Algorithmen und Turingmaschinen: Eine historische Perspektive

Informatik-Didaktik-Kolloquium, Jena, März 2012

Wolfgang Thomas

RWTH AACHEN

Alan M. Turing (1912-1954)

Plan

1. ”Prähistorisches”: Al-Khwarizmi und Leibniz
2. Der Weg zu Hilberts ”Entscheidungsproblem”
3. Turings Durchbruch von 1936
4. Was ist ein “Algorithmus”?

”Prähistorisches”: Al-Khwarizmi and Leibniz

Bagdad um das Jahr 800

Muhammad Abu-Abdullah Abu-Jafar ibn Musa Al-Khwarizmi Al-Madjusti Al-Qutrubulli

(ca. 780-850)

Al-Khwarizmi: Über die indischen Zahlen

Gottfried Wilhelm Leibniz (1646-1716)

Aus einem Brief an den Herzog Johann Friedrich

“In Philosophia habe ich ein Mittel funden, dasjenige was Cartesius und andere per Algebram et Analysin in Arithmetica et Geometria gethan, in allen scientien zuwege zu bringen per Artem Combinatoriam [. . .]. Dadurch alle Notiones compositae der ganzen welt in wenig simplices als deren Alphabet reduciret, und aus solches alphabets combination wiederumb alle dinge, samt ihren theorematibus, und was nur von ihnen zu inventiren müglich, ordinata methodo, mit der zeit zu finden, ein weg gebahnet wird.”

Arithmetisierung der Logik I (1685-87)

Non inelegans specimen demonstrandi in abstractis

(Eine nicht unelegante Art des abstrakten Schließens)

Theorem XIII.

Si coincidentibus addendo alia fiant coincidentia, addita sunt inter se communicantia.

Wenn von Gleichen durch Hinzufügen von Neuem Gleiche entstehen, dann gibt es unter den Hinzufügungen Überschneidungen.

Wenn $A + B = A + N$ **und** $A \neq A + B$**, dann** $B \cap N \neq \emptyset$

Diese Untersuchung bereitet die Boolesche Algebra vor, unter Verwendung arithmetischer Notation.

Arithmetisierung der Logik II (1679)

Elementa calculi

(Elemente eines Kalküls)

Verbi gratia quia Homo est Animal rationale (et quia Aurum est metallum ponderosissimum) hinc si sit Animalis (metalii) numerus a ut 2 (m ut 3) Rationalis (ponderosissimi) vero numerus r ut 3 (p ut 5) erit numerus hominis seu h idem quot ar id est in hoc exemplo $2, 3$ seu 6 (et numerus auri solis s idem quot mp id est in hoc exemplo $3, 5$ seu 15.

Dies nimmt die Idee der ”Gödelisierung” (Gödel 1931) vorweg: die Kodierung zusammengesetzter Begriffe durch Zahlen, unter Benutzung der Primfaktorzerlegung.

Optimismus

aus dem Manuskript “Grundlagen einer universalen Charakteristik”:

Denn wenn einmal von den Missionaren diese Sprache eingeführt sein wird, dann wird auch die wahre Religion, die mit der Vernunft am besten vereinbar ist, auf festem Grund stehen, und einen Abfall von ihr wird man in Zukunft ebenso wenig zu befürchten haben, wie man eine Abkehr der Menschen von der Arithmetik und der Geometrie, die sie einmal gelernt haben, befürchtet.

Ich glaube, dass einige Auserlesene die Sache in fünf Jahren leisten können, dass sie jedoch schon nach zwei Jahren so weit sein werden, die Lehren, die im Leben am meisten gebraucht werden, d.h. die der Moral und Metaphysik, nach einem unfehlbaren Rechenverfahren zu beherrschen.

Der Weg zu Hilberts Entscheidungsproblem

Gottlob Frege (1848-1925)

BEGRIFFSSCHRIFT,

EINE DER ARITHMETISCHEN NACHGEBILDETE

FORMELSPRACHE

DES REINEN DENKENS.

VON

DR. GOTTLOB FREGE,

PRIVATDOCENTEN DER MATHEMATIK AN DER UNIVERSITÄT JENA.

HALLE A/S.

VERLAG VON LOUIS NEBERT.

1879.

(358):
(125):
(302)::
(188)::
(413
(α
(β
(γ
(δ
(μ
(ν
(ξ

B. Russell

A.N. Whitehead

$*54{\cdot}443.\ \vdash :: x \neq y : \phi(x,y) . \equiv . \phi(y,x) : \supset :.$
$z, w \in \iota`x \cup \iota`y . z \neq w . \supset_{z,w} . \phi(z,w) : \equiv . \phi(x,y) \quad [*54{\cdot}442]$

$*54{\cdot}45.\ \vdash :. (\exists z, w) . z, w \in \iota`x \cup \iota`y . \phi(z,w) .$
$\equiv : \phi(x,x) . \vee . \phi(x,y) . \vee . \phi(y,x) . \vee . \phi(y,y) \quad [*51{\cdot}235]$

$*54{\cdot}451.\ \vdash :: \sim\phi(x,x) . \sim\phi(y,y) . \supset :. (\exists z, w) . z, w \in \iota`x \cup \iota`y . \phi(z,w) .$
$\equiv : \phi(x,y) . \vee . \phi(y,x) \quad [*54{\cdot}45]$

$*54{\cdot}452.\ \vdash :: \sim\phi(x,x) . \sim\phi(y,y) : \phi(x,y) . \equiv . \phi(y,x) : \supset :$
$(\exists z, w) . z, w \in \iota`x \cup \iota`y . \phi(z,w) . \equiv . \phi(x,y) \quad [*54{\cdot}451]$

$*54{\cdot}46.\ \vdash : (\exists z, w) . z, w \in \iota`x \cup \iota`y . z \neq w . \equiv . x \neq y \quad [*54{\cdot}452 . *13{\cdot}15{\cdot}16]$

$*54{\cdot}5.\ \vdash :. \alpha \in 2 . \supset : \alpha \subset \iota`z \cup \iota`w . \equiv . \alpha = \iota`z \cup \iota`w$

Dem.

$\vdash . *54{\cdot}4 . \supset$

$\vdash :. \alpha \subset \iota`z \cup \iota`w . \supset : \alpha = \Lambda . \vee . \alpha = \iota`z . \vee . \alpha = \iota`w . \vee . \alpha = \iota`z \cup \iota`w \qquad (1)$

$\vdash . *54{\cdot}3 . *24{\cdot}54 . \qquad \supset \vdash : \mathrm{Hp} . \supset . \alpha \neq \Lambda \qquad (2)$

$\vdash . *54{\cdot}26 \frac{z, z}{x, y} . *13{\cdot}15 . \qquad \supset \vdash : \mathrm{Hp} . \supset . \alpha \neq \iota`z \qquad (3)$

$\vdash . (3) \frac{w}{z} . \qquad \supset \vdash : \mathrm{Hp} . \supset . \alpha \neq \iota`w \qquad (4)$

$\vdash . (1) . (2) . (3) . (4) . *3{\cdot}53 . \supset \vdash : \mathrm{Hp} . \supset : \alpha \subset \iota`z \cup \iota`w . \supset . \alpha = \iota`z \cup \iota`w \qquad (5)$

David Hilbert (1862-1943)

Hilberts Programm

Kodiere das mathematische Vorgehen in formaler Weise:

- **Entwicklung eines Beweiskalküls**
- **Entwicklung einer Axiomatisierung mathematischer Theorien**
- **Analyse der formalen Beweise zwecks Ausschluss von Widersprüchen**

Fundamentale Ergebnisse für die Logik erster Stufe:

- **Korrektheit und Vollständigkeit des Beweiskalküls (Gödel 1930)**
- **Unvollständigkeit jeder Axiomatisierung der Struktur** $(\mathbb{N}, +, \cdot, 0, 1)$ **(Gödel 1931)**

Kurt Gödel (1906-1978)

Entscheidungsprobleme Hilberts

- **Hilberts Entscheidungsproblem (1928)**
  Das Entscheidungsproblem ist gelöst, wenn man ein Verfahren kennt, das bei einem vorgelegten logischen Ausdruck durch endlich viele Operationen die Entscheidung über die Allgemeingültigkeit bzw. Erfüllbarkeit erlaubt.

- **Das 10. Hilbertsche Problem (1900)**
  Eine diophantische Gleichung mit irgendwelchen Unbekannten und mit ganzen rationalen Zahlenkoeffizienten sei vorgelegt: Man soll ein Verfahren angeben, nach welchem sich mittels einer endlichen Anzahl von Operationen entscheiden lässt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.

Axel Thue (1863-1922)

Thues Problem (1910)

Gegeben zwei Terme s, t und eine Axiomenmenge aus Gleichungen $u(x_1, \ldots, x_n) = v(x_1, \ldots, x_n)$

entscheide, ob man von s durch endlich viele Anwendungen der Axiome schließlich t erhalten kann.

Thues Einschätzung:

Eine Lösung dieser Aufgabe im allgemeinsten Falle dürfte vielleicht mit unüberwindlichen Schwierigkeiten verbunden sein.

DIE LÖSUNG EINES SPEZIALFALLES EINES GENERELLEN LOGISCHEN PROBLEMS

Abteilung A.

§ 1.

Es kann eintreffen, dass man aus einem beliebigen Begriffe A einer Begriffskategorie P und aus einem beliebigen Begriffe B einer Begriffskategorie Q durch ein gewisses Verfahren oder Operation θ eindeutig einen Begriff C einer Begriffskategorie R bilden kann.

Wir können C z. B. durch den Ausdruck

$$[(A)\ \theta\ (B)]$$ bezeichnen.

Turings Durchbruch

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

Reprinted with the kind permission of the London Mathematical Society from the Proceedings of the London Mathematical Society, ser. 2, vol. 42 (1936-7), pp. 230-265; corrections, Ibid, vol 43 (1937) pp. 544-546.

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

Turings Beiträge in dieser Arbeit

1. Ein abstraktes Maschinenmodell (”Turingmaschine”), darauf aufbauend eine Präzisierung von ”berechenbar”,
2. eine sorgfältige Rechtfertigung dafür,
3. Konzeption und Definition einer universellen (programmierbaren) Maschine,
4. Nachweis eines (Turing-) unentscheidbaren Problems,
5. Nachweis der Unlösbarkeit des Hilbertschen Entscheidungsproblems,
6. Grundlegung einer ”berechenbaren Analysis”
7. Äquivalenz zwischen Turingmaschinen and ”λ-Kalkül”

Motivierung der Turingmaschine

Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent†. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of single symbols. Thus an Arabic numeral such as

Kontrolle
"Computer"
Lese- und
Schreibkopf
Rechenband
⊔ ⊔ ⊔ 0 1 ⊔ 0 0 1 ⊔ 1 0 1 1 ⊔

Turingmaschine

- Unendliches, in Felder eingeteiltes ”Rechenband”
- Beschriftung eines Feldes jeweils durch Symbol eines endlichen Alphabets (Leersymbol als Standardeintrag)
- Kontrolleinheit, mit endlich vielen Zuständen
- Zugriff auf Band über Lese- und Schreibkopf, der sich auf einem Feld (dem ”Arbeitsfeld” AF) befindet
- Instruktion für einen Schritt:
  ”in Zustand p mit a auf AF:
  überschreibe a durch b, bewege den Kopf ein Feld nach links bzw. rechts, und gehe in Zustand q”

Zwei Dimensionen

Natürlicherer Zugang: Zweidimensionale Rechenebene

Konkretisierung zum Beispiel mit Kara (Reichert-Nievergelt-Hartmann)

Komplikationen:

- **Vier statt zwei Bewegungsrichtungen**
- **Schwerfälligere Notation einer Konfiguration**
- **Schwerfälligerer Nachweis einer universellen Maschine**

Technisch etwas mühsam, aber elementar ist der formale Äquivalenznachweis zu Turingmaschinen.

Rechtfertigung der Church-Turing-These

1. **Turings Analyse des Rechenvorgangs (”Intuition”)**

2. **Formale Äquivalenz zu anderen Zugängen (”Übergeordnete Signifikanz”)**

3. **Erfolgreiche Behandlung komplexer Beispiele (”Erfahrung”)**

Universalität und Unentscheidbarkeit

(in heutiger Formulierung)

Man ordnet einer Turingmaschine eine ”S.D” zu, eine ”standard description”, die auf das Band geschrieben werden kann.

Zu einer S.D s sei M_s die durch s beschriebene Turingmaschine.

Es gibt eine Turingmaschine U, die bei Vorgabe einer S.D s und einer weiteren Eingabe e so arbeitet wie die Turingmaschine M_s auf e.

Heute sagt man: U ist eine Turingmaschine, die als Interpreter für Turingmaschinen funktioniert.

Unentscheidbarkeit

Folgende Eigenschaft einer S.D s ist nicht Turing-entscheidbar:

M_s angesetzt auf s druckt irgendwann das Symbol 0

und ebenfalls folgende Eigenschaft:

M_s angesetzt auf das leere Band druckt irgendwann das Symbol 0

Zum Hilbertschen Entscheidungsproblem

Turing führt dazu eine Problemreduktion durch:

Zu jeder Turingmaschine M konstruiert er eine Logik-Formel F_M erster Stufe mit:

M angesetzt auf das leere Band druckt irgendwann 0

gdw.

die Formel F_M ist allgemeingültig
(oder, nach Gödels Vollständigkeitssatz: ist im Beweiskalkül ableitbar)

Konfluenz der Ideen 1936

A. Church

S. Kleene

E. Post

Alonzo Church

AN UNSOLVABLE PROBLEM OF ELEMENTARY NUMBER THEORY.[1]

Reprinted from THE AMERICAN JOURNAL OF MATHEMATICS, vol. 58, pp. 345-363 (1936) with the kind permission of The Johns Hopkins Press.

1. Introduction. There is a class of problems of elementary number theory which can be stated in the form that it is required to find an effectively calculable function f of n positive integers, such that $f(x_1, x_2, \cdots, x_n) = 2$ [2] is a necessary and sufficient condition for the truth of a certain proposition of elementary number theory involving $x_1, x_2, \cdots, x_n$ as free variables.

An example of such a problem is the problem to find a means of determining of any given positive integer n whether or not there exist positive integers x, y, z, such that $x^n + y^n = z^n$. For this may be interpreted, required to find an effectively calculable function f, such that $f(n)$ is equal to 2 if and only if there exist positive integers x, y, z, such that $x^n + y^n = z^n$. Clearly the condition that the function f be effectively calculable is an essential part of the problem, since without it the problem becomes trivial.

Die Church'sche These

We now define the notion, already discussed, of an effectively calculable function of positive integers by identifying it with the notion of recursive function of positive integers (or of a λ-definable function of positive integers).

A NOTE ON THE ENTSCHEIDUNGSPROBLEM

ALONZO CHURCH

In a recent paper[1] the author has proposed a definition of the commonly used term "effectively calculable" and has shown on the basis of this definition that the general case of the Entscheidungsproblem is unsolvable in any system of symbolic logic which is adequate to a certain portion of arithmetic and is ω-consistent. The purpose of the present note is to outline an extension of this result to the engere Funktionenkalkül of Hilbert and Ackermann.[2]

In the author's cited paper it is pointed out that there can be associated recursively with every well-formed formula[3] a recursive enumeration of the formulas into which it is convertible.[3] This means the existence of a recursively defined function a of two positive integers such that, if y is the Gödel representation of a well-formed formula $\boldsymbol{Y}$ then $a(x, y)$ is the Gödel representation of the xth formula in the enumeration of the formulas into which $\boldsymbol{Y}$ is convertible.

Stephen C. Kleene

GENERAL RECURSIVE FUNCTIONS OF NATURAL NUMBERS[1]

Reprinted from MATHEMATISCHE ANNALEN Band 112, Heft 5 (1936) pp. 727-742, with the kind permission of Springer-Verlag.

The substitution

$$\varphi(x_1, \ldots, x_n) = \theta(\chi_1(x_1, \ldots, x_n), \ldots, \chi_m(x_1, \ldots, x_n)), \tag{1}$$

and the ordinary recursion with respect to one variable

$$\begin{aligned} \varphi(0, x_2, \ldots, x_n) &= \psi(x_2, \ldots, x_n) \\ \varphi(y+1, x_2, \ldots, x_n) &= \chi(y, \varphi(y, x_2, \ldots, x_n), x_2, \ldots, x_n), \end{aligned} \tag{2}$$

where $\theta, \chi_1, \ldots, \chi_m, \psi, \chi$ are given functions of natural numbers, are examples of the definition of a function φ by equations which provide a step by step process for computing the value $\varphi(k_1, \ldots, k_n)$ for any given set $k_1, \ldots, k_n$ of natural numbers. It is known that there are other definitions of this sort, e. g. certain recursions with respect to two or more variables simultaneously, which cannot be reduced to a succession of substitutions and ordinary recursions[2]). Hence, a characterization of the notion of recursive definition in general, which would include all these cases, is desirable. A definition of general recursive function of natural numbers was suggested by Herbrand to Gödel, and was used by Gödel with an important modification in a series of lectures at Princeton in 1934. In this paper we offer several observations on general recursive functions, using essentially Gödel's form of the definition.

FINITE COMBINATORY PROCESSES—FORMULATION 1

EMIL L. POST

The present formulation should prove significant in the development of symbolic logic along the lines of Gödel's theorem on the incompleteness of symbolic logics[1] and Church's results concerning absolutely unsolvable problems.[2]

We have in mind a *general problem* consisting of a class of *specific problems*. A solution of the general problem will then be one which furnishes an answer to each specific problem.

In the following formulation of such a solution two concepts are involved: that of a *symbol space* in which the work leading from problem to answer is to be carried out,[3] and a fixed unalterable *set of directions* which will both direct operations in the symbol space and determine the order in which those directions are to be applied.

In the present formulation the symbol space is to consist of a two way infinite sequence of spaces or boxes, i.e., ordinally similar to the series of integers $\cdots$, $-3, -2, -1, 0, 1, 2, 3, \cdots$. The problem solver or worker is to move and work in this symbol space, being capable of being in, and operating in but one box at a time. And apart from the presence of the worker, a box is to admit of but two possible conditions, i.e., being empty or unmarked, and having a single mark in it, say a vertical stroke.

One box is to be singled out and called the starting point. We now further assume that a specific problem is to be given in symbolic form by a finite number of boxes being marked with a stroke. Likewise the answer is to be given in symbolic form by such a configuration of marked boxes. To be specific, the answer is to be the configuration of marked boxes left at the conclusion of the solving process.

The worker is assumed to be capable of performing the following primitive acts:[4]

(a) *Marking the box he is in (assumed empty),*
(b) *Erasing the mark in the box he is in (assumed marked),*
(c) *Moving to the box on his right,*
(d) *Moving to the box on his left,*
(e) *Determining whether the box he is in, is or is not marked.*

The set of directions which, be it noted, is the same for all specific problems and thus corresponds to the general problem, is to be of the following form. It is to be headed:

Start at the starting point and follow direction 1.

Received October 7, 1936. The reader should compare an article by A. M. Turing, *On computable numbers*, shortly forthcoming in the ***Proceedings of the London Mathematical Society.*** The present article, however, although bearing a later date, was written entirely independently of Turing's. *Editor.*

[1] Kurt Gödel, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, ***Monatshefte für Mathematik und Physik,*** vol. 38 (1931), pp. 173–198.

[2] Alonzo Church, *An unsolvable problem of elementary number theory*, ***American Journal of Mathematics,*** vol. 58 (1936), pp. 345–363.

[3] Symbol space, and time.

[4] As well as otherwise following the directions described below.

Gödels Bewertung (1946)

Gödel betont

[. . .]

the great importance of the concept of general recursiveness (or Turing's computability). It seems to me that this importance 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.

[. . .]

By a kind of miracle it is not necessary to distinguish orders

[. . .]

Was ist ein “Algorithmus”?

Spielarten von Algorithmen

Turing bezieht sich auf Algorithmen zur Symbolmanipulation, wie sie in der Arithmetik und der formalen Logik benutzt werden.

Andere Typen von Algorithmen

- **Algorithmen der Euklidischen Geometrie (etwa für Dreieckskonstruktionen),**
- **Abstrakte Algorithmen über den reellen Zahlen, etwa das Horner-Schema:**
  $a + x(b + x(c + dx))$ **zur Auswertung von**
  $a + bx + cx^2 + dx^3$
- **Graphalgorithmen, Algorithmen über Bäumen etc.**

Abstrakte Zustandsmaschinen

Vorläufer:

- **A. N. Kolmogorov, V. A. Uspenski, On the definition of an algorithm (1963)**
- **R. Gandy, Church's Thesis and principles for mechanism (1980)**

Ausgestaltung durch "abstract state machines"' ("ASM's"):

- **Y. Gurevich, Sequential abstract state machines capture sequential algorithms (2000)**
- **N. Dershowitz, Y. Gurevich, A natural axiomatization of computability and proof of Church's Thesis (2008)**

Neue Dimensionen

- **Nicht-terminierende reaktive Systeme**
  Beispiel: Controller in industriellen Anlagen
- **Etablierung von Gleichgewichten in verteilten Systemen**
  Beispiel: Auktionsmechanismen im Internet
- **Verfahren der Kognition und Klassifikation**
  Beispiele: Data Mining, Bildverarbeitung, automatische Sprachübersetzung
- **Schichtung von Algorithmen zu hochkomplexen Systemen**
  Beispiele:
  Betriebssysteme, Systeme der Unternehmensverwaltung

Der Sprung von der Turingmaschine zum Betriebssystem entspricht dem Sprung vom Atom zum Lebewesen.

Schlusswort

Turings Analyse des Begriffs ”berechenbar” ist

- **der Abschluss von jahrhundertelangen Bemühungen um die Klärung dessen, was ”symbolisches Rechnen” (in Arithmetik und Logik) vermag,**
- **der Beginn einer neuen mathematischen Wissenschaft und dann auch Ingenieuerdisziplin, der Informatik.**

In der Informatik wird heute eine Fülle algorithmischer Verfahren studiert, die nicht mehr unmittelbar in den Turing’schen Rahmen (des symbolischen Rechnens) passen.