Algorithmus: *Vortrag über Algorithmik*

Armin P. Barth

15. Januar 2004

Initialisierung: Seien dies die behandelten Themen:

- Geschichte der Algorithmik
- Was ist ein Algorithmus?
- Was versteht man unter dem Begriff 'berechenbar'?
- Bewährte Beispiele von Algorithmen
- praktische und theoretische Grenzen der Algorithmik

begin Vortrag

1 STEP 1: Traumdeutung: Von Leibniz' Traum bis zu Turings Deutung

> Leibniz ist der konservativste Revolutionär der abendländischen Geistesgeschichte gewesen. Aus jedem Kiesel Funken schlagend und auf eine ihm eigene Art mit diesen Funken überall Lichter entzündend, die noch keiner vor ihm entzündet hatte. Ein erleuchtender grosser Positivist, wenn unter einem Positivisten ein Mensch verstanden wird, der überall das Positive sieht und zu Ehren bringt.

Gottfried Wilhelm Leibniz, über den Heinrich Scholz so positiv urteilt, hatte einen Traum, eine grossartige Idee, die, obwohl unrealisierbar, den Anstoss zu vielen Überlegungen gab, die später die Theorie der Algorithmen befruchten sollten. Ihm fiel auf, dass mathematische Schlussfolgerungen deshalb so sicher sind, weil sie aus Berechnungs- und Beweisschritten aufgebaut sind, und es schien ihm plausibel, dass auch Argumente in nicht-mathematischen Disziplinen sicherer und stringenter gemacht werden könnten, wenn diese Disziplinen ebenfalls über die Möglichkeit zu rechnen und zu beweisen verfügen würden. Nun ist diese Möglichkeit natürlich nicht leicht zu realisieren, denken sie nur an Disziplinen wie Politik, Theologie oder Philosophie. Voraussetzung ist eine künstliche Sprache, in der sich gedankliche Prozesse der jeweiligen Disziplin so ausdrücken lassen, dass Rechenschritte, besser: formale Umformschritte, möglich werden, die auf rechnerische Weise zu neuen gedanklichen Inhalten führen. Den Objekten und Relationen der Disziplin müssen also Objekte und Relationen innerhalb der Kunstsprache und den mechanisch bildbaren Aussagen der Kunstsprache müssen sinnvolle Aussagen der Disziplin entsprechen, und die mechanisch vollziehbaren Umformschritte der Aussagen der Kunstsprache müssen wie bei mathematischen Beweisen zu neuen Aussagen führen, die, wenn sie als Aussagen der Disziplin interpretiert werden, neue Einsichten in innerdisziplinäre Zusammenhänge ermöglichen.

Leibniz schwebte also nichts Geringeres als eine Art *Gedankenrechnung* vor, die die gesamte Metaphysik endlich auf sicheren Grund stellen würde, so dass bei einer allfälligen Uneinigkeit in einer metaphysischen Frage bloss gerechnet zu werden braucht, um die Frage eindeutig und definitiv zu entscheiden. Leibniz selber schrieb[1]:

> Et si quelqu'un doutait de ce que j'aurais avancé, je lui dirais: Contons, Monsieur, et ainsi prenant la plume et de l'encre, nous sortirions bientôt d'affair.

Versuchen wir, kurz und prägnant zu sagen, was Leibniz also für jede Disziplin zu schaffen beabsichtigte: Zunächst sollte eine Kunstsprache hergestellt werden, in der festgelegt ist, welche Symbole benutzt und wie aus den Symbolen Wörter und Aussagen gebildet werden sollen - und wie die Objekte der Kunstsprache inhaltlich zu interpretieren sind. Und dann sollte ein *calculus ratiocinator*, also eine Art intelligenter Algorithmus, geschaffen werden, durch dessen Ausübung das unsichere inhaltliche Schliessen innerhalb der Disziplin ersetzt wird durch ein sicheres Rechnen mit den Symbolen, Wörtern und Aussagen der Kunstsprache. Beide Instrumente zusammen sollten eine *mathesis universalis* bilden, eine Erweiterung des Kompetenzbereichs des mathematischen Denkens auf alle nicht-mathematischen Disziplinen der Menschheit.

Welch kühner Plan! Stellen wir uns nur vor, man könnte jede politische, wirtschaftliche, juristische, moralische, ethische, theologische, philosophische Frage durch eine mühelose Rechnung innerhalb der mathesis universalis entscheiden. Wir wären ja dann im Besitz eines Algorithmus, der jede Frage beantwortet und jedes Problem löst, das den menschlichen Geist jemals herausfordert. Es dürfte nicht erstaunen, dass Leibniz dieses Ziel nie erreicht hat. Heute wissen wir, dass er es nicht erreichen konnte, auch wenn Gott ihm ein viel längeres Leben gegeben hätte. Immerhin sind aber später einige bescheidene Teilerfolge erzielt worden.

[1] L. Couturat, *Opuscules et fragments inédits de Leibniz*, Alcan, Paris 1903, pp.153ff

Im 19. Jahrhundert hatte George Boole, ohne etwas von Leibniz' Traum zu wissen, an dessen partieller Realisierung gearbeitet. Er war noch nicht einmal 16jährig, als er Lehrer wurde in einer kleinen Methodistenschule 40 Meilen entfernt von zu Hause. Dort entliess man ihn zwar nach zwei Jahren, weil er es nicht lassen konnte, sich auch Sonntags und sogar in der Kirche der Mathematik zu widmen, aber während dieser kurzen Periode wurde Boole inspiriert von der Idee, dass es möglich sein müsste, logische Beziehungen mittels Algebra auszudrücken. Nach eigener Aussage soll ihn diese Idee während eines Spaziergangs über ein Feld wie ein Blitz getroffen haben, und ein Biograph verglich diese Erfahrung später mit der Erfahrung, die Paulus in Damaskus machte.

Natürlich realisierte Booles Algebra nur einen kleinen Teil von Leibniz' Traum, denn berechnen lässt sich damit die Wahrheit oder Falschheit nur von Aussagen sehr spezieller Struktur. Bei vielen Aussagen der Prädikatenlogik versagt Booles Kalkül, und deshalb hätte sich Leibniz mit ihm niemals zufrieden gegeben.

An Freges Arbeit kurz nach Booles Algebra hätte Leibniz schon mehr Freude gehabt. Frege entwickelte kurz nach Booles Arbeit seine sog. *Begriffsschrift*, und im Gegensatz zu Boole kannte er Leibniz' Traum genau und schrieb seine Schrift in der erklärten Absicht, etwas Wesentliches beizusteuern zur Realisierung dieses Traums. Frege notierte in der Einleitung zur Begriffsschrift:

> Auch Leibniz hat die Vorteile einer angemessenen Bezeichnungsweise erkannt, vielleicht überschätzt. Sein Gedanke einer allgemeinen Charakteristik, eines calculus $(\cdots)$ ratiocinator war zu riesenhaft, als dass ein Versuch ihn zu verwirklichen über die blossen Vorbereitungen hätte hinausgelangen können. $(\cdots)$ Wenn aber auch dies hohe Ziel mit einem Anlaufe nicht erreicht werden kann, so braucht man doch an einer langsamen, schrittweisen Annäherung nicht zu verzweifeln. $(\cdots)$ Man kann in den arithmetischen, geometrischen, chemischen Zeichen Verwirklichungen des Leibnizschen Gedankens sehen. Die hier vorgeschlagene Begriffsschrift fügt diesen ein neues hinzu und zwar das in der Mitte gelegene, welches allen anderen benachbart ist.

Wenn man die Begriffsschrift zu lesen versucht, ist man zunächst einmal vor den Kopf gestossen. Frege entwickelt nämlich eine künstliche Sprache mit einer präzisen Syntax, in der logisches Schliessen als ein mechanistisches Manipulieren von Symbolen abläuft. So wird beispielsweise der *Modus Ponens* benutzt, bei dem aus der Annahme, dass die beiden Aussagen A und $A \to B$ gesichert sind, auf die Gültigkeit von B geschlossen wird.

Freges Kunstsprache wurde in der Folge zur Standardsprache, welche Studenten der Mathematik und der Philosophie gelehrt wurde, und dank ihrer Möglichkeit zur mechanistischen Symbolverarbeitung wird sie oft als wichtigster Vorläufer moderner Computersprachen bezeichnet.

Zum ersten Mal schien Leibniz' Traum realisierbar, wären da nicht drei Einwände, die Frege selbst noch zu Lebzeiten schrecklich bewusst wurden und von denen er sich nie mehr erholte:

1. Dank Freges Sprache konnte bei vielen Konklusionen gezeigt werden, aufgrund welcher Regeln sie aus welchen Annahmen folgen. Was aber, wenn dies für eine bestimmte Konklusion nicht gelang? Lag das dann daran, dass es nicht beharrlich genug versucht worden war, oder war es vielleicht innerhalb des Systems prinzipiell unmöglich? Darauf konnte Frege keine Antwort geben, und natürlich hätte Leibniz diesen Mangel niemals dulden können.

2. Im Jahr 1902 zerstörte ein Brief von B. Russell Freges Hoffnung, eine saubere und widerspruchsfreie logische Grundlage für die gesamte Mathematik legen zu können. Er hatte den Plan, die natürlichen Zahlen in rein logischen Ausdrücken zu definieren und alle Eigenschaften dann mit Hilfe seiner Kunstsprache abzuleiten, um schliesslich den gesamten Rest der Mathematik miteinzubeziehen. (Die Geometrie freilich gedachte Frege separat zu behandeln, obwohl man sie ja mit Hilfe der Koordinaten auf die Arithmetik zurückführen könnte.)

   Dann aber fand Frege Russells Brief vor. Und darin zeigte Russell, dass in Freges Aufbau logische Widersprüche unausweichlich waren. Frege hatte das Konzept der *Menge von Mengen* benutzt, und Russell zeigte, wie daraus Inkonsistenzen abgeleitet werden können: Nennt man nämlich eine Menge *ausserordentlich*, gewenn sie Element von sich selber ist, und *ordentlich*, gewenn sie das nicht ist, so führt das Konzept der Menge von Mengen zu zerstörerischen Widersprüchen:

   M *extraordinary* : $M \in M$
   M *ordinary* : $M \notin M$

   Example: Let M be the set of all things that can be defined in fewer than 19 english words.
   $\Rightarrow M$ is extraordinary!

   Let ϵ be the set of all ordinary sets. Then:
   ϵ extraordinary $\Rightarrow \epsilon \in \epsilon \Rightarrow \epsilon$ ordinary
   ϵ ordinary $\Rightarrow \epsilon \notin \epsilon \Rightarrow \epsilon$ extraordinary

3. Leibniz hatte eine Sprache gefordert, in der auch die komplexeste Frage durch einfaches Rechnen, also durch einfache Manipulation von Symbolen geklärt werden kann. Freges Sprache aber ist unerträglich kompliziert! Schon die einfachsten Konklusionen werden zu wahren Herausforderungen und können in vernünftiger Zeit weder gefunden noch notiert werden.

Aus all diesen Gründen konnte Freges Ansatz keinen dauernden Erfolg haben, und Leibniz' Traum harrte weiterhin seiner Erfüllung.

1928 fand in Bologna ein internationaler Mathematikkongress statt. Für David Hilbert war es der erste Kongress seit 16 Jahren, weil der Kongress von 1916 aus Kriegsgründen ausfiel und an den Kongressen von 1920 und 1924 die deutschen Mathematiker aus einer Art von Rache nicht eingeladen waren. Deshalb ergriff Hilbert die Gelegenheit 1928 eifrig und formulierte in Bologna ein Problem, das, aus seiner Sicht, dringend gelöst werden musste. Er schlug ein formales System vor, das, basierend auf der Prädikatenlogik 1. Stufe aus Freges Begriffsschrift und den Peano-Axiomen der Zahlentheorie, in der Lage sein sollte, in rein mechanistischer Weise Arithmetik zu treiben. Und Hilbert verlangte, dass die *Vollständigkeit* dieses Systems zu beweisen sei:

> Let PA^+ be the system of Peanoaxioms plus first order logic from the Begriffsschrift.
>
> Prove for any proposition $\mathbb{P}$ in PA^+ that
> $PA^+ \triangleright \mathbb{P}$ or $PA^+ \triangleright \neg\mathbb{P}$ ($\triangleright$ means *provable*)
> i.e. PA^+ is complete.

Was zwei Jahre später geschah, entspricht nicht im mindesten Hilberts Wünschen, und es verbitterte ihn dauerhaft. Der tschechisch-österreichische Mathematiker Kurt Gödel bewies nämlich die Unvollständigkeit von PA^+, indem er eine konkrete Proposition konstruierte, die in PA^+ weder beweisbar noch widerlegbar ist.

Gödel trug seine Entdeckungen in Königsberg vor, in derselben Woche, in der Hilbert, ebenfalls in Königsberg, ein Symposium eröffnete und dabei seiner Überzeugung Ausdruck verlieh, die später auf seinen Grabstein eingemeisselt werden würde:

> Wir müssen wissen; wir werden wissen.

Gödels Sätze machten eindringlich klar, dass ein mathematischer Kalkül mit klar definiertem Zeichensatz, klar definierten Bildungsregeln für Terme und Formeln, klar definierten Axiomensystemen und Schlussregeln, der mächtig genug ist, die Peano-Arithmetik und die Prädikatenlogik 1. Stufe abzubilden, zwingend Unvollständigkeit nach sich zieht, dass als Leibniz' Traum unmöglich in Erfüllung gehen kann, nicht einmal für gewisse Teilbereiche der Mathematik, die dafür doch am ehesten geeignet erscheint.

Bisher war noch gar nie von *Algorithmen* die Rede! Wir können Leibniz' Traum aber auch mit Hilfe dieses Begriffes formulieren, nämlich: Gesucht ist ein Algorithmus, der jede beliebige menschliche Frage zu entscheiden gestattet. Freilich wird dadurch nichts klarer. Und auch wenn Behmann 1922 einen Algorithmus forderte, der über die Richtigkeit oder Falschheit einer beliebig vorgelegten Behauptung nach einer endlichen Anzahl von Schritten zu entscheiden gestattet, und auch wenn er dieses sog. *Entscheidungsproblem* für das „Hauptproblem der modernen Logik“ hielt, so begleitet uns doch noch immer und immer mehr das unangenehme Gefühl, letztlich doch nicht genau zu verstehen, was ein Algorithmus ist.

Das war lange Zeit nicht weiter schlimm. Zu einem Zeitpunkt aber, als plötzlich die Vermutung aufkam, dass gewisse Probleme *nicht* algorithmisch lösbar sind, und man folglich die Nicht-Existenz eines Algorithmus beweisen sollte, da war eine exakte Definition auf einmal unerlässlich. Denn wie will man beweisen, dass etwas, von dem man nicht genau weiss, was es ist, nicht existiert! In den 30er-Jahren des letzten Jahrhunderts wurden immer weitere algorithmisch unlösbare Probleme bekannt. Damit wurde Leibniz' Traum ein für allemal verscheucht, und gleichzeitig musste eine exakte Vorstellung davon entwickelt werden, was genau ein Algorithmus ist.

Das bringt uns zu Alan Turing, denn eine der klassischen exakten Definitionen von Algorithmus lautet:

> Definition: Ein *Algorithmus* ist eine anhaltende Turing-Maschine (TM).

Es muss gleich angefügt werden, dass andere Mathematiker andere Definitionen vorschlugen, Church die „λ-Definability", Markov den „normal algorithm" und Gödel die „rekursiven Funktionen". Aber es hat sich gezeigt, dass all diese Definitionen untereinander äquivalent sind, dass sie also dieselbe Klasse von Problemen erfassen. Das ist ein starkes Argument dafür, dass man sehr genau verstanden hat, was ein *automatisierbarer Prozess* ist, und unter anderem darum konnte Church im Jahr 1936 seine berühmte These formulieren, wonach die Klasse der von einer TM entscheidbaren Probleme genau die Klasse derjenigen Probleme ist, die im intuitiven Sinne als *berechenbar* gelten. Der erst vage Begriff der *Berechenbarkeit* kann also ersetzt werden durch den präzisen Begriff der *TM-Berechenbarkeit*, und damit ist sehr viel gewonnen, denn nun kann von gewissen Problemen bewiesen werden, dass sie nicht berechenbar, also nicht algorithmisch lösbar sind, und nun kann der Algorithmus an sich zum lohnenden Studienobjekt gemacht werden.

Nun ist es an der Zeit, die TM zu definieren, damit obige Definition des Algorithmus sinnvoll und verständlich wird:

Definition: Unter einer Turing-Maschine (TM) versteht man ein 6-Tupel $(Q, A, \bar{A}, B, q_0, \delta)$ mit

- $Q \neq \{\}$: endliche Menge sog. Zustände q_i
- A: endliches Alphabet der Eingabe- und Ausgabesymbole, meist $A = \{0, 1\}$
- $\bar{A} \supseteq A$: endliches Rechenalphabet zur Notation der Zwischenschritte
- $q_0 \in Q$: Startzustand
- $B \in A$: Blankzeichen
- $\delta : Q \times \bar{A} \to Q \times \bar{A} \times \{L, R, S\}$: partielle Programmfunktion, die einigen 2-Tupeln (Zustand, Zeichen) ein 3-Tupel (neuer Zustand, neues Zeichen, Bewegung) zuordnet.

Es ist wichtig zu betonen, dass die TM ein abstraktes Denkmodell ist. Keinesfalls geht es darum, möglichst viele dieser Maschinen konkret zu bauen. Vielmehr halten wir mit der TM nun ein Modell für algorithmisches Berechnen in Händen, das es erlaubt zu beweisen, dass gewisse Aufgaben nicht algorithmisch berechenbar sind, und das es erlaubt, gewisse allgemeine Analysen über Algorithmen durchzuführen, wie etwa Analysen des zeitlichen Aufwandes oder des benötigten Speicherplatzes. Turing, der seine Maschinen natürlich nicht Turing-Maschinen, sondern *automatic machines* nannte, hat überdies gezeigt, wie diese Maschine so erweitert werden kann, dass sie nicht nur ein einziges Problem löst, sondern ein beliebiges codiertes Programm akzeptiert und ausführt, und damit war zum ersten Mal entstanden die revolutionäre Idee des Universalcomputers, der jedes beliebige Problem löst, für das ihm ein Algorithmus eingegeben wird. Man kann, denke ich, mit Recht sagen, dass dies die Geburtsstunde des modernen Computers war.

2 Step 2: Erfahrungen mit Turing-Maschinen im Unterricht

3 Step 3: Algorithmisch unlösbare Probleme

4 Step 4: Beispiele von Algorithmen im Unterricht

5 Step 5: Spezielgebiete der Algorithmik

If überzogen, then entschuldigen,
else auf Fragen eingehen.

end Vortrag