

Algorithmen, Strukturen, Maschinen: Inhalte der Informatik¹

Friedrich L. Bauer, T. U. München

1 Einleitung

Die Informatik ist eine junge Wissenschaft, der Bezeichnung nach nicht älter als dreissig Jahre. Sie formierte sich zwischen 1936 und 1951. Ihre Wurzeln sind allerdings viel älter, zu einem guten Teil liegen sie ausserhalb der Mathematik: in der Logik, der Kryptologie, der Instrumentenkunde, der Feinmechanik, der Nachrichtentechnik.

Neu war um die Mitte des Jahrhunderts die Zusammenschau vieler vorher isoliert betrachteter Ideen und Phänomene.

Ich will in diesem Vortrag versuchen, Ihnen die breite Basis der Informatik, ihre Grundinhalte näherzubringen. Ich konzentriere mich dabei auf drei Begriffe: Algorithmus, Struktur, Maschine. Unter Hintanstellung meiner beruflichen Arbeitsmethode, der Formalisierung, will ich (fast) ohne Formelkram auskommen und deshalb die phänomenologische Seite in den Vordergrund stellen. Die Fachgenossen mögen mir verzeihen, wenn dadurch für sie das Gleichgewicht zwischen formalem und intuitivem Verständnis nicht gewahrt zu sein scheint.

2 Ursprünge des Zählens

Zählen ist eine Errungenschaft des Menschen, eine relativ spät einsetzende geistige Leistung des homo sapiens. Es entstand unabhängig in verschiedenen Kulturkreisen, wie wir aus dem Auftreten verschiedener Zahlssysteme wissen: bei den Azteken, in China, in Ägypten, bei den Sumerern.

Vom Zählen der Tage zwischen den Neumonden und dem Zählen der Neumonde zwischen den Äquinoktien geht der Anstoss aus zur Betrachtung der Sterne; da der Himmel das Reich der Götter ist, bekommt auch das Zählen göttliche Aspekte, Zahlen werden mystische Symbole. So könnte man Kroneckers Ausspruch verstehen: «Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk.» Gegenüber dem bis heute fortbestehenden Mystischen gewinnt aber die praktische Seite der Zahlen bald die Oberhand.

Der Umgang mit Münzen und Gewichten, die Chronologie und die Landmessung bestimmen in der Antike und im Mittelalter, wohin sich die Mathematik entwickelt. Zum Rechnen benutzt man den Abakus, das Rechenbrett.

3 Algorithmen

In Indien entsteht aus sumerischen Anfängen, wohl einige Jahrhunderte nach Christi Geburt, eine neue Art, Zahlen zu schreiben: ein Stellenwertsystem. Schlüsselpunkt war die Erfindung eines Lückenzeichens, eines Zeichens für «Nichts», das indische sunya. Im Arabischen wurde das als sifr übersetzt, und daraus wurde chiffré, Ziffer, cipher, aber auch italienisch zefiro, zevero, woraus zéro und zero entstand.

¹ Festvortrag vor der Bayerischen Akademie der Wissenschaften; Nachdruck aus dem Jahrbuch 1987 (ohne Bildmaterial) mit freundlicher Genehmigung des Autors und des Verlegers.

Aber auch bei den Mayas in Mexiko sind Stellenwertschreibweise und ein Lückenzeichen zu finden – sicher unabhängig von Babylon.

Die Kunst, mit den neuen Ziffern zu rechnen, bekam den Namen *algorithmus*; die Leute, die das konnten, hiessen *algoristen*, im Gegensatz zu den *abacisten*. Die Bezeichnung rührt her von Abu Dsch'far Mohammed ibn Musa al-Khowarizmi – einem Mann, der um 820 n. Chr. in der ostpersischen Region Khowarizm lebte, nahe der Stadt Khiva in der heutigen Sowjetrepublik Uzbekistan. Um 1100 wurde al-Khowarizmis Buch «Hisab al dschabr w'al muqabala» als «algebra et almucabala» ins Lateinische übersetzt, und schon 1202 pries Leonardo di Pisa, Hofmathematiker des Kaisers Friedrich II., besser bekannt als Fibonacci, die neue Kunst in den höchsten Tönen. Im 16. Jahrhundert war der Kampf zwischen dem alten und dem neuen Rechnen voll entbrannt. Erst die Französische Revolution lässt Rechenbretter und Rechenpfennige (Leg-Geld) in West- und Mitteleuropa verschwinden; in Osteuropa, in Asien hält sich der Abakus bis heute.

Algorithmen nannte man dann später das Verfahren zum Rechnen mit Ziffern. Die Universität Altdorf lockte um 1550 Studenten an mit dem Versprechen, sie würden dort das Dividieren erlernen. Noch Leibniz spricht vom «Algorithmus der Multiplikation». Über die Jahrhunderte hinweg nahm dann das Wort Algorithmus eine allgemeinere Bedeutung an; so spricht man – obwohl Euklid von al-Khowarizmi noch nichts wusste – vom «Euklidischen Algorithmus» zur Bestimmung des grössten gemeinsamen Teilers zweier Zahlen und auch vom «Algorithmus der altägyptischen Multiplikation», die beide nichts mit dem Ziffernrechnen an sich zu tun haben. Algorithmus bedeutet heute «ein Spiel mit Zeichen nach festgesetzten Regeln». Damit gibt es auch Algorithmen für das Mühlespiel, für das Stricken eines Zöpfchenmusters, für das Aufsperrn eines Safes.

4 Die Befreiung des Menschen von der Last gleichförmiger geistiger Tätigkeit

Das Rechnen gilt zu Recht als geistige Tätigkeit, wenschon es ziemlich eintönig ist. Pascals Schwester etwa sagte, ihr Bruder habe die Rechenmaschine erfunden, weil er es leid gewesen sei, für seinen Vater, der Steuereinnahmer war, die Rechnung zu führen. Auch Schickard, Tübinger Professor der biblischen Sprachen, Pascal um 18 Jahre voraus, stand unter dem Eindruck umfangreicher Berechnungen, und zwar in der Astronomie. Bei Leibniz, der eine seiner Maschinen gar für den Kaiser von China bestimmt hatte, findet sich schon der Gedanke, dass geistige Routinearbeit des Menschen unwürdig sei. Automaten hatte schon das Rokoko gesehen, Androiden, Automaten in Menschengestalt, waren in der ersten Hälfte des 19. Jahrhunderts zur literarischen Mode geworden – bei Jean Paul, bei E. T. A. Hoffmann.

Ist Leibniz' Maschine als Hilfsmittel zur Durchführung des Multiplikationsalgorithmus, so strebte Charles Babbage um 1830 mehr an: die volle Automatisierung des Ablaufs eines allgemeineren Algorithmus. Er stützte sich bei seinen Versuchen auf einen Lochkartensteuerungs-Mechanismus, mit dem Jacquard um 1800 Webstühle automatisiert hatte. Er scheiterte an technischen Schwierigkeiten. Konrad Zuse, der 1934 Arbeiten aufnahm, die zum Erfolg führten, wusste damals nichts von Babbage. Drei Jahre später, 1937, bricht mit Alan M. Turing der Gedanke einer universalen Maschine durch, die *jeden* Algorithmus ausführen kann – nicht nur einen bestimmten und nicht nur einen arithmetischen. Was eine Turing-Maschine oder verwandterweise ein Markov-Algorithmus tut, erinnert in der Tat oft mehr an Stricken und Weben als ans Rechnen – man betrachte nur das über das Rechenband fahrende «Schiffchen» des Markov-Algorithmus für die Bildung der Ableitung eines Binärwortes. Schon Ada Countess Lovelace, Schülerin und Vertraute Babbages, hatte geschrieben «the analytical engine weaves algebraic patterns». War das Poesie oder Vision? Wohl das letztere, denn Lady Lovelace hatte schon erkannt, dass die Maschine «general symbols» kombinieren könne. Ihr folgend, könnte man sagen «Algorithmen weben Muster».

5 Strukturen

Muster gibt es vielerlei, in der Mathematik und anderswo. In der belebten Natur finden wir an einem kühlen Morgen ein Spinnennetz, von Tautropfen überzogen. Unter dem Mikroskop erweisen sich die Kieselsäureskelette der Radiolarien als bizarre regelmässige Strukturen.

Die Entdeckung des genetischen Codes brachte biologische Molekülstrukturen zum Vorschein. Heute benutzt man Rechner, um sie zu zeichnen. Bilder, die den strukturellen Aufbau der ringförmigen Ribonukleinsäuresequenzen von Viroiden (Exocortis-Krankheit der Citrusbäume, Spindelknottensucht der Kartoffel, Staudenkrankheit der Chrysantheme) zeigen, erinnern fast an musikalische Notation.

In der unbelebten Natur finden wir Kristalle und als ihr mathematisches Gerüst Kristallgruppen. Es gibt vielerlei Muster allein im Zusammenhang mit diskreten Gruppen, etwa Cayley-Diagramme, die die Verknüpfungsstruktur von Gruppen widerspiegeln, Polytope, Polyeder oder Tesselierungen einer hyperbolischen Ebene, wie sie Walther von Dyck, in der Nachfolge von Felix Klein, untersucht hat. Der holländische Grafiker M. C. Escher hat diese Strukturen künstlerisch verwandt und verwandelt.

6 Algorithmen erzeugen Strukturen

Im Zusammenhang mit nichtlinearen Iterationen sind in den letzten Jahren, angestossen durch Mandelbrot, viele eindrucksvolle Computergraphiken entstanden, darunter solche, die an Klöppelspitzen erinnern.

Unerschöpflich sind die von Menschen geschaffenen Strukturen beim Weben, Stricken, Häkeln, Knüpfen, Knoten, Klöppeln, Flechten. Für alle diese Arbeiten wurden Maschinen erdacht und somit Algorithmen festgelegt. Die Umkehrung würde zur These führen: *Was Algorithmen tun, ist die Erzeugung von Strukturen*. In der Tat dienen Strukturen oft als «gefrorene Algorithmen»: Tabellen sind Beispiele. Präziser können wir sagen: Die Informatik handelt von der Erzeugung neuer Strukturen aus alten. Aus Gründen der Denkökonomie werden diese Strukturen abstrakt aufgefasst. Sie mögen *Objektstrukturen* heissen, wobei die Objekte im jeweiligen Zusammenhang nicht weiter zu zerlegende Primitive sind. Es gibt vielerlei solche Objekte. Zahlen beispielsweise dienen häufig als Objekte – wenn es sich aber um den Aufbau eines Rechenwerkes handelt, sind Zahlen Objektstrukturen, gebildet aus Ziffern.

Abweichend von unserem gebräuchlichen Dezimalsystem kommt das Dualsystem mit bloss zwei Ziffern, Null und Eins, aus. Das war Leibniz' Entdeckung in der *arithmetica dyadica*, 1679. Leibniz sah darin ein weiterreichendes philosophisches Prinzip, das wir heute Binärprinzip, salopp auch Ja-Nein-Prinzip, Alles-oder-Nichts-Prinzip nennen. So ist sein «anchora sacra», das «heilige Noch», vielleicht die Ahnung, dass alle effektiv handhabbaren Objekte als Strukturen aus dyadischen – wir sagen heute binären – Elementen aufgefasst werden können; es ist jedenfalls die Vorwegnahme der formalisierten zweiwertigen («dyadischen») Aussagenlogik. Wie man für das Dualsystem eine arithmetische Maschine bauen kann, hat Leibniz schon angegeben. Leibniz als Vorläufer von Boole, Turing und Zuse zugleich – wahrlich ein grosser Mann.

Dezimalzahlen und auch Dualzahlen sind deutlich gegliederte Strukturen. Schon in den Anfängen der Informatik betrachtete man gegliederte Daten, etwa Konrad Zuse im Plankalkül «Angablagen» für die Bestandteile technischer Strukturen oder für Feldbesetzungen beim Schach. In ALGOL 60 findet sich das wieder in den dynamischen mehrstufigen Feldern, die gut geeignet sind, Matrizen und Tensoren darzustellen. Nichtnumerische Anwendungen führten Mitte der fünfziger Jahre auf die Verwendung von Listen mit Verweisen. Dijkstra und Knuth verdanken wir einleuchtende Illustrationen des funktionellen Verhaltens solcher Strukturen am Beispiel von Gleisanlagen. Aber erst allmählich bildet sich die Abstraktion heraus, die sich vom Zwang des Speicherskeletts frei macht und zu den rekursiven Datenstrukturen von Hoare 1970 führt, zu deren Definition eine bloss Formel genügt, die den erzeugenden Algorithmus beschreibt. In diesen Strukturen treten nicht mehr allein Ziffern und Zahlen, sondern beliebige Zeichen und Zeichengruppen auf.

7 Rechnerstrukturen

Auch Rechengeräte weisen Strukturen auf. Dies zeigt ein Blick auf den mechanischen Speicher von Konrad Zuse. Deutlich kommt es in den Schaltbildern für elektromagnetische und elektronische Rechenanlagen zum Ausdruck. Man sieht es auch am äusseren Erscheinungsbild der PERM, der von Hans Piloty und meinem verehrten Lehrer und väterlichen Freund Robert Sauer stammenden Maschine aus den frühen fünfziger Jahren. Seit den sechziger Jahren findet überdies mit dem Aufkommen der gedruckten und geätzten Schaltungen eine Konvergenz der Schaltungen gegen die Schaltbilder statt. In der heutigen Mikrominiaturisierung sind Schaltung und Schaltbild eins geworden. Auf kleinstem Raum sind Tausende von Schaltelementen vereinigt, eindrucksvolle Muster bildend. Hochkomplizierte Algorithmen erzeugen diese Muster, die als Vorlagen für photolithographische Prozesse dienen. Im Rasterelektronenmikroskop kann man die Oberfläche der so auf einem Siliciumplättchen entstehenden Schaltung sichtbar machen. Dabei spielen Abmessungen unter $1\ \mu\text{m}$ eine Rolle, gut zehnmals feiner als die von Nervenfasern. Der Grössenvergleich mit Nervennetzen mag beeindruckend sein, sagt aber wenig über funktionelle Analogien. Insbesondere arbeitet das Zentralnervensystem nicht nach dem «Alles-oder-Nichts-Prinzip» der Computer. Für Computer gibt es keine Psychopharmaka; dass sie Launen haben mögen, ist eine Mär.

8 Strukturen der Algorithmen

Algorithmen weben Muster, haben wir festgestellt. Kein Wunder, dass sie selbst Struktur tragen. Allerdings wäre es einfach anzunehmen, die Erzeugnisse von Algorithmen wären blosse Abbilder, der Replikation von Desoxyribonukleinsäure-Ketten vergleichbar. Algorithmen fangen typischerweise den Aufbau von komplizierten Mustern auf einfachere Art ein: Das Zauberwort heisst Rekursion. Als «Bild im Bild» ist dieses Prinzip seit dem Altertum bekannt und künstlerisch wie literarisch ausgebeutet worden.

Nichtsdestoweniger zeigen Algorithmen Gesetzmässigkeiten des Aufbaus, Strukturen. In Maschinenprogrammen wird diese Struktur verwischt. Ablaufpläne – und mehr noch symbolisch notierte Algorithmen – bringen diese innere Struktur zum Ausdruck.

Programmieren ist nichts anderes als Strukturieren. Es bringt Struktur in ein amorphes Problem – so viel Struktur zumindest, dass damit eine operative Problemlösung entsteht. Hat man schon eine Problemlösung, so kann Programmieren trivial sein: es entartet zum Kodieren, zur rein notationalen Aufgabe.

9 Programmieren als Entwicklungsprozess – Programmieren als Kalkül

Häufig muss jedoch eine Problemlösung melioriert werden: es muss nach effizienteren Fassungen gesucht werden. Darin liegt die Kunst des Programmierens. Der oben erwähnte Algorithmus der altägyptischen Multiplikation, durchgeführt etwa mit römischen Zahlen, ist bereits eine Meliorierung des trivialen Verfahrens der 43mal wiederholten Addition.

Programmieren ist also, recht verstanden, stets ein solcher Entwicklungsprozess. Entwickelt wird die Struktur des Algorithmus, aber auch die Struktur der Objekte, mit denen dieser Algorithmus arbeitet. Oft gehen diese beiden Entwicklungen Hand in Hand, etwa wenn der effizientere Algorithmus der ägyptischen Multiplikation durch den Leibnizschen Übergang zur Dualzahl-darstellung unterstützt wird.

Programmentwicklung bedeutet in der Regel Verfeinerung der Programm- und Objektstruktur. Damit wird aber Programmentwicklung selbst zum Gegenstand eines Algorithmus. Rutishauser und Ershov bezeichneten schon Anfang der fünfziger Jahre zu Recht ihre Compiler als «programmierende Programme». Weit darüber hinausgehend, wird heute Programmieren zu einem rationalen Prozess, einer schrittweisen Transformation von Strukturen, vom geeignet strukturierten Problem zur Maschine führend. *Programmieren selbst bekommt algorithmische Züge.* Wir können

uns durch Abstraktion unabhängig machen von zeitweiligen technologischen Zwängen, die bisher die Programmierung stranguliert haben. Programmieren als Kalkül wird insbesondere überall dort unentbehrlich, wo Programmierfehler unerträglich sind, so in sensitiven Bereichen politischer, militärischer und medizinischer Natur, oder aus ökonomischen Gründen, wie bei der Ausarbeitung von Schaltungen, die in grossen Serien gefertigt werden.

10 Informatische Maschinen

So anziehend mancher Informatiker das abstrakte Spiel mit Algorithmen und Strukturen finden mag: es dient letztlich doch der Durchführung *konkreter* Aufgaben auf informatischen Maschinen. Im Gegensatz zu den Kraftmaschinen, die die erste industrielle Revolution mit sich brachten, sind die informatischen Maschinen energetisch sehr genügsam. Zwar brauchte unsere Münchner PERM als Röhrenmaschine im Jahre 1955 noch mehr Strom als ein Bauerndorf, aber das war nur eine Kinderkrankheit; heute kann die gleiche Rechenleistung energetisch mit Solarzellen erbracht werden. Informatische Maschinen führen zu keiner Umweltverschmutzung, sie passen in schöne Landschaften.

Informatische Maschinen haben die Aufgabe, Algorithmen auf Strukturen auszuführen. Sie können hierbei so ineffizient sein wie die lediglich zu theoretischen Zwecken dienende «Turing-Maschine» oder so effizient wie die «von Neumann-Maschine», ein Baumuster, das sich über nunmehr vierzig Jahre mit nur kleinen Abwandlungen gehalten hat. Der praktische Informatiker braucht jedenfalls Maschinen, um sein Brot verdienen zu können – ohne Hardware ist Software nutzlos.

Maschinen haben zwei Seiten: einerseits ihren inneren Aufbau, andererseits ihr äusseres Verhalten, wenn sie als «schwarzer Kasten» aufgefasst werden.

Vom inneren Aufbau her haben wir Maschinen schon als Beispiele für Strukturen angeführt. Abstrakt spricht man von Schaltstrukturen. Der Abstraktionsprozess verleiht Unabhängigkeit von technologischen Entwicklungen. Das bemerkte schon Zuse, der eine abstrakte Schaltkreisnotation verwandte, als er von mechanischen Schaltgliedern zu elektromagnetischen überging und 1943 Schreyers elektronische Lösungen ins Auge fasste. Schecher beschrieb 1969 einen Kleinrechner in ALGOL 68. Eine solche abstrakte Beschreibung der *Schaltung*, bis hinab zum letzten Bit, passt auf *eine* Seite.

11 Rechenstrukturen: Strukturen zum Rechnen

Wir haben aber auch gelernt, das äussere Verhalten einer Maschine als «schwarzer Kasten» abstrakt und formal zu erfassen mit dem Begriff einer Rechenstruktur, einem Konglomerat von Rechenoperationen und Objektstrukturen, das lediglich durch die Eigenschaften ihres Zusammenspiels gekennzeichnet ist – Eigenschaften, die man etwa in der Form von Gleichungen abfasst.

Wiederum braucht man für eine abstrakte Beschreibung, diesmal aber des *Verhaltens* eines Kleinrechners, nur eine halbe Seite. Der mathematische Hintergrund dieses Ansatzes, Rechenstrukturen als endlich erzeugbare Modelle solcher abstrakter Spezifikationen aufzufassen, führt zu ausgeprägt algebraischer Denkweise. Es entwickelt sich derzeit eine starke Bindung zwischen Informatik und Algebra, die neben die schon klassische zwischen Informatik und Logik (und die weniger oft beschworene, aber ebenfalls fruchtbare zwischen Informatik und Verbandstheorie) tritt.

Es wird also künftig möglich sein, dass Informatiker nicht nur Algorithmen beschreiben, sondern auch die Charakterisierung der Maschinen, für die diese Algorithmen bestimmt sind, festlegen. Die Nutzenanwendung einer souveränen Handhabung der Abstraktion wird vor allem im Zusammenhang mit der Mikrominiaturisierung und dem zunehmenden Trend zum individuelleren Bau von Rechenanlagen zu finden sein: Wir können auf diesem Weg gut durchorganisierte, programmierungsfreundliche Rechenanlagen erhalten, die optimal angepasst sind. Dies führt zu ei-

ner allmählichen Abkehr vom Konzept der «von Neumann-Maschine» und wird tief in den architektonischen Aufbau der Maschinen eingreifen.

12 Schluss

Der Dreiklang meines Vortrags lautet: Algorithmen, Strukturen, Maschinen. – Die Synthese beginnt sich abzuzeichnen: So wie Strukturen gefrorene Algorithmen sind, sind informatische Maschinen lebendige Algorithmen, die Strukturen verändern.

Was aber das Seitenthema betrifft: Ein weiter Weg war zu gehen, bis technologischer Fortschritt für viele Menschen die Befreiung von der Last eintöniger geistiger Tätigkeit brachte. Wie immer, übertreibt die Menschheit auch hier: Es gibt Leute, die sagen, man brauche nun in der Schule das Rechnen nicht mehr zu lernen. Dahinter kann die teuflische Absicht stehen, der Menschheit allmählich auch das Denken abzugewöhnen – für Ideologen und Diktatoren wahrlich schöne Aussichten.

Das Gegenteil wird jedoch durch die Freisetzung des Menschen von der Routine bewirkt: Er gewinnt Zeit und behält Energie, um sich stärker der wahren Mittel des Geistes, der Inspiration und der Intuition, zu bedienen. Computer sollen dem Menschen helfen, sich in komplexen Situationen zurechtzufinden. Den *Computer* nicht zum mystischen Gegenstand werden zu lassen, ist unsere Aufgabe. Die Fachleute trifft die Verpflichtung, neben ihrem beruflichen Fortschrittsglauben solide staatsbürgerliche Verantwortung zu zeigen und auch der Öffentlichkeit gegenüber zu vertreten – einer *breiten* Öffentlichkeit, die auf das Urteil der Fachleute angewiesen ist. Dann wird man nicht glauben, Computer seien ein Allheilmittel.

Die Fachgenossen und eine *interessierte* Öffentlichkeit sollten sich indessen vor Augen halten, dass es Grenzen des Verstandes gibt. Die Logik hat mit den Gödelschen Resultaten erstmals Grenzen der Beweisbarkeit aufgezeigt; im Verfolg dieser Ergebnisse gibt es auch Grenzen der Algorithmisierbarkeit. Es gibt Problemklassen, die zu umfangreich sind, als dass ein einziger Algorithmus oder auch endlich viele, noch so allgemeine Algorithmen eine Lösung für die gesamte Klasse bringen könnten. Das Programmieren als allgemeine Aufgabe gehört übrigens dazu.

Je mehr wir also mechanisieren, automatisieren, algorithmisieren, um so mehr werden Inspiration und Intuition auch künftig erforderlich sein, um den Erkenntnisdrang des Menschen zu stillen. Nach wie vor hat der Geist Vorrang vor dem Golem. Ich möchte das dadurch unterstreichen, dass ich zum Schluss einer Computergrafik entgegenstelle das Bild der Rosette des Nordfensters im Querschiff von Notre-Dame in Paris.

Zudem möchte ich bemerken, dass nicht einmal das für die Informatik so fruchtbare Binärprinzip auf jede Lebenslage übertragen werden muss. Gerade die moderne Logik beschränkt sich nicht auf Zweiwertigkeit: modale Logiken finden neuerdings nützliche Anwendungen. Und im Zusammenleben der Menschen kommt man mit blossen Ja oder Nein nicht aus; die Wahrheit und die Liebe erfordern meist eine subtilere Antwort, die man mit Algorithmen allein nicht findet.