

TEOREMA:

The incomputability of Turing's K gives us as an application the undecidability of the Entscheidungsproblem.

# INCOMPLETENESS


Goal: Axiomatic framework that formalizes ALL of mathematics and provides a proof for every true statement.

# Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I[1]).

Von **Kurt Gödel** in Wien.

## 1.

Die Entwicklung der Mathematik in der Richtung zu größerer Exaktheit hat bekanntlich dazu geführt, daß weite Gebiete von ihr formalisiert wurden, in der Art, daß das Beweisen nach einigen wenigen mechanischen Regeln vollzogen werden kann. Die umfassendsten derzeit aufgestellten formalen Systeme sind das System der Principia Mathematica (PM)[2]) einerseits, das Zermelo-Fraenkelsche (von J. v. Neumann weiter ausgebildete) Axiomensystem der Mengenlehre[3]) andererseits. Diese beiden Systeme sind so weit, daß alle heute in der Mathematik angewendeten Beweismethoden in ihnen formalisiert, d. h. auf einige wenige Axiome und Schlußregeln zurückgeführt sind. Es liegt daher die Vermutung nahe, daß diese Axiome und Schlußregeln dazu ausreichen, alle mathematischen Fragen, die sich in den betreffenden Systemen überhaupt formal ausdrücken lassen, auch zu entscheiden. Im folgenden wird gezeigt, daß dies nicht der Fall ist, sondern daß es in den beiden angeführten Systemen sogar relativ einfache Probleme aus der Theorie der gewöhnlichen ganzen Zahlen gibt[4]), die sich aus den Axiomen nicht

Kurt GÖDEL
1906-1978

Informal notion : foundations for mathematics
proof system

Goal: Prove that for every $\varphi$,
either $\vdash \varphi$ or $\vdash \neg\varphi$.

COMPLETENESS

Theorem (Gödel). If $\Lambda$ is an axiom system for the theory of $(\mathbb{N}, +, \cdot, <)$ that meets basic requirements, then $\Lambda$ is not complete.

# COMPUTABILITY



ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

*By* A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

Alan TURING
1912-1954

Machine model

TURING machine

was based on human
(pencil-&-paper) computations
algorithms.

Turing machines have a single tape, a
read/write-head moving on tape,
reading what is under the head,
reacting to this based on the read
symbol and moving left or right &
re-writing symbols.

# REGISTER MACHINES

In contrast to Turing machines, RM have finitely many registers - last serve as data storage units.

It has no moving R/W-head but accesses the registers directly by address.

Marvin MINSKY
1927-2016

Joachim LAMBEK
1922-2014

Hao WANG
1921-1995

Howard STORGIS
1936-1990

John SHEPHERDSON
1926-2015

# Terminology -

Our basic object is supposed to be a string of symbols.

Fix nonempty, finite set $\Sigma$, called the alphabet. It's elements are called letters.

We write $\Sigma^*$ for the set of all finite seq. of elts of $\Sigma$. These are called WORDS. We write $W$ for these.

Note that $W$ is countably infinite.

By Cantor's theorem, $\wp(W) = \{A ; A \subseteq W\}$ is uncountable.

Remark If "computable" is defined in some way s.t. only countably many things can be computable, we immediately get that those are noncomputable sets.

Some notation for elts of $W$:

$\varepsilon$ for the empty word

$w, v \in W \rightsquigarrow$ write $wv$ for the concatenation of $w$ & $v$.

$|wv| = |w| + |v|$ $\quad wv(k) := \begin{cases} w(k) & \text{if } k < |w| \\ v(\ell) & \text{if } k = |w| + \ell \end{cases}$

$w \in W \Longrightarrow$ there is some $n \in \mathbb{N}$ s.t. $w : n \longrightarrow \Sigma$. Call this the length of $w$, $|w|$.

If $\sigma \in \Sigma$, $w \in W$, write $\sigma w$ for the obvious.

If $\sigma \in \Sigma$, we write $\sigma^n$ for the unique word of length $n$ consisting just of $\sigma$.

If $w \in W$, $w^n$ is defined by recursion:

$w^0 := \varepsilon$

$w^{n+1} := w^n w$.

## 4.1 Register machines

Let $\Sigma$ be an alphabet and $Q$ a non-empty finite set whose elements we shall call *states*. A tuple of the form

$$(0, k, a, q) \in \mathbb{N} \times \mathbb{N} \times \Sigma \times Q,$$
$$(1, k, a, q, q') \in \mathbb{N} \times \mathbb{N} \times \Sigma \times Q \times Q,$$
$$(2, k, q, q') \in \mathbb{N} \times \mathbb{N} \times Q \times Q \text{ or}$$
$$(3, k, q, q') \in \mathbb{N} \times \mathbb{N} \times Q \times Q$$

is called a $(\Sigma, Q)$-*instruction*. For improved readability, we write

$$+(k, a, q) := (0, k, a, q), \qquad (\text{"add"})$$
$$?(k, a, q, q') := (1, k, a, q, q'), \qquad (\text{"check"})$$
$$?(k, \varepsilon, q, q') := (2, k, q, q') \text{ and} \qquad (\text{"check"})$$
$$-(k, q, q') := (3, k, q, q') \qquad (\text{"remove"})$$

| Instruction | Interpretation |
|---|---|
| $+(k, a, q)$ | "Add the letter $a$ to the content of register $k$ and go to state $q$." |
| $?(k, a, q, q')$ | "Check whether the last letter in register $k$ is $a$; if so, go to state $q$; otherwise, go to state $q'$." |
| $?(k, \varepsilon, q, q')$ | "Check whether register $k$ is empty; if so, go to state $q$; otherwise, go to state $q'$." |
| $-(k, q, q')$ | "Check whether register $k$ is empty; if so, go to state $q$; otherwise, remove the final letter of its content and go to state $q''$". |

**Definition** A $\Sigma$-register machine is a triple $(\Sigma, Q, P)$ where $\Sigma$ is an alphabet, $Q$ is a nonempty finite set of states, $P$ is a function with domain $Q$ s.t. for all $q \in Q$, $P(q)$ is an instruction.

Also assume that $q_H \neq q_S \in Q$.
HALT STATE  START STATE

**Note** If $(\Sigma, Q, \tau)$ is a RM, then $\mathrm{ran}(\tau)$ is finite, so there is a maximal $k$ s.t. $\tau(q)$ refers to state $k$. We call this the UPPER REGISTER INDEX of $(\Sigma, Q, \tau)$.