

Buch mit Diskette

Dr. Hartwig Thomas

Der britische Mathematiker Alan M. Turing hat den Computer erfunden. Ausserhalb eines engen Kreises von Spezialisten sind sein Name und Werk kaum bekannt. Sei 1987 ist eine deutsche Sammlung vorwiegend nicht-technischer Artikel Turings für „Laien“ als Geheimtipp unter Kennern im Umlauf. Alle abgedruckten Beiträge befassen sich mit der Frage, was Denkmaschinen sind und was sie sein könnten. Dabei spielt der Kernbegriff Emulation eine ausschlaggebende Rolle. Die Sammlung ist eine wichtige Grundlage für eine intelligente Auseinandersetzung mit intelligenten Maschinen auch ausserhalb einer technisch orientierten Insidergruppe.

In seinem Artikel „Über berechenbare Zahlen mit einer Anwendung auf das Entscheidungsproblem“¹ führt Alan Turing den Beweis, dass das berühmte „Entscheidungsproblem“ nicht lösbar ist. In einem sorgfältig edierten Band liegen dieser mathematische Artikel und andere Schriften Turings, die sich an ein nicht-mathematisches Publikum richten, nun auch auf deutsch vor². Die Herausgabe dieses Bands ist um so schätzbarer, als sich das deutschsprachige Verlagswesen bis heute recht schwer tut mit ernsthaften Überlegungen zur Computerentwicklung. Das Buch mit dem weissen Einband ohne Klappentext, mit dezenten Schwarzweiss-Photos, einem Vorwort und einem Nachwort über Alan Turing dokumentiert seine Anerkennung dieser Entwicklung durch eine beiliegende PC-Diskette, auf der sich vier Programme finden, die Turings Leistungen in der ihnen gemässen Form belegen.

Im Jahr 1900 zählte David Hilbert auf dem Internationalen Mathematikerkongress in Paris eine Reihe von wichtigen, in diesem Jahrhundert zu lösen, Problemen auf. Diesem Vater des mathematischen Formalismus und seinen Zuhörern war damals schon klar, dass die Mathematik eine Aktivität ist, die man vielleicht mit „Symbolmanipulation“ um-

schreiben könnte, auch wenn sie vorgibt, nur von Grössen und Zahlen zu handeln. Trotzdem hält sich sogar heute noch das Gerücht hartnäckig, dass Mathematik mit Rechnen und Zahlen zu tun habe. Hilbert hätte sich wohl kaum träumen lassen, dass sein berühmtes zehntes Problem, das Entscheidungsproblem schon wenige Jahre später und dazu noch negativ gelöst sein würde.

Das Hilbertsche Entscheidungsproblem:

„Eine diophantische Gleichung mit irgendwelchen Unbekannten und mit ganzen rationalen Zahlkoeffizienten 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.“

(Eine diophantische Gleichung ist eine Gleichung mit ganzzahligen Koeffizienten, in der nur die vier Grundrechenarten und ganzzahlige Exponentiation vorkommen. Unter „ganzen rationalen Zahlen“ versteht Hilbert, was wir heute schlicht „ganze Zahlen“ nennen.)

Der berühmte Beweis von Kurt Gödel³ zeigte, dass mit keinem Axiomensystem der Zahlentheorie alle Sätze entscheidbar sind⁴.

Im oben erwähnten Artikel führt Alan Turing in sehr klarer und unpräziser Weise die – seither so genannte – Turingmaschine als theoretische Konstruktion ein. Er identifiziert die Tätigkeit eines Mathematikers bei der Berechnung einer Zahl in einem Kalkül mit dem Funktionieren einer hypothetischen Maschine. Diese wird mit dem notwendigen Minimum ausgestattet, um dem Mathematiker geistig ebenbürtig zu sein: einem Speicher (vorgestellt als potentiell unendliches Band mit Feldern, auf denen Zeichen aus einem endlichen Alphabet stehen können, und das von der Maschine über einen Lesekopf Feld für Feld examiniert werden kann) und einer Steuereinheit, die endlich viele Zustände annehmen kann und vom Band lesen und auf das Band schrei-

1 Alan Turing, On Computable Numbers with an Application to the Entscheidungsproblem, Proceedings of the London Mathematical Society (2)42, 1937

2 Alan Turing, Intelligence Service, Hg. Bernhard Dotzler und Friedrich Kittler, Brinkmann & Bose, Berlin, 1987

3 Kurt Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, Monatshefte Math. Phys., 38 (1931), 173-198

4 Eine bemerkenswerte Variation dieses Resultats wurde 1987 von Gregory J. Chaitin vorgestellt zusammen mit einem expliziten Beispiel einer diophantischen Gleichung (5 Seiten), von der man zeigen kann, dass es unmöglich ist, zu entscheiden, ob sie endlich oder unendlich viele ganzzahlige Lösungen besitzt in:

G. J. Chaitin, Algorithmic Information Theory, Cambridge Tracts in Theoretical Computer Science 1, Cambridge University Press, 1987

ben kann. Eine solche Turingmaschine ist in ihrem Funktionieren eindeutig festgelegt, wenn man für jeden Zustand und jedes gelesene Zeichen festhält, in welchen neuen Zustand die Maschine beim nächsten Schritt übergeht. Da das Zeichenalphabet und die Zustandsmenge der Steuereinheit endlich sind, kann eine solche Maschine mit einer endlichen Liste von Zustandsänderungen beschrieben werden. Jede Turingmaschine verarbeitet einen Input auf dem Band zu einem Output, und kommt beim Endzustand zum Halten, sofern sie nicht in einer endlosen Ramifikation stecken bleibt.

Nachdem Turing gezeigt hat, dass es für eine Reihe von elementaren Problemen angebbare Maschinenbeschreibungen gibt (Kopiervorgänge, arithmetische Operationen...) holt er zum grossen Schlag aus, der die gödelsche Argumentation veranschaulicht: Er stellt fest, dass seine Maschinen nicht nur reichlich immaterielles Material verarbeiten, nämlich Symbole auf einem Band, sondern, dass sie selbst auch aus ähnlich immateriellem Material bestehen: sie können durch eine Folge von Symbolen vollständig beschrieben werden (in seinem Artikel hat er ja auch nur mit solchen Beschreibungen gearbeitet, da 1937 noch keine realen Computer zur Verfügung standen).

Nichts naheliegender also, einer Maschine den Bauplan einer anderen Maschine als Input zu „füttern“. Dank dieser überraschenden Wendung stiess Turing zum Wesen der Symbolmanipulation vor. Er erfand die Universelle Turingmaschine. Es stellt sich als einfach heraus, eine Turingmaschine zu konstruieren, die nach Lesen des Bauplans einer beliebigen anderen Turingmaschine, diese „nachmacht“, d.h. sich in jedem Punkt so verhält, als ob sie diese spezielle Turingmaschine wäre. Dieser Vorgang und diese Idee sind uns im Computerzeitalter derart vertraut, dass wir das Erstaunen darüber vergessen haben. Die Bezeichnung Emulation hat sich für dieses Verfahren eingebürgert.

Mit dieser Universalmaschine hat Turing den modernen Computer erfunden. Alle modernen Computer sind ihrem Konzept gemäss Turingmaschinen (in der Praxis setzen mechanische Abnutzung und endliche Menge von zu Verfügung stehendem Speicher, etwa Magnetband oder Disketten, diesem Konzept Grenzen). In dem halben Jahrhundert seit dieser Erfindung wurden verschiedenste Versuche unternommen, den Bereich der im Prinzip von Menschen berechenbaren Formeln, Zahlen, Sätze anders und allgemeiner als Turing zu formulieren.

Bis heute haben sich alle diese Versuche als der Turingmaschine äquivalent erwiesen. Obwohl man eine Definition eines Begriffs wie Berechenbarkeit

natürlich nicht beweisen kann, da es sich ja um eine Definition handelt, wird unter Mathematikern angenommen, dass alle „vernünftigen“ Definitionen dieses Begriffs auf dasselbe herauskommen. (Dieser Sachverhalt wird unter Mathematikern auch Church-Hypothese genannt, was zur weiteren Verwirrung beiträgt, da sich eine Definition natürlich auch nicht als „Hypothese“ formulieren lässt.)

Und der zentrale „Trick“ des Computerzeitalters ist schon in Turings erstem Artikel klar erkannt: der Computer kann als Maschine gebaut werden, die alles kann, als universelle Metamaschine also, die sich je nach Bauplan – heute nennt man das „Programm“ – mit der man sie füttert in eine konkrete Maschine verwandelt. Auch diese "konkreten" Maschinen können natürlich wieder mehr oder weniger universell sein. Ein Software-Hersteller, der es mit den Spezialvorstellungen seiner Kunden über irgendein Thema nicht verderben will, offeriert im allgemeinen eine „Makrosprache“ (Textverarbeitung) oder „Befehlssprache“ (dBase), in der man alles das selber programmieren kann, was vom Hersteller nicht vorgesehen war und behauptet dann stinkfroh, sein Programm könne alles, was dasjenige der Konkurrenz auch könne (man müsste es nur mit dem „Bauplan“ des Programms der Konkurrenz füttern, damit die Emulation läuft ...). Auch der PC-Benutzer ist sich heute in den seltensten Fällen darüber im Klaren, dass er für seine 2500 Franken mitnichten eine Maschine gekauft hat, die Buchhaltung „kann“ und CAD „kann“ und Typographie „kann“ und übers Telefon kommunizieren „kann“, sondern dass seine neue Errungenschaft mit Bauplänen konkreter Maschinen gefüttert werden muss, bevor sie zu etwas brauchbar ist. Von einem Neugeborenen sagen wir auch nicht, es „könne“ Lesen, obwohl wir die meisten von ihnen bei richtiger Programmierung dazu veranlassen können, diese Disposition zu verwirklichen. Um also die universelle Metamaschine zur benutzbaren Maschine zu machen, müssen wir noch einmal teures Geld auf den Tisch legen, um sie mit Programmen zu füttern. Beim Auto hat man sich daran gewöhnt, dass es nicht ohne Benzin fährt. Beim PC tun sich viele Benutzer noch schwer mit dem Softwarekauf und kommen sich betrogen vor, weil man ihnen ja eine Maschine verkauft hat, die alles kann.

Dieses Konzept einer kannibalistischen Maschine, welche die allgemeinst vorstellbaren Denkvorgänge übernehmen kann, und die nach Verzehren des Bauplans ihrer Artgenossen deren Intelligenz erbt, ist genügend mit scheinbaren Paradoxien geladen, so dass man mit ihrer Hilfe auch die Unlösbarkeit des hilbertschen Entscheidungsproblems herleiten kann. Dies überrascht uns aus heutiger Perspektive kaum mehr. Auch Alan Turing war an seinen denkenden

Maschinen nicht nur als geistreiche theoretische Hilfskonstruktion im Beweis eines berühmten Theorems interessiert. In den weiteren Beiträgen im besprochenen Band schreibt er unter Titeln wie „Lös- bare und unlösbare Probleme“, „Intelligente Maschi- nen“, „Spielprogramme“, „Rechenmaschinen und Intelligenz“ immer wieder über Universelle Turing- maschinen und die Konsequenzen ihrer Existenz.

Uns ist das zentrale Konzept der Emulation der- massen alltäglich geworden, dass wir kaum mehr merken, wovon wir abstrahieren, wenn wir sagen, „mit dem Amiga kann man einen PC emulieren“, „Der Laserdrucker kann sowohl einen Epson als auch einen HP emulieren“, „Dieses Programm emu- liert die Difference Engine von Charles Babbage“ oder „Mit Sprint kann man die MS-Word-Oberflä- che emulieren“. Wir meinen, dass der symbolische, Intelligenzaspekt nachgeahmt wird und sehen von äusseren Unterschieden als völlig unerheblich ab. Selbst auf der Begleitdiskette zum hier besproche- nen Buch werden drei andere symbolverarbeitende Maschinen mitgeliefert, ohne dass die Tatsache auch nur erwähnt wird, dass es sich hier um Emulationen handelt.⁵ (Die Programme auf dieser Diskette würde man als professioneller Software-Hersteller gerne ausgereifter sehen. Bei gutwilliger Benutzung durch einen PC-Profi der ersten Stunde sollten nicht drei der vier Anwendungen beim ersten Versuch mit ir- gendwelchen "run time errors" abstürzen. Anderer- seits demonstrieren die Programme Turings Denken sehr schön und es wäre falsch, an eine „Gratisdisket- te“ aus dem Jahr 1987 dieselben Qualitätsmassstäbe anzulegen wie an ein Programmprodukt im Jahr 1990.)

Eine heute weitverbreitete feinsinnige Bemerkung über das Wesen der Computer lautet: digitale Ma- schinen könnten ja nur – in einem manichäischen Maschinen-Universum – zwischen Ja und Nein, zwi- schen Schwarz und Weiss unterscheiden. Die von solchen „Meinungen“ implizierte Überlegenheit des psychisch reifen Menschen, der auch Zwischentöne kennt, wird von Turing dadurch widerlegt, dass er zeigt, dass jegliche diskrete Maschine (mit endli- chem „Grauton-Alphabet“) mit einer Turingmaschi- ne emuliert werden kann und dass sogar Analogma- schinen von Turingmaschinen beliebig genau nach- geahmt werden können. Wie den Mathematikern zu Beginn dieses Jahrhunderts ist Turing und einigen Spezialisten klar, dass der Computer – trotz seinem

5 Auf der Diskette befinden sich neben dem Spielpro- gram „Nim“ Emulationen der der „Difference Engine“ von Charles Babbage, der deutschen Kryptologiemas- chine ENIGMA und eine Emulation der Universellen Turingmaschine.

Namen – ebenso wie die Mathematik, keine Rechen- maschine ist, sondern eine Maschine zur freien Ma- nipulation von Symbolen.

In den weiteren Artikeln der Sammlung verfolgt Turing lange vor der allgemeinen Verbreitung sol- cher Maschinen die häretische Frage, warum seine Turingmaschine nicht auch das menschliche Denken emu- lieren können sollte. In "Rechenmaschinen und Intelligenz" beschäftigt ihn die Frage, inwiefern man solchen Maschinen das Denken absprechen könne. Er zählt Einwände auf, die er argumentativ zu wider- legen versucht und unterstellt den entschiedensten Gegnern der Denkmaschinen, nämlich gewissen In- tellektuellen, eine aus ökonomischer Konkur- renzangst verzerrte, unredliche Argumentationswei- se.

Einwände gegen die Intelligenz der Computer ge- mäss Turings Klassifikation:

1. Der theologische Einwand: nur die un- sterbliche Seele kann denken und Gott gab diese nur den Menschen.
2. Der „Vogel-Strauss“-Einwand: die Fol- gen wären zu schrecklich.
3. Der mathematische Einwand: Turingma- schinen können die Lösbarkeit gewisser Gleichungen nicht entscheiden.
4. Der Bewusstseinseinwand: Maschinen können nicht fühlen und nicht kreativ tätig sein.
5. Verschiedene Unfähigkeiten: Maschinen können (noch) nicht Initiative besitzen, Sinn für Humor haben, Recht von Unrecht unter- scheiden, Fehler machen, sich verlieben, gerne Erdbeeren mit Schlagsahne essen....
6. Der Einwand von Lady Lovelace: Com- puter können immer nur das tun, was ihnen befohlen wird.
7. Der Mensch ist nicht digital, sondern etwa ein Hybridrechner.
8. Das Argument aus dem informellen Cha- rakter des menschlichen Verhaltens.

Turings Bemerkung, dass es nicht ausgeschlossen sei, dass Denkmaschinen einmal die Macht überneh- men könnten, erinnert uns daran, dass „Aemulatio“ auf lateinisch nicht nur „Nachahmung“ sondern auch „Wettstreit“ heisst.

Die deutsche Übersetzung von Turings Schriften liest sich flüssig. Die anglophile Übernahme des Plurals „Terms“ im Deutschen sowohl für „Terme“ in mathematischen Formeln, für „Termini“ im Sinne von „Fachausdrücke“ als auch in der wörtlichen Übertragung des englischen „in terms of“ zur Umgehung der Suche nach deutschen Formulierungen, mag einigen Lesern als neudeutsche Bereicherung der Sprache gefallen, dem Rezensenten stiess sie nur unangenehm auf.

Im Nachwort stellen uns die Herausgeber Alan Turing als Mitarbeiter des „Intelligence Service“ vor, dessen Arbeit entscheidend war, für die Entschlüsselung der deutschen Kryptologiemaschine ENIGMA. Dieser Beitrag des ehemaligen Pazifisten Alan Turing hat sicher den zweiten Weltkrieg mitentschieden. Im kalten Krieg der sich in England auch gegen Homosexuelle wie Alan Turing richtete, vergass das Vaterland seine Leistungen reichlich schnell und verurteilte ihn 1952 zur Hormonkur. Wenig später nahm er sich das Leben. Aus diesem Material konstruiert das reichlich postmoderne Nachwort die Geburt des Computers aus dem Geiste des totalen Kriegs, der Geheimdienste, des Intelligence Service.

Ich hoffe, bei den Lesern dieser Besprechung, Neugier geweckt zu haben und den Wunsch, sich Turings Argumentationen im Urtext zu Gemüte zu führen.

Sein Beitrag zum Verständnis der heutigen Computergesellschaft ist auch heute noch gewichtig und viel zu wenig bekannt. Die Herausgeber haben mit "Intelligence Service" der Intelligenz im deutschsprachigen Raum einen Dienst erwiesen haben.

26.3.90