

1 Was ist Informatik?

Der Begriff „Informatik“ wurde bereits Ende der 50er Jahre von dem deutschen Professor Dr. Karl Steinbuch geprägt. Den Begriff „Informatik“ kann man sich entweder zusammengesetzt aus den Wörtern „Information“ und „Automatik“ denken – will man die Tatsache betonen, dass es in der Informatik um die automatische Verarbeitung von Informationen geht. Will man dagegen die Ähnlichkeit der Informatik und der Mathematik betonen, so kann man sich den Begriff „Informatik“ zusammengesetzt aus den Wörtern „Information“ und „Mathematik“ denken.

Während man die Mathematik als eine Wissenschaft des „formal¹ Denkbaren“ auffassen kann, beschäftigt sich die Informatik mit dem „formal Realisierbaren“, also mit Berechnungen, die maschinell ausgeführt werden können. Ihre Fragestellungen, etwa zur Berechenbarkeit oder zur künstlichen Intelligenz, reichen bis in die Philosophie hinein. Kleines Beispiel: Ist die folgende Funktion (mit Hilfe einer automatischen Rechenmaschine, also einem Computer) berechenbar?

$$f(x) = \begin{cases} 1, & \text{falls } \pi\text{'s Nachkommadarstellung } \underbrace{7 \dots 7}_{x\text{-mal}} \text{ enthält} \\ 0, & \text{andernfalls} \end{cases}$$

Aufgabe 1.1

Überlegen Sie sich, wie ein Computer-Programm – unabhängig von einer bestimmten Programmiersprache – aussehen könnte, das diese Funktion f versucht zu berechnen. Was spricht dagegen, die Funktion f als „berechenbar“ zu bezeichnen?

Ausgewählte Pioniere der Informatik

Die folgende Liste der „Pioniere der Informatik“ ist subjektiv und könnte, von einer anderen Person erstellt, anders aussehen. Diese Liste enthält wichtige Persönlichkeiten aus den Bereichen der Theoretischen (Gödel, Turing, Chomsky, Knuth, Cook) und Praktischen Informatik (Richie, Thompson, Torvalds, Stallman, Peyton-Jones).

Vieles von dem, was auf den folgenden Seiten beschrieben wird, ist für den Informatik-Anfänger nicht einfach zu verstehen, und es wird auch nicht erwartet, dass Sie alles Beschriebene im Detail genau verstehen – es ist eher dafür gedacht,

¹ „Formal“ bedeutet hier, dass man es mit Konstrukten zu tun hat, deren Form gemäß eindeutigen Regeln klar festgelegt ist.

ihr Interesse für die Informatik und die theoretischen Grundlagen der Informatik zu wecken, und auch um Sie in die Lage zu versetzen, zumindest von den „großen“ Problemen der Informatik „schon einmal gehört“ zu haben.

Kurt Gödel

1906 – 1978



Gödel beschäftigte sich unter anderem mit maschinell (d. h. durch eine automatische Rechenmaschine wie ein Computer) ableitbaren Aussagen. Von ihm stammt der „Gödelsche Unvollständigkeitssatz“. Mit ihm beweist Gödel, dass es sogar in dem einfachsten Teilbereich der Mathematik, der Zahlentheorie (also der Theorie ganzer Zahlen), wahre Aussagen geben muss, die man nicht automatisiert ableiten bzw. beweisen kann. Dies versetzte der Mathematik der damaligen Zeit, den 30er Jahren des 20. Jahrhunderts, einen herben Schlag. Bis dahin war man fest davon überzeugt, die Mathematik widerspruchsfrei *formalisieren* zu können, d. h. in eine Form zu bringen, die auch eine Maschine „verstehen“ könnte und mit der eine Maschine so umgehen könnte, dass sie aus den formalisierten Grundvoraussetzungen (oft auch *Axiome* genannt) alle wahren Aussagen der Mathematik automatisch ableiten könnte.

Alan Turing

1912 – 1954



Der junge Turing interessierte sich für das sogenannte „Entscheidungsproblem“ (auch im englischsprachigen Raum so genannt), eines der „Hilbertschen Probleme“². Das Entscheidungsproblem lässt sich folgendermaßen formulieren: Gibt es, zumindest prinzipiell, eine mechanisch anwendbare Methode, die entscheiden kann, ob eine gegebene mathematische Behauptung beweisbar ist?

Um dieses Problem zu lösen, versuchte/hom Turing zunächst formal zu definieren, was unter einer „mechanisch anwendbaren Methode“ zu verstehen sei. Er drückte seine Definition in Form einer theoretischen Maschine aus, die bestimmte, wohl definierte Operationen auf einem Streifen Papier durchführen konnte. Diese „Maschine“, mit der Turing definierte, was unter „berechenbar“ zu verstehen sei, nennt man die *Turingmaschine*. Mit Hilfe dieses formalen Konzeptes einer Rechenmaschine konnte Turing beweisen, dass das Entscheidungsproblem unlösbar ist.

² Wichtige ungelöste mathematische Probleme, formuliert von dem deutschen Mathematiker David Hilbert, im Jahre 1900, einige davon wurden erst kürzlich gelöst, einige sind immer noch ungelöst.

Es stellte sich im Weiteren heraus, dass die Turingmaschine tatsächlich exakt die Fähigkeiten jedes Computers beschreibt: Eine Turingmaschine ist prinzipiell nicht mehr oder weniger leistungsfähig als jeder heutige Computer (ausgenommen: Quantencomputer). Mit Hilfe des einfachen mathematischen Modells der Turingmaschine lassen sich viele grundsätzliche Dinge über Berechenbarkeit zeigen.

Außerdem war Turing maßgeblich am Entschlüsseln geheimer Nachrichten der Nazis beteiligt.

Noam Chomsky

*1928



Noam Chomsky formalisiert die Darstellung und den Aufbau natürlicher Sprachen mit formalen „Grammatiken“. Diese Grammatiken definierten rekursiv den Aufbau und das Verhältnis der einzelnen Sprachelemente zueinander. Er entwickelte die sog. *Chomsky-Hierarchie*, in der er Klassen von Sprachen nach ihrer Komplexität „sortierte“: Typ-3-Sprachen sind die einfachsten; Typ-0-Sprachen die komplexesten. Um eine Typ-0-Sprache zu bauen bzw. zu erkennen, benötigt man den vollen Funktionsumfang einer Turing-Maschine. Um Typ-1-Sprachen zu erkennen genügen einfachere „Maschinen“, usw.

Besonders im Compilerbau und in vielen Bereichen der Theoretischen Informatik sind Chomskys Arbeiten auch heute noch sehr wichtig.

Donald E. Knuth

*1938



Donald E. Knuth ist Autor der mehrbändigen Buchreihe „The Art of Computer Programming“ und der Begründer der formalen Analyse und Laufzeitanalyse von Algorithmen. Er war der erste, der die Groß-Oh-Notation systematisch zur Beschreibung des Laufzeitverhaltens von Algorithmen verwendete. Unter anderem hat er den „Knuth-Morris-Pratt“-Algorithmus entwickelt, der im Vergleich zu älteren Ansätzen eine deutlich schnellere String-Matching³-Methode bereitstellte (d. h. eine Methode, um alle Vorkommen eines bestimmten Strings in einem Text zu suchen).

Außerdem hat Knuth das Textsatzsystem \TeX entwickelt (mit dem auch dieses Buch geschrieben wurde) inklusive der zugehörigen Zeichendefinitionssprache Metafont.

³ Der Begriff „Matching“ steht für „Abgleich“ oder auch Paarung und ist ein in der Informatik bzw. Algorithmik häufig gebrauchter Begriff, der das Finden einer Übereinstimmung oder das Finden passender Paare oder ähnliches bezeichnet.

Stephen A. Cook

*1939



Stephen A. Cook formulierte und erforschte das für die Theoretische Informatik grundlegende und bis heute ungelöste P/NP -Problem. Es gilt als eines der wichtigsten ungelösten Probleme der Mathematik und Informatik. In der Komplexitätstheorie, einem Teilgebiet der Theoretischen Informatik, geht es darum, „Probleme“⁴ anhand des Zeitaufwands zu klassifizieren, der benötigt wird, um sie zu lösen. Die Anzahl der zur Lösung maximal benötigten Berechnungsschritte wird auch als *Zeitkomplexität* bezeichnet.

Die Komplexitätsklasse P bezeichnet die Menge von Problemen, deren Laufzeit auf einer Turingmaschine sich im Verhältnis zur Größe n der Eingabe durch ein Polynom n^k beschreiben lässt: Das Sortierproblem, beispielsweise, liegt in P , da man Algorithmen angeben kann, die eine Liste mit n Elementen (die Größe des Problems) in n^2 Schritten sortieren kann⁵. Die Probleme in P werden auch oft als die „praktisch lösbaren“ Probleme bezeichnet. Die Komplexitätsklasse NP bezeichnet die Menge von Problemen, die sich mit einer nicht-deterministischen Turing-Maschine in n^k -Schritten lösen lassen. Anders ausgedrückt: In NP befinden sich diejenigen Probleme, bei denen man eine Lösung in polynomieller Zeit auf deren Richtigkeit überprüfen kann.

Das P/NP -Problem ist nun die Frage, ob die beiden Komplexitätsklassen identisch sind, d. h. ob es Probleme gibt, die sich in NP befinden, aber nicht in P . Man vermutet, dass $P \neq NP$ gilt.

Denis Richie

1941 – 2011

Ken Thompson

*1943



Denis Richie und Ken Thompson entwickelten um 1970 das Betriebssystem Unix. Ein Betriebssystem ist die Software eines Rechners, die direkt mit der Hardware kommuniziert. Unix verwendete zu dieser Zeit innovative Konzepte, wie ein hierarchisch organisiertes Dateisystem – strukturiert in Verzeichnissen, die selbst wiederum Verzeichnisse enthalten können – und die Verwendung des Dateisystems als Schnittstelle auf externe Geräte und Prozesse. So werden in Unix CD-Laufwerke, weitere Festplatten, USB-Sticks usw. als spezielle Dateien im Dateisystem abgebildet.

⁴ In der Theoretischen Informatik bezeichnet man als „Problem“ oft eine bestimmte zu lösende Aufgabe, wie beispielsweise das Sortieren einer Liste oder das Suchen eines bestimmten Textfragments in einem Buch.

⁵ Es gibt schnellere Algorithmen, die dies in $n \cdot \log n$ Schritten schaffen – etwa Quicksort.

**Richard
Stallman**
*1953



Richard Stallman ist der Gründer der Bewegung für freie Software. Er startete 1983 das sogenannte GNU-Projekt. GNU ist eine rekursive⁶ Abkürzung und steht für „GNU is not UNIX“. Dem GNU-Projekt ist unter anderem zu verdanken, dass Linux heute so erfolgreich ist. Das GNU-Projekt initiierte außerdem die „freie Software“-Bewegung. Hintergrund des GNU-Projekts war, dass seit etwa Ende der 70er Jahr Softwarefirmen begannen, Software nicht mehr in Form von Quelltexten auszuliefern, sondern nur noch in Form eines rein maschinenlesbaren Formats, auch „Binärformat“ genannt. Zudem wurde sie nur zusammen mit speziellen Lizenzen vertrieben, die dem Anwender verboten, die Software zu verändern und weiterzuverteilen. Für Stallman war dieser Verlust der Kontrolle an der eingesetzten Software ein Verlust an Freiheit.

Als Reaktion schuf er eine eigene Lizenz, die sogenannte Copyleft-Lizenz⁷ auch GPL (= GNU General Public Licence) genannt. GPL garantiert dem Nutzer weitgehende Rechte an der Software, und stellt gleichzeitig sicher, dass für alle Veränderungen, die an der Software vorgenommen werden, die gleiche Lizenz gilt.

Im Rahmen des GNU-Projektes schuf Stallman die erste Version des emacs-Editors (eines komplexen, programmierbaren Texteditors), des GNU-Debuggers `gdb` und des ersten freien, plattformübergreifenden C-Compilers⁸, auch heute noch unter dem Programmnamen `gcc` bekannt.

⁶ Rekursiv bedeutet: definiert mit Hilfe von sich selbst

⁷ Gemeint als Gegenteil von „Copyright“.

⁸ Ein Compiler, ein Übersetzer, ist ein Computerprogramm, das den Code einer höheren Programmiersprache in Maschinencode übersetzt.

**Tim
Berners-Lee**
*1955



Tim Berners-Lee, Erfinder der HTML (Hypertext Markup Language) und Vordenker des World-Wide-Web und des Semantic Web, arbeitet Ende der 80er Jahr am CERN⁹. Ein Teil der Laboratorien des CERN befinden sich auf französischem, ein anderer Teil auf schweizer Gebiet. In beiden herrschte eine unterschiedliche Netzwerkinfrastruktur. Berners-Lee schlug dem CERN ein Projekt vor, das auf dem Prinzip des Hypertexts¹⁰ beruhte und Datenaustausch und -aktualisierung auf beiden Seiten erleichtern sollte.

Er verwirklichte dieses Projekt und baute sowohl den ersten Web-Server als auch den ersten Webbrowser.

Berners-Lees Vorstellung von der Zukunft des Web ist das „Semantische Web“.

Linus Torvalds
*1969



Im Jahre 1991 entwickelte der finnische Informatik-Student Linus Torvalds eine einfache Terminalemulation, die er zum externen Zugriff auf einen größeren Unix-Rechner brauchte; er schrieb diese Emulation sehr hardwarenah und unabhängig von einem bestimmten Betriebssystem – er wollte die Möglichkeiten seines neu gekauften (und damals sehr teuren) Rechners mit 80386-Prozessor voll ausschöpfen. Als Grundlage dienten ihm das Minix-System und der gcc (der GNU-C-Compiler). Irgendwann, so Torvalds in seinem Buch „Just For Fun“^[13], bemerkte er, dass er eigentlich ein einfaches Betriebssystem geschrieben hatte. Schließlich kündigte er im Usenet¹¹ in der Gruppe `comp.os.minix` mit dem inzwischen berühmt gewordenen Posting sein neues Betriebssystem an:

9 CERN ist eine Großforschungseinrichtung in der Nähe von Genf in der Schweiz. Das CERN betreibt physikalische Grundlagenforschung; bekannt ist es vor allem durch seinen Teilchenbeschleuniger.

10 Netzartige Darstellung von Text und Informationen.

11 Ein weltweites elektronisches Netzwerk, ebenso wie das WWW, das Diskussionsforen bereitstellt, an denen grundsätzlich jeder teilnehmen kann.

Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-) Linus (torvalds@kruuna.helsinki.fi) PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

Aufgabe 1.2

Beantworten Sie durch eine kleine Internet-Recherche folgende Fragen:

- (a) Gibt es heute noch das Minix-Betriebssystem?
- (b) In welchen Punkten unterscheidet sich Minix von Linux/Unix und Windows?

Aufgabe 1.3

1992 kam es durch einen Usenet-Artikel Andrew S. Tanenbaums in der Newsgroup `comp.os.minix` mit Titel „Linux is obsolete“ zu einer berühmten Debatte um die Struktur des Linux-Kernels, in dem der Informatikprofessor und Autor des Mikrokernel-Systems `Minix` Tanenbaum eine Reihe von Kritikpunkten an dem damals jungen Linux-Projekt anbrachte.

Sie können diese Diskussion leicht finden (einfach „Linux is Obsolete“ in eine Suchmaschine Ihrer Wahl eingeben). Lesen Sie die Diskussion und versuchen Sie sowohl Tanenbaums Kritikpunkte als auch Torvalds Gegendarstellungen (so gut wie möglich) zu verstehen.