

1

Was ist ein Computer?

Überblick

Dieses Kapitel beschäftigt sich mit dem Wesen des Computers und stellt eine Einführung in die automatisierte Informationsverarbeitung dar. Zunächst gehen wir der Frage nach, was einen Computer ausmacht und welche Ideen ihm zugrunde liegen. Dazu unternehmen wir einen Streifzug durch die Geschichte und werfen einen Blick auf die wichtigsten Innovationen, die zum modernen Computer geführt haben. Zudem lernen wir die Turing-Maschine kennen, eine skurrile, minimalistische Rechenmaschine, die zunächst nur in der Vorstellung existiert und trotzdem für die Informatik von zentraler Bedeutung ist; an ihr studieren wir die Grundprinzipien eines Computers konkret und anschaulich. Schließlich wechseln wir die Perspektive und untersuchen anhand einer Bakterie, wie in der Natur Informationen verarbeitet werden. Wir stellen die Bakterie der Turing-Maschine gegenüber und entdecken verblüffende Analogien. Dass es sich dabei nicht nur um eine philosophische Betrachtung handelt, sehen wir am Schluss: wenn sich nämlich Biologie und Informatik in der Realität treffen und es darum geht, mit DNA-Molekülen zu rechnen.

Die Kapitelüberschrift mag harmlos daherkommen, doch die Frage hat es in sich. Zwar gehen wir tagtäglich mit Computern um und haben insofern eine intuitive Vorstellung davon, was mit dem Begriff des Computers gemeint ist. Das Verständnis in Worte zu fassen und auf den Punkt zu bringen, ist hingegen gar nicht so einfach – fällt Ihnen spontan eine eingängige Formulierung ein? Wie auch immer: Will man sich mit der Informatik beschäftigen, kommt man um eine genaue Antwort nicht herum, denn um den Computer dreht sich dabei ja alles. Wie man ihn baut, wie man ihn einsetzt und was man generell mit ihm bewerkstelligen kann. Nicht ohne

Grund spricht man im Englischen von *Computer Science*, also von einer Wissenschaft, deren Untersuchungsgegenstand der Computer ist. Was also ist ein Computer?

Maschinen zum Rechnen

Ein Auto ist ein Ding, das fährt, ein Flugzeug ist ein Ding, das fliegt, und ein Computer ist ein Ding, das rechnet – so oder so ähnlich könnte es in einem Lexikon für Kinder stehen. Wollen wir es genauer wissen, so lassen sich unterschiedlichste Definitionen finden. Egal ob kurz oder lang, im Kern sagen sie alle das Gleiche, nur leider verhält es sich mit ihnen ungefähr so wie mit einer Trockensuppe: Ohne Erläuterung bzw. Flüssigkeit sind sie ausgesprochen staubig. Leichter lässt sich das Wesen eines Computers anhand konkreter Beispiele veranschaulichen, insbesondere an Beispielen, die Meilensteine in seiner Entwicklungsgeschichte markieren. Wir werden also im Folgenden einige Vorläufer des Computers kennenlernen und drei fundamentale Prinzipien herauschälen. Doch beginnen wir zunächst mit den Bildern, die wir heute mit dem Wort Computer verbinden.

Die Erscheinungsbilder des Computers sind vielfältig

Wenn ich Ihnen sagen würde, dass in dem in Abb. 1.1 dargestellten Karton – über seine Größe sei nichts ausgesagt – ein Computer steckt, wie würden Sie sich dann den Inhalt vorstellen?

Sicherlich kommt Ihnen ein Bildschirm in den Sinn und ein entsprechendes Gehäuse, das über ein oder mehrere seitlich oder rückseitig angebrachte

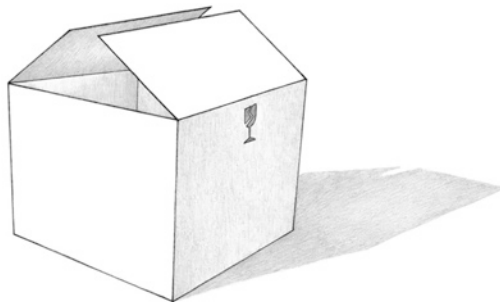


Abb. 1.1 Der Computer im Karton. (© Susanne Staubli, Eckart Zitzler)

Buchsen verfügt. Vielleicht ist noch eine Tastatur vorhanden, die entweder integriert oder als separate Komponente angeschlossen ist? Oder schwebt Ihnen ein Hochleistungscomputer vor, eine dieser riesigen Rechenanlagen, die den Platz eines einzelnen Gehäuses bei Weitem sprengt und stattdessen in unzähligen, mit aufwändigen Kühlsystemen ausgestatteten Schränken untergebracht ist, die wiederum ganze Hallen füllen? Diese Erscheinungsbilder haben sich im Laufe der Zeit herausgebildet und repräsentieren verschiedene Anwendungszwecke, siehe Anm. 1.1. Natürlich decken sie nur einen Teil eines Spektrums an Varianten ab, die sich hinsichtlich ihrer Leistungsfähigkeit, ihres Stromverbrauchs, ihrer Handlichkeit, ihrer Robustheit, ihrer Kosten usw. unterscheiden. Sie alle verkörpern die klassische Sicht auf den Computer: ein eigenständiger Apparat, mit dem Berechnungen oder ganz allgemein Programme ausgeführt werden können.

Anmerkung 1.1: Spielarten von Computern

Abb. 1.2 zeigt eine Auswahl historischer Erscheinungsformen von Computern. Der klassische Desktop-Computer (unten links), bestehend aus einem zentralen Gehäuse sowie weiteren, häufig separaten Komponenten wie Bildschirm und Tastatur, trat bereits in der 1970er-Jahren in Erscheinung. Gut 30 Jahre später sind tragbare Geräte sehr verbreitet: Laptops (oben Mitte), die die Komponenten eines Desktop-Computers in einem kompakten, leichten Gerät vereinen, sowie Tablet-Computer (oben links), die über ein flaches Gehäuse mit berührungsempfindlichem Bildschirm verfügen und per Fingerbewegungen gesteuert werden. Supercomputer (rechts) sind auf das Hochleistungsrechnen ausgerichtet und nur in großen Rechenzentren anzutreffen.

Die andere Sicht ist die des eingebetteten Computers, der in andere Gerätschaften oder Anlagen eingebunden ist und dort als Teil eines Ganzen weitestgehend unsichtbar seine Dienste verrichtet; ein Paradebeispiel hierfür ist die Raumfahrt (Abb. 1.3). Solche auf einen spezifischen technischen Kontext zugeschnittene Computertechnologie finden wir überall in unserem Alltag, z. B. im Fotoapparat und in der Waschmaschine. Auch im Krankenhaus ist sie nicht mehr wegzudenken; die Bandbreite reicht vom Computertomografen bis zum Blutdruckcomputer. Letztlich treffen wir eingebettete Computer überall dort an, wo etwas zu steuern ist – beispielsweise im Auto, in der Gepäckförderanlage oder im Wasserkraftwerk. Computer sind also allgegenwärtig und vor allem vielfältig in ihrer Erscheinung.

Das war natürlich nicht immer so. Der Begriff des Computers unterliegt einem ständigen Wandel, gemäß dem technologischen Fortschritt. Während um 1950 noch riesige Konstruktionen mit diesem Wort assoziiert wurden und

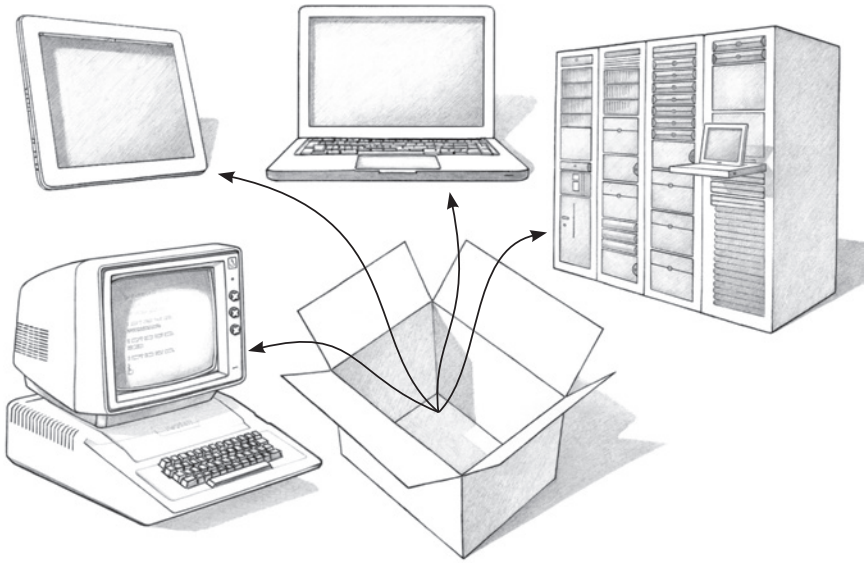


Abb. 1.2 Varianten von Computern. (© Susanne Staubli, Eckart Zitzler)

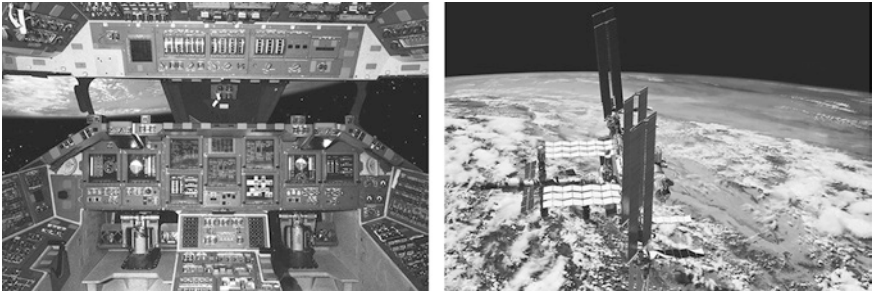


Abb. 1.3 Die computerbasierten Steuerungssysteme im Cockpit einer Raumfähre (links) und eine Raumstation (rechts), die von einer Vielzahl von Computern gesteuert wird. (© NASA)

die Vorstellung, dass einmal in jedem Haushalt ein persönlicher Computer stehen könne, absurd erschien, hatte sich dieses Bild 40 Jahre später grundlegend gewandelt. Nun verknüpfte man mit dem Begriff ein Gerät, das sich bequem zuhause auf dem Schreibtisch platzieren ließ: den Personal Computer (vgl. Abb. 1.2 links unten). Heutzutage ist die Bezeichnung aufgrund der Vielfalt und der rasanten Entwicklung viel verschwommener, und es ist anzunehmen, dass sich unsere Bilder in den kommenden Jahren wiederum stark verändern werden.

Die Grundideen sind hingegen immer noch dieselben. Ihre Wurzeln reichen weit in die menschliche Kulturgeschichte zurück – kaum verwunderlich, wenn wir das Wort Computer auf seine ursprüngliche Bedeutung zurückführen, nämlich Rechner von lateinisch *computare*, zu Deutsch berechnen/zusammenrechnen. Die Motivation war ursprünglich, immer ausgeklügeltere Hilfsmittel zum Rechnen zu ersinnen. Werfen wir also einen Blick auf die Anfänge und schauen uns an, wie sich die Computeridee stufenweise entwickelt hat.

Ein lebendes Fossil: der Abakus

Bereits vor schätzungsweise 3000 Jahren trat mit dem Abakus (Abb. 1.4) ein erstes Rechengerät in Erscheinung, das als Vorläufer des Rechnens mit Stift und Papier betrachtet werden kann – nicht von ungefähr, schließlich steht sein Name sinngemäß für „eine mit Sand oder Staub bedeckte Fläche zum Schreiben“. Entsprechend dient dieses altertümliche Rechenutensil dem Zweck, Zahlen festzuhalten und somit das Gehirn zu entlasten: Indem wir nicht alles im Kopf behalten müssen, sind wir ausdauernder und machen weniger Fehler beim Rechnen.

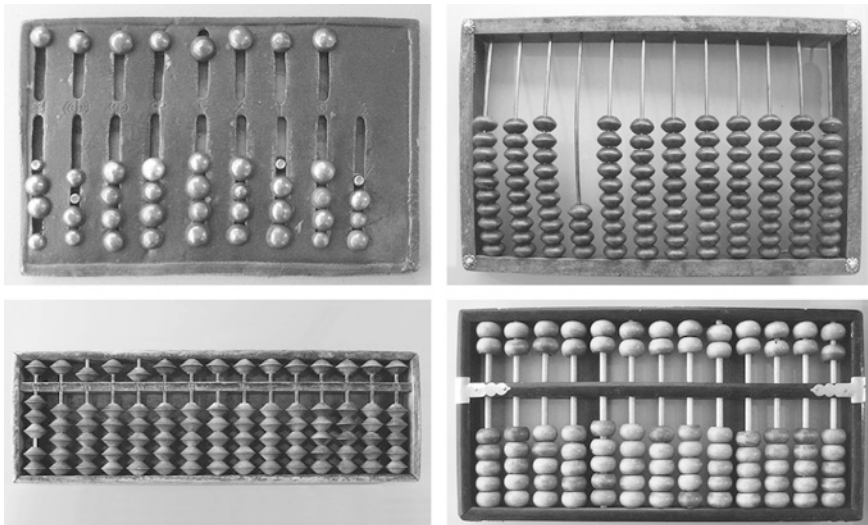


Abb. 1.4 Verschiedene Abakus-Varianten: ein römischer Handabakus (oben links), ein russischer Stschoty (oben rechts), ein japanischer Soroban (unten links) und ein chinesischer Suanpan (unten rechts). (© Arithmeum, Rheinische Friedrich-Wilhelms-Universität Bonn)

Auf dem Abakus wird eine Dezimalzahl, wie wir es gewohnt sind, als Folge von Ziffern dargestellt, und die Wertigkeit einer Ziffer wird über eine entsprechende Anzahl von Kugeln repräsentiert. Die Kugeln sind in der Regel aufgereiht an Stäben aus Metall oder Holz, die parallel zueinander in einen Holzrahmen eingespannt sind. Jeder Stab fungiert als Platzhalter für eine Ziffer, und indem wir die Kugeln auf dem Stab nach unten oder oben schieben, können wir beliebige Werte zwischen 0 und 9 einstellen. Die Gesamtzahl ergibt sich, indem wir die codierten Ziffern von den Stäben ablesen und aneinanderfügen; Abb. 1.5 oben visualisiert das Prinzip für einen polnischen Liczydło, die Leserichtung ist hier wie gewohnt von rechts nach links. Die größte Zahl, die der Abakus aufnehmen kann, ist demnach durch die Anzahl der Stäbe bestimmt: Bei zehn Stellen z. B. liegt die Darstellungsgrenze bei 9.999.999.999.

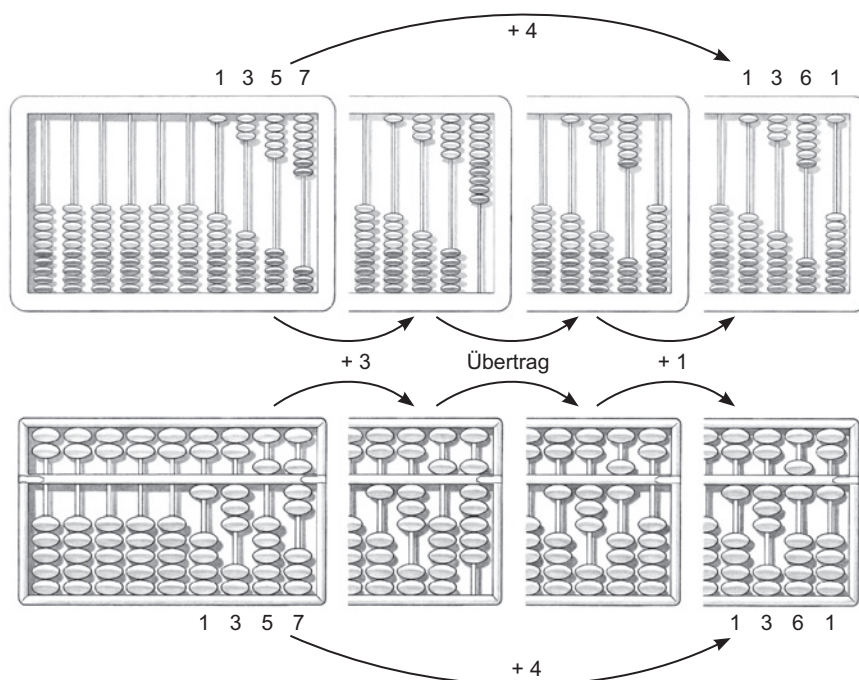


Abb. 1.5 Illustration, wie mit zwei Abakus-Varianten – einem polnischen Liczydło (oben) und einem chinesischen Suanpan (unten) – die Zahl 1357 dargestellt und anschließend die Zahl 4 dazuaddiert werden kann. (© Susanne Staubli, Eckart Zitzler)

Die genaue Ausgestaltung eines Abakus kann variieren, nicht nur hinsichtlich der verwendeten Materialien, sondern auch bezüglich der Bauart. Bei einem polnischen Abakus (Abb. 1.5 oben) z. B. besitzen die jeweils unteren fünf Kugeln eine andere Farbe als die oberen fünf Kugeln – zur besseren Übersichtlichkeit. Aus dem gleichen Grund ist bei einem chinesischen Abakus (Abb. 1.5 unten) eine Querleiste in den Rahmen eingebaut, der die Kugeln auf den Stäben in zwei Gruppen unterteilt. Die Kugeln der unteren Hälfte weisen wie beim polnischen Abakus eine Wertigkeit von eins auf, aber die oberen Kugeln zählen fünf. Summiert man die Wertigkeiten der Kugeln, die Richtung Mittelleiste geschoben sind, auf, dann kommt man auf die repräsentierte Ziffer. Abgesehen von diesen länderspezifischen Spielarten ist die Funktionsweise jedoch immer dieselbe.

Des Weiteren erlaubt der Abakus, grundlegende Operationen wie Addieren und Subtrahieren effizient und auf einfache Art und Weise auszuführen – die Vereinfachung wird erzielt, indem der Rechengvorgang in kleinere, schnell durchzuführende Einzelschritte zerlegt wird. Wir kennen diese Vorgehensweise vom Rechnen mit Papier und Stift: Beispielsweise werden zwei aus mehreren Ziffern bestehende Zahlen addiert, indem die Einzelziffern beider Zahlen positionsweise, von rechts nach links, aufsummiert und mögliche Überträge bei der nächsten Position dazugezählt werden. Das gleiche Schema findet beim Abakus Anwendung, wie Bsp. 1.1 zeigt. Dabei dient der Abakus genauso wie das Papier als Gedächtnisstütze, mit der sich nach jedem Einzelschritt das aktuelle Zwischenergebnis festhalten lässt; nach Durchführung aller Einzelschritte zeigt er das Endergebnis an.

Beispiel 1.1: Addieren mit dem Abakus

Nehmen wir an, wir wollten zur Zahl 1357 die Zahl 4 addieren (vgl. Abb. 1.5). Der Additionsvorgang beginnt bei der kleinstwertigen Position, also bei der Ziffer 7 bzw. dem Stab rechts außen. Zuerst versuchen wir dort vier Kugeln nach oben bzw. Richtung Mittelleiste zu verschieben. Da jedoch nur noch drei einwertige Kugeln zur Verfügung stehen, können wir an dieser Position nur drei dazuzählen. Jetzt repräsentiert der rechte Stab die Zahl 10, also einen Übertrag. Wir lösen den Übertrag auf, indem wir eine Kugel auf dem nächsten Stab links nach oben bzw. zur Mitte schieben und auf dem Stab rechts außen alle Kugeln in ihre Ausgangsstellung (die Ziffer 0 repräsentierend) zurückversetzen. Danach können wir den Restbetrag hinzufügen: Drei haben wir bereits addiert, es sollten aber vier sein; deswegen zählen wir noch eins zu rechten Position dazu, indem wir dort eine Kugel nach oben bzw. Richtung Mittelleiste verschieben.

Mit seiner Speicherfunktion besitzen der Abakus, der übrigens hier und da auch heute noch im Einsatz ist, und verwandte Rechenhilfen wie der Rechenschieber ein fundamentales Merkmal eines Computers. Allerdings sind sie nicht in der Lage, die Speicherinhalte selbstständig zu verändern, also automatisch ganze Berechnungen mit mehreren Rechenschritten durchzuführen. Die Einführung der Automatik – dieser Begriff ist ja in der zweiten Hälfte des Worts „Informatik“ enthalten – erfolgte erst später.

Wenn es beim Rechnen knackt, knirscht und surrt

Die ersten richtigen Rechenmaschinen wurden im 17. Jahrhundert entwickelt. Ihnen konnte man regelrecht beim Rechnen zuhören, denn ihr Inneres ähnelte einem mechanischen Uhrwerk. Ein in der Tat zutreffender Vergleich, schließlich handelt es sich bei einer Uhr um nichts anderes als um eine Zählmaschine: Jede Sekunde zählt sie eins zur aktuellen, gespeicherten Zeit hinzu, die sich selbst aus drei Zahlen – repräsentiert durch Sekunden-, Minuten- und Stundenzähler – zusammensetzt. Die drei Zähler sind miteinander gekoppelt, d. h., nach 60 s erhöht sich der Minutenzähler um eins und nach 60 min der Stundenzähler, während gleichzeitig der Sekunden- bzw. der Minutenzähler zurückgesetzt wird. Eine Uhr beherrscht demnach genau eine Rechenoperation, die Addition von eins. Das Gleiche gilt für einen Handzähler, ein per Tastendruck gesteuertes Zählwerk, mit dem sich Personen, Runden, Versuche usw. zählen lassen. Abgesehen vom verwendeten Zahlensystem besteht der einzige Unterschied zwischen den beiden Geräten darin, dass die Rechenoperation bei letzterem manuell über eine Taste ausgelöst wird, während ersteres autonom und zeitlich äußerst präzisiert hochzählt (Abb. 1.6).

Die Grundidee, Rechenschritte auf mechanische Vorgänge abzubilden, lässt sich am leichtesten am Beispiel derjenigen Grundrechenart erläutern und nachvollziehen, die wir als erste in der Schule erlernen: der Addition. Ein Taschenrechner aus dem 19. Jahrhundert, die Addiermaschine des Amerikaners Archibald M. Stephenson, soll uns als Anschauungsobjekt dienen; er war darauf ausgerichtet, seinen Benutzer beim stupiden Aufsummieren von Zahlenkolonnen zu entlasten und somit die Fehlerquote zu reduzieren.

Wie in Abb. 1.7 ersichtlich, besitzt diese Metallkonstruktion zwei mit Zacken versehene Drehscheiben, auf denen Zahlen eingraviert sind: rechts die Ziffern 0 bis 9, links die Werte 0 bis 19. Analog dem Abakus steht die rechte Scheibe für die Einer und die linke Scheibe für die Zehner; die größte

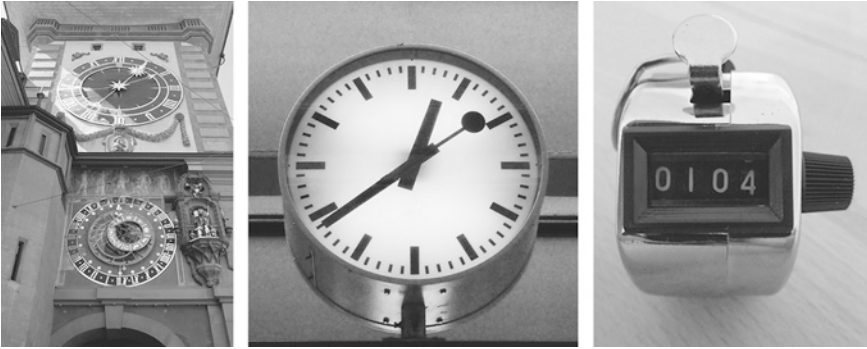


Abb. 1.6 Relikte aus der Zeit des mechanischen Rechnens: Turmuhr und Astrolabium der Zytglogge in Bern (links), eine Bahnhofsuhr (Mitte) und ein Handzähler mit Zähltaste und seitlichem Rückstellknopf (rechts). (© Eckart Zitzler)



Abb. 1.7 Mechanischer Taschenrechner von A. M. Stephenson, links mit montierter und rechts ohne Deckplatte. (© Reinhard Atzbach)

darstellbare Zahl ist demnach die 199. Eine Deckplatte mit zwei Kerben stellt sicher, dass immer nur die jeweils oberste Zahl einer jeden Scheibe sichtbar ist. Darüber hinaus sind Löcher in die Scheiben eingestanz, die unmittelbar bei den Zahlen positioniert sind (links nur bei der 0). Wie funktioniert nun dieser Rechner?

Bsp. 1.2 gibt Ihnen einen Einblick in den zugrunde liegenden Mechanismus. Einfach, ja fast schon trivial ist er; die Hauptschwierigkeit liegt in der Handhabung des Übertrags, also beim Zusammenspiel der Scheiben. Vielleicht sind Sie jetzt etwas enttäuscht: Das soll ein mechanischer Rechner sein? Im Wesentlichen ja, doch seien Sie beruhigt: Die Realisierung anderer Rechenoperationen ist ungleich komplizierter und nicht in ein paar Worten erklärt. Dennoch, auch moderne Rechner bauen auf simplen Grundprinzipien auf und solche verkörpern ja eine mechanische Uhr, ein Handzähler und Stephenson's Addiermaschine.

Beispiel 1.2: Funktionsweise des Stephenson-Addierers

Nehmen wir an, wir wollten die Summe aus 3, 5 und 2 berechnen. Als Erstes richten wir die Drehscheiben so aus, dass in beiden Kerben die 0 zu sehen ist – der Speicher ist somit zurückgesetzt und das Ergebnis einer eventuellen vorherigen Berechnung gelöscht. Anschließend bewegen wir die rechte Scheibe um drei Positionen, d. h., wir geben die 3 ein. Dazu können wir einen Stift nehmen, ihn in das Loch stecken, das mit der 3 auf der Deckplatte korrespondiert, und auf diese Weise die Scheibe im Uhrzeigersinn so weit drehen, bis sich der Stift direkt unterhalb der Kerbe befindet. Das Ganze wiederholen wir mit der 5: Stift in das Loch bei der 5 stecken, weiterdrehen bis zur Kerbe, und schon ist die erste Addition vollzogen. In der rechten Kerbe ist als Zwischenresultat die 8 zu sehen, da die Scheibe insgesamt um acht Positionen bewegt wurde. Für die letzte Addition drehen wir die Scheibe erneut, diesmal um zwei weitere Positionen. Es kommt wieder die 0 zum Vorschein, obwohl wir an sich bei der 10 angelangt wären. Der entstandene Übertrag müsste auf die linke Drehscheibe übertragen werden, und genau das passiert auch. Die an der rechten Drehscheibe angebrachte Zacke greift in die Zacken der linken Drehscheibe ein und schiebt diese um eine Position weiter, sobald der Übergang von der 9 zur 0 vollzogen wird.

Die in Abb. 1.8 gezeigte Pascaline, eine der ersten mechanischen Rechenmaschinen überhaupt und die erste, von der mehrere Exemplare angefertigt wurden, arbeitete ebenfalls nach diesem Schema. Wie Stephenson's Addierer besaß sie mehrere nebeneinander angebrachte Metallwählscheiben, die über einen ausgeklügelten Mechanismus ineinandergriffen und so Überträge weiterreichen konnten. Die Ergebnisse waren allerdings nicht direkt von den Scheiben ablesbar, sondern wurden darüber in separaten Kästchen angezeigt. Um diesen Bildschirm-Vorläufer anzusteuern, bedurfte es einer entsprechenden Übersetzung mittels Zahnrädchen. Die Pascaline war hauptsächlich auf das Summieren ausgerichtet, doch über einen Trick konnte man auch subtrahieren. Wie Sie eventuell schon

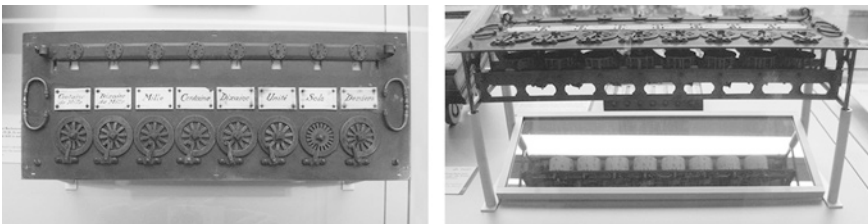


Abb. 1.8 Diese Fotografien einer Pascaline zeigen einmal die Oberseite (links) und einmal die Vorder- und Unterseite mit geöffnetem Bodendeckel (rechts). (© Arithmeum, Rheinische Friedrich-Wilhelms-Universität Bonn)

vermutet haben, war es Blaise Pascal, der französische Universalgelehrte, der diesen für damalige Verhältnisse wundersamen Kasten 1642 erfunden hat; 350 Jahre später wurde eine bekannte Programmiersprache nach ihm benannt.

Das war also die Einführung der Automatik. Der menschliche Benutzer musste zwar immer noch Kraft aufwenden, um die Zahnräder zu bewegen – in diesem Sinne funktionierten die Maschinen nicht vollautomatisch wie z. B. eine Schiebetür mit Bewegungssensor –, doch die einzelnen Denkschritte zwischendurch fielen weg und wurden durch mechanische Vorgänge ersetzt. Damit war das automatisierte Rechnen ins Leben gerufen und man kam dem modernen Computer einen gewaltigen Schritt näher. Doch die Programmierbarkeit fehlte noch, d. h. die Möglichkeit, die Berechnungsvorschrift zu variieren und flexibel einstellen zu können, welche Rechenschritte in welcher Reihenfolge automatisch ausgeführt werden sollen.

Die Geburt des modernen Computers

Wenn wir uns heutzutage eines Computers bedienen, ist es für uns selbstverständlich, ihn für verschiedenste Zwecke einzusetzen. Wir schreiben Texte, versenden Nachrichten, hören Musik, schauen Filme, bearbeiten Fotos usw. Wenn uns eine bestimmte Anwendung nicht gefällt, installieren wir das Produkt eines anderen Anbieters. Ist unsere Anwendung veraltet, so kann sie mittels eines Updates aktualisiert werden. Und das alles auf ein und demselben Gerät und ohne dieses verändern zu müssen. Weil unser Computer eben programmierbar ist und mit beliebigen Programmen gefüttert werden kann.

Die Grundlage für diese Flexibilität wurde im frühen 19. Jahrhundert gelegt, als Charles Babbage das Konzept für einen universellen, programmierbaren Rechner vorstellte und Ada Lovelace ein dazugehöriges Programm verfasste (Abb. 1.9). Babbage, ein britischer Mathematiker, begann erst im Alter von 30 Jahren, sich mit dem automatisierten Rechnen zu beschäftigen – dafür dann umso intensiver. Einen Großteil seines Lebens widmete er diesem Thema und tatsächlich gelang ihm 1834, als er sich die Analytical Engine ausdachte, etwas, das aus heutiger Sicht als großer Wurf und Pionierleistung angesehen werden muss – auch wenn diese Maschine nie Wirklichkeit wurde (Abb. 1.10). Man könnte sagen, Babbage war der Erste, der einen vollwertigen Computer beschrieb, und in der Tat spiegeln seine Pläne grundlegende Elemente eines modernen Computers wider: 1000 Zahlen hätte die Analytical Engine speichern können und das Rechenwerk – er nannte es „Mühle“ – wäre in der Lage gewesen, alle vier Grundrechenarten auszuführen; darüber hinaus wären weitere Aktionen

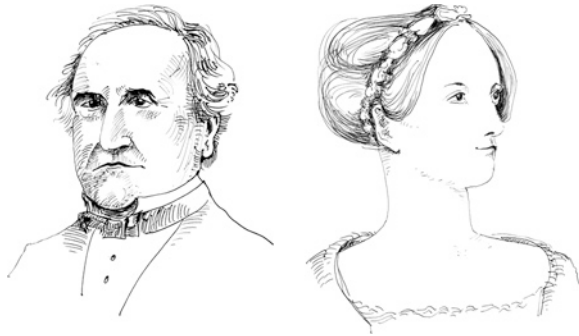


Abb. 1.9 Charles Babbage, der Erfinder der Analytical Engine, und Ada Lovelace, die das erste namhafte Programm für einen Computer entwickelte. (© Camille Büsser)

wie Vergleiche von Zahlen möglich gewesen. Gegenüber der Pascaline, die gerade mal eine Zahl speichern und Additionen sowie auf Umwegen auch Subtraktionen durchführen konnte, stellte das einen riesigen Fortschritt dar.

Wie die Idee der Programmierbarkeit effektiv in Babbages Maschine hätte umgesetzt werden sollen, ist hingegen etwas schwerer zu erklären; dafür ist seine Konstruktion zu komplex, schließlich hätte sie von einer Dampfmaschine bedient werden müssen – so viele Zahnräder hätten da ineinandergegriffen, dass ein Mensch den Kraftaufwand gar nicht hätte bewerkstelligen können. Eine grobe Vorstellung können Sie bekommen, wenn Sie an die Gangschaltung eines Fahrrads oder das Getriebe eines Autos denken: Hier ist die Übersetzung zwischen verschiedenen Zahnrädern flexibel einstellbar und wird beispielsweise über einen Schalthebel „programmiert“. Fakt ist auf jeden Fall, dass Ada Countess of Lovelace, eine 24 Jahre jüngere Zeitgenossin Babbages und Tochter des bekannten englischen Dichters Lord Byron, den Nutzen der Analytical Engine erkannte und 1842 einen schriftlichen Plan zur Berechnung von Bernoulli-Zahlen mithilfe dieser Maschine vorlegte. Aus diesem Grund wird sie als erste Programmiererin in der Geschichte angesehen.

Babbages und Lovelaces Ideen waren den technischen Möglichkeiten ihrer Zeit allerdings so weit voraus, dass es noch weitere 100 Jahre dauern sollte, bis in den 1940er-Jahren die ersten funktionstüchtigen Computer gebaut werden konnten; diese arbeiteten nicht mechanisch, sondern rechneten mittels Strom. Zuerst gelang das 1941 Konrad Zuse und Helmut Schreyer in Deutschland. Der erste rein elektronische Computer, der ENIAC (Abb. 1.11), wurde von J. Presper Eckert und John W. Mauchly

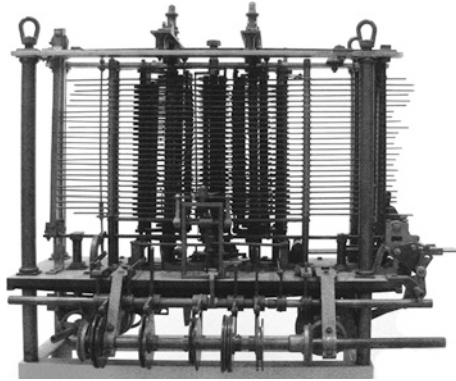


Abb. 1.10 Ein Versuchsmodell der Analytical Engine. (© Arithmeum, Rheinische Friedrich-Wilhelms-Universität Bonn)

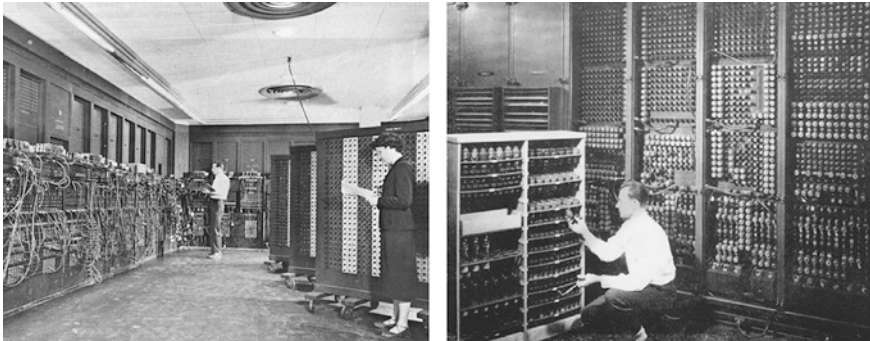


Abb. 1.11 Der ENIAC – die Abkürzung steht für *Electronic Numerical Integrator and Computer*. (© U. S. Army („U.S. Army Photo“))

in den USA entwickelt und schließlich 1946 der Öffentlichkeit vorgestellt. Diese Dinosaurier des Computerzeitalters waren riesig, schwer und gefräßig: Der ENIAC beispielsweise beanspruchte eine Grundfläche von 10×17 m, wog 27 t und hatte einen so hohen Stromverbrauch, dass Gerüchten zufolge die Lichter in Philadelphia flackerten, wenn der ENIAC eingeschaltet wurde. Der Durchbruch wurde eingeläutet mit der Erfindung des Transistors im Jahr 1947, einem kleinen, durch Strom steuerbaren Schalter, durch dessen Verwendung elektronische Rechner billiger, schneller, zuverlässiger und kleiner wurden. Es sollte noch eine Weile dauern, bis die Technologie ausgereift war, doch mit der Einführung des Mikroprozessors – des Kernstücks eines Computers – in den 1970er-Jahren nahm eine rasante Entwicklung ihren Lauf. Zunächst eroberte in den 1980er-Jahren der Personal Computer in verschiedensten Ausführungen Privathaushalte und

Büros, in den 1990ern folgte die weltweite Vernetzung durch das Internet und zu Beginn des 21. Jahrhunderts kamen kleine, mobile Computer-Endgeräte hinzu. Welchen gewaltigen Fortschritt all das bedeutet, sieht man heute.

Drei Wesensmerkmale eines Computers

Letztlich war es also immer wieder die Technologie – von der Zahnradmechanik bis zur Halbleitertechnik –, die der lang gereiften Computeridee zum Erfolg verhalf. Heutzutage ist für uns ein Computer ein elektronisches Gerät, das, wie wir gesehen haben, in verschiedensten Formen in Erscheinung tritt und dessen Anwendungsspektrum weit über das pure Rechnen hinausgeht. Doch das Konzept eines Computers ist nicht an eine bestimmte Technologie gebunden. Charles Babbages Pläne sahen eine mechanische Umsetzung vor, der erste funktionsfähige Computer – die Z3 von Konrad Zuse – arbeitete elektromechanisch, der ENIAC funktionierte rein elektronisch und nutzte vor allem Elektronenröhren, und heutige Computer basieren auf der Halbleitertechnologie. Selbst mit Wasser und DNA-Molekülen wurde schon gerechnet, und welche Rolle Quantencomputer – also Computer, die auf quantenmechanischen Prinzipien beruhen – spielen werden, wird die Zukunft zeigen.

Ob dieser Vielfalt an Computerausprägungen kann einem schwindlig werden. Umso mehr drängt sich die Ausgangsfrage wieder auf: Was ist eigentlich ein Computer, was macht sein Ureigenstes aus? Wie wir gesehen haben, lässt sie sich weder über das Erscheinungsbild noch über die Technologie beantworten. Doch drei Stichworte sind bereits gefallen, nämlich Speicher, Automatik und Programmierbarkeit. Rekapitulieren wir:

- Die Vorläufer der Rechenmaschinen sind geschickt konzipierte Gedächtnisstützen wie der Abakus, mit denen sich Berechnungen einfacher und zuverlässiger durchführen lassen; der Speicher, den diese Hilfsmittel verkörpern, wird durch Menschenhand bedient und verändert.
- Demgegenüber kann eine echte Rechenmaschine selbstständig Zwischenergebnisse produzieren und ablegen. Zum Speicher kommt ein z. B. mechanisches Rechenwerk hinzu, das die Speicherinhalte lesen und manipulieren kann. Das Rechenwerk automatisiert Berechnungsschritte, die sonst von Hand ausgeführt werden; die Schritte und ihre Abfolge sind dabei fest vorgegeben.

- Der moderne Computer stellt den dritten Meilenstein in dieser Entwicklung dar, indem er das Konzept der Programmierbarkeit realisiert. Bei ihm ist das Rechenwerk flexibel einstellbar, d. h., die auszuführenden Rechenoperationen und ihre Reihenfolge lassen sich – innerhalb eines vorgegebenen Rahmens – frei bestimmen. Damit ist ein Computer nicht auf spezielle Berechnungen festgelegt, sondern universell einsetzbar.

Somit haben wir die zentralen Prinzipien identifiziert, auch wenn eine fassbare Definition noch aussteht. Die wird im Folgenden nachgeliefert, allerdings nehmen wir dabei einen leicht veränderten Blickwinkel ein, indem wir unsere Ausgangsfrage umformulieren: Was bliebe übrig, wenn wir einen Computer auf sein Minimum reduzieren wollten, ohne das Charakteristische seines Wesens zu verlieren?

Des Computers Kern: die Turing-Maschine

Mit dieser Frage hat sich der Brite Alan Turing (Abb. 1.12) bereits beschäftigt, bevor die ersten funktionsfähigen Computer überhaupt gebaut waren – wenn auch eher indirekt. Er wollte nämlich wissen, was ein Mensch alles berechnen kann und was nicht. Zu diesem Zweck hat er sich ein einfaches Gerät ausgedacht, mit dem er den Vorgang des menschlichen Rechnens an sich nachbilden wollte. Von allen nicht essenziellen Details hat er dabei abstrahiert, weshalb wir auch so etwas wie eine Tastatur oder einen Bildschirm vergeblich suchen werden.



Abb. 1.12 Alan Turing, der mit seinen theoretischen Arbeiten die Informatik grundlegend geprägt hat. (© Camille Büsser)

Ein Maschinenkonstrukt, um das menschliche Rechnen nachzubilden

Die von ihm im Jahr 1936 vorgeschlagene Rechenmaschine gilt heute als Grundmodell eines Computers und wird zu seinen Ehren als Turing-Maschine bezeichnet. Auf den ersten Blick – werfen sie mal einen Blick auf Abb. 1.13 – sieht sie allerdings eher skurril aus und scheint wenig mit einem heutigen Computer gemein zu haben.

Der Grund ist, dass sie an sich gar nicht dafür gedacht war, in die Realität umgesetzt zu werden. Turing ging es um das Wesen eines Computers, um die grundlegenden Prinzipien des automatisierten Rechnens, aber es war nicht seine Absicht, eine praktisch einsetzbare Maschine zu entwerfen. In diesem Sinne ist sein Computerkonzept ein Gedankenexperiment, ein Gebilde der Vorstellung. Entscheidend für ihn war, dass seine Maschine, so wie er sie sich auf dem Papier ausgedacht hatte, prinzipiell gebaut werden konnte – insofern hatte er eher den in Abb. 1.14 skizzierten abstrakten Bauplan im Kopf als eine konkrete Umsetzung.

Turings Minimalcomputer besteht aus einem Band, einer Vorrichtung zum Lesen, Beschriften und Bewegen des Bandes – dem sogenannten Lese- und Schreibkopf – sowie einer Steuereinheit, quasi dem Gehirn der Maschine, das die Berechnungen steuert. Das Ganze erinnert stark und vielleicht nicht von ungefähr an ein Tonbandgerät – den analogen Vorläufer der digitalen Tonaufzeichnung –, das ebenfalls in den 1930er-Jahren entwickelt wurde. Turings Assoziation war jedoch die des rechnenden Menschen: Das Band versinnbildlicht das karierte Notizbuch, der Lese- und Schreibkopf

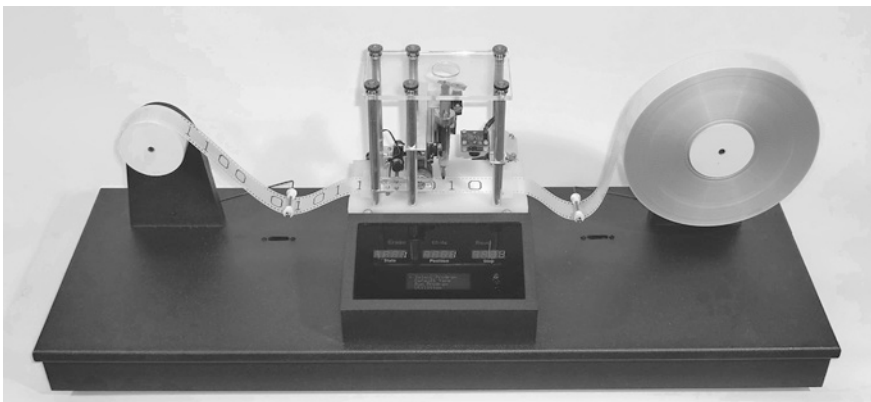


Abb. 1.13 Eine Turing-Maschine in Aktion; dieses Exemplar wurde von Mike Davey konstruiert und gebaut. (© Michael Davey)

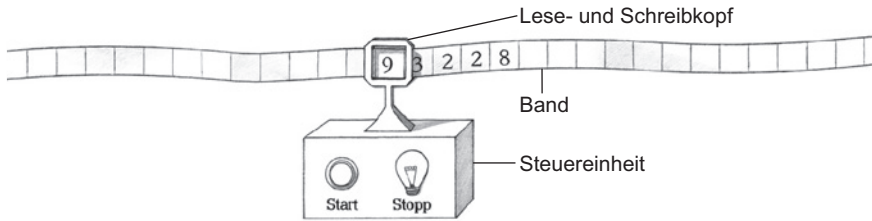


Abb. 1.14 Schematische Zeichnung einer Turing-Maschine. (© Susanne Staubli, Eckart Zitzler)

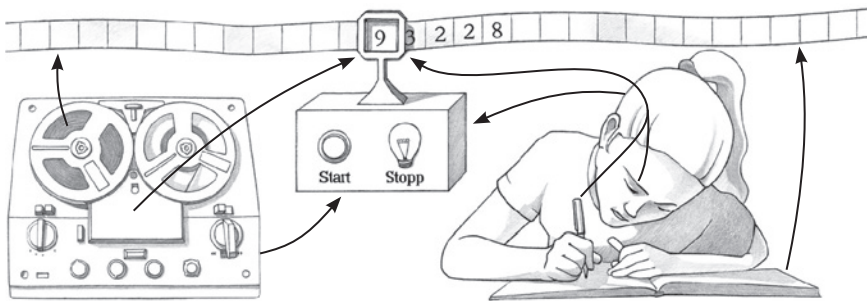


Abb. 1.15 Tonbandgerät und der rechnende Mensch im Vergleich zur Turing-Maschine. (© Susanne Staubli, Eckart Zitzler)

steht sowohl für die Hand mit dem Stift als auch für das Auge, und die Steuereinheit entspricht dem logisch denkenden Kopf, der die Rechnungen ausführt (Abb. 1.15).

Rechnen als Textbearbeitung

Turings altmodisch anmutende Maschine wirkt jedoch nicht nur in Bezug auf ihr Aussehen eigenartig, sondern auch hinsichtlich ihrer Arbeitsweise. Denn können Sie sich vorstellen, dass eine Maschine, die nichts anderes macht, als einen vorgegebenen Text durch schrittweises Ersetzen einzelner Zeichen in einen neuen Text umzuwandeln, tatsächlich rechnen kann? So absurd das auch erscheinen mag – wir werden nachher anhand zweier einfacher Rechenbeispiele den Beweis führen. Doch vorher schauen wir uns an, wie eine Turing-Maschine im Detail funktioniert.

Da gibt es zunächst einmal das Band. Aus welchem Material es besteht, ob aus Papier, Folie oder Magnetband, soll uns hier nicht weiter kümmern. Entscheidend ist, dass sich mit ihm Zeichen, seien es Ziffern,

Buchstaben oder sonstige Symbole, zwischenspeichern lassen und es damit als Gedächtnis der Maschine fungiert. Analog zu dem Karomuster auf dem Notizpapier ist das Band in einzelne Zellen aufgeteilt, von denen jede ein Zeichen aufnehmen kann. Der Einfachheit halber betrachten wir hier nur die Zahlen von 0 bis 9 und das Leerzeichen als infrage kommende Zeichen. Folglich repräsentiert das Band zu jedem Zeitpunkt eine riesige Zeichenkette (den oben erwähnten Text), die am Anfang aus lauter Leerzeichen und der Eingabe, z. B. einer Zahl in Form einer Ziffernfolge, besteht.

Mit dem Lese- und Schreibkopf kann der Speicher, also das Band, ausgelesen und manipuliert werden. Er kann das Zeichen an der Bandposition, wo er gerade steht, einlesen sowie ggf. ausradieren und durch ein anderes Zeichen ersetzen. Zudem ist der Lese- und Schreibkopf in der Lage, das Band jeweils um eine Zelle nach links oder rechts zu verschieben. Auch hier gibt es wieder verschiedene Möglichkeiten, so eine Lese-/Schreibvorrichtung zu realisieren: Bestände das Band aus Folie, so könnte beispielsweise ein wasserlöslicher Folienstift zum Beschriften, ein feuchtes Tuch zum Ausradieren und eine Kamera zum Einlesen dienen.

Schließlich ist da noch die Steuereinheit, im Bauplan als Kasten mit verborgenem Innenleben skizziert. Wie der Name sagt, steuert sie die Berechnung; dazu ist sie direkt mit dem Lese- und Schreibkopf verbunden. Sie macht nichts anderes, als den folgenden, aus drei Schritten bestehenden Vorgang immerzu zu wiederholen: erstens liest sie das aktuelle Zeichen unter dem Lese- und Schreibkopf; zweitens überschreibt sie es mit einem neuen Zeichen; drittens bewegt sie das Band um eine Zelle oder stoppt die Berechnung. Diese drei Schritte stellen zusammen die einzige Art von Rechenoperation – wir könnten auch sagen: Befehl – dar, die die Turing-Maschine kennt: das Zeichen unter dem Lese- und Schreibkopf durch ein anderes Zeichen ersetzen und anschließend das Band verschieben. Wie gesagt, Rechnen heißt in diesem Fall Textbearbeitung und tatsächlich lassen sich mit dieser Methode beliebig komplizierte Berechnungen durchführen.

Eine Rechenoperation wird über eine Tabelle beschrieben

Woher weiß die Steuereinheit nun, welches Zeichen geschrieben und ob das Band nach links oder nach rechts bewegt werden soll? Dazu muss die Maschine entsprechend programmiert werden, und zwar mittels einer Tabelle. Diese Tabelle legt für jedes gelesene Zeichen fest, welches Zeichen zu schreiben und in welche Richtung das Band zu verschieben ist. Abb. 1.16

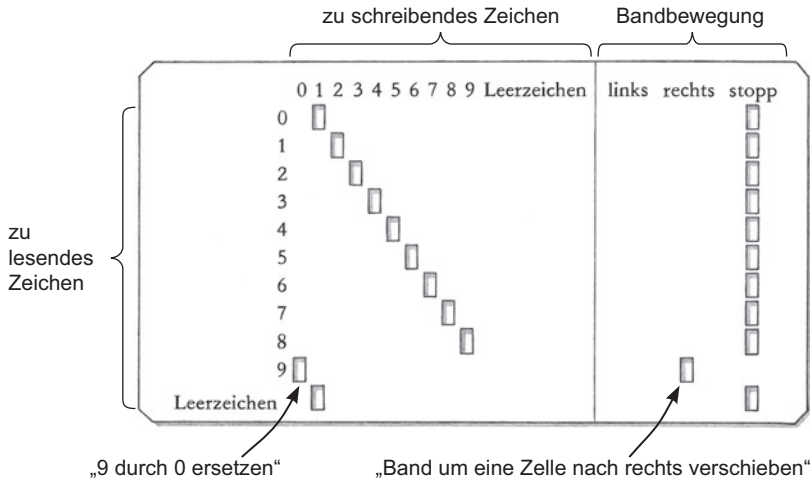


Abb. 1.16 Eine exemplarische Tabelle in Form einer Lochkarte. (© Susanne Staubli, Eckart Zitzler)

veranschaulicht das Konzept und demonstriert, wie eine Tabelle mittels eines Lochcodes auf Kartonpapier gespeichert werden kann. Die dort abgebildete Tabelle ist so gestaltet, dass bei ihrer Verwendung jede Ziffer von 0 bis 8 durch die nächsthöhere Ziffer, eine 9 durch eine 0 und ein Leerzeichen durch eine 1 ersetzt wird. Sie definiert eine konkrete Rechenoperation und verkörpert als solche bereits ein Programm – ein Programm zur Addition von 1, wie Sie anhand von Bsp. 1.3 selbst überprüfen können.

Beispiel 1.3: Eins hinzuzählen mit einer Turing-Maschine

Angenommen, die Steuereinheit sei mit der in Abb. 1.16 dargestellten Tabelle programmiert. Des Weiteren lassen Sie uns die Situation betrachten, dass die Zahl 4999 auf dem Band steht; wie diese Zahl dorthin gelangt, soll an dieser Stelle nicht interessieren. Die Ausgangslage ist also wie in Abb. 1.17 unter Punkt 1 skizziert: Außer den vier Ziffern enthält das Band einzig Leerzeichen, und der Lese- und Schreibkopf ist über der am weitesten rechts stehenden 9 platziert.

Starten wir nun die Berechnung, z. B. durch Drücken eines entsprechenden Startknopfes auf der Steuereinheit, so arbeitet die Turing-Maschine wie anfangs beschrieben. Sie liest zunächst die 9 unter dem Lese- und Schreibkopf ein und schaut dann in der Tabelle nach, was zu tun ist. Wie dort festgeschrieben, ersetzt sie die 9 auf dem Band durch eine 0 und bewegt anschließend das Band um eine Position nach rechts. Neu steht die Zahl 4990 auf dem Band.

Jetzt beginnt sie mit der zweiten Rechenoperation. Wiederum befindet sich eine 9 unter dem Lese- und Schreibkopf und wiederum wird diese mit einer 0 überschrieben und das Band nach rechts verschoben. Damit erhalten wir als zweites Zwischenergebnis die Zahl 4900.

Die dritte Rechenoperation ist identisch zu den vorherigen zwei und führt zum Zwischenergebnis 4000.

Das anschließende Lesen der 4 veranlasst die Steuereinheit, die 4 durch eine 5 zu ersetzen und die Berechnung zu stoppen – wie es durch die Tabelle vorgegeben ist. Das Ende der Berechnung könnte dabei durch das Aufleuchten eines Lämpchens signalisiert werden. Am Schluss steht die Zahl 5000 auf dem Band.

Natürlich stellt sich die Frage, wie die auf einer Tabelle festgehaltenen Informationen – also das Programm – in die Steuereinheit eingepeist werden können. Benutzen wir wie in Abb. 1.16 eine Lochkarte, um die Tabelle zu beschreiben, so kann die Steuereinheit mit einer entsprechenden Lesevorrichtung ausgestattet und das Programm über dieses Zusatzgerät geladen werden. So ein Lochkartenleser gehörte lange Zeit zur Standardausrüstung eines Rechenzentrums, denn Lochkarten waren die mobilen Speichermedien der Computer-Steinzeit (Abb. 1.18).

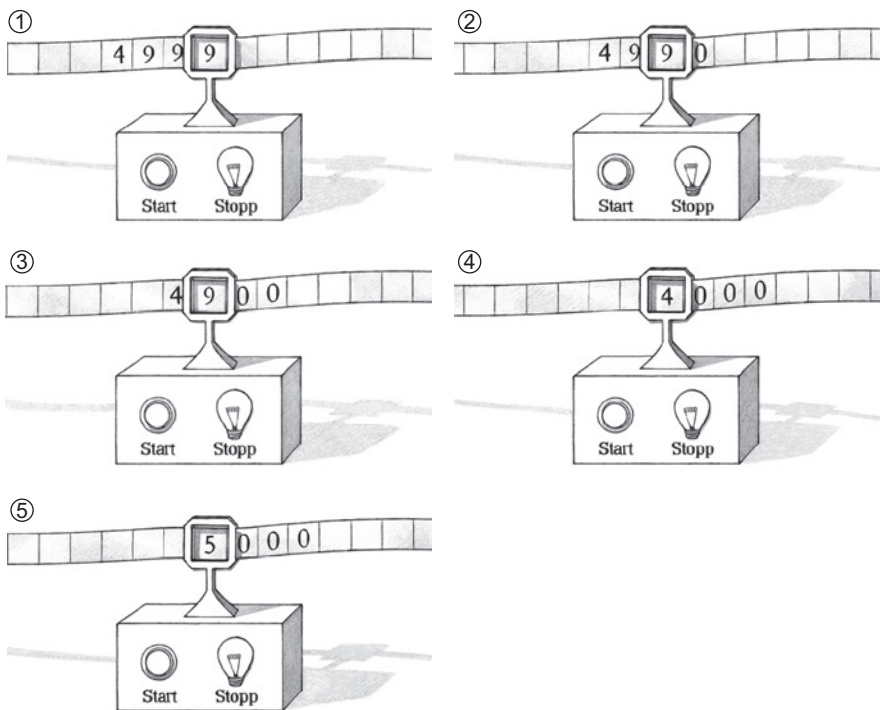


Abb. 1.17 Rechenbeispiel, wie eine Turing-Maschine zu einer vorgegebenen Zahl eins hinzuzählt. (© Susanne Staubli, Eckart Zitzler)

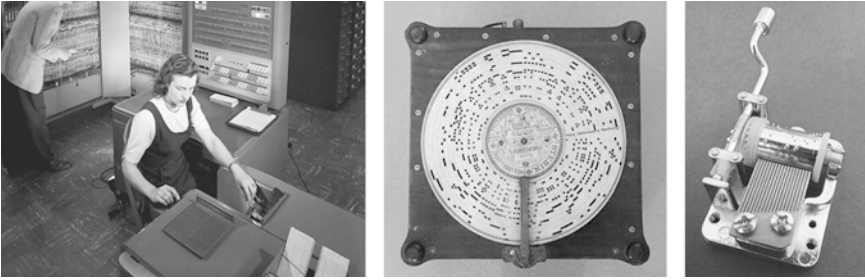


Abb. 1.18 Eine Frau bedient 1957 einen Lochkartenleser (links); ein mechanisches Musikinstrument (um 1900), das über eine mit Löchern versehene Pappscheibe „programmiert“ wird und somit verschiedene Melodien abspielen kann (Mitte); eine Spieluhr, bei der eine spezifische Melodie auf einer Walze festgeschrieben ist (rechts). (© NASA (links), Eckart Titzler (Mitte, mit freundlicher Genehmigung des Museums ENTER Solothurn), Eckart Titzler (rechts))

Das Prinzip, einen mit eingestanzten Löchern versehenen Kartonstreifen als Speichermedium zu verwenden, war bis in die 1980er-Jahre weit verbreitet, und selbst heute findet es im weiteren Sinne immer noch Verwendung: Spieluhren beispielsweise geben Melodien wieder, die auf Lochscheiben, Lochstreifen oder auf mit Stiften versehenen Metallwalzen festgehalten sind.

Mehrere Tabellen können mehrere Rechenoperationen ausführen

Der in Bsp. 1.3 beschriebene Rechenvorgang entspricht haargenau unserem Vorgehen, wenn wir schriftlich eine 1 zu einer Zahl hinzuaddieren. Wir gehen ziffernweise von rechts nach links, ersetzen die aktuelle Ziffer um die nächsthöhere und hören auf, wenn es keinen Übertrag mehr gibt. Die Tabelle verkörpert so gesehen das Regelwissen zur positionsweisen Addition von 1, welches wir als Kinder zu Beginn unserer Schulzeit lernen.

Doch sobald wir nur ein wenig kompliziertere Rechnungen, z. B. die Addition von 2, durchführen lassen wollen, reicht eine einzige Tabelle nicht mehr aus. Daher bietet die Turing-Maschine die Möglichkeit, mehrere Tabellen zu benutzen, wobei sie natürlich wissen muss, wann welche zu verwenden ist. Zu diesem Zweck werden die Tabellen kurzerhand nummeriert und jeweils um eine neue Spaltengruppe erweitert. Für jedes Zeichen, welches der Lese- und Schreibkopf möglicherweise einlesen kann, werden nicht nur das zu schreibende Zeichen und die Bandbewegung angegeben, sondern

auch die Nummer der Tabelle, die bei der nächsten Rechenoperation zu verwenden ist (siehe Abb. 1.19). Darüber hinaus muss festgelegt werden, mit welcher Tabelle die Programmausführung beginnen soll; per Konvention ist dies stets die Tabelle mit der Nummer 1.

Bsp. 1.4 zeigt auf, wie eine Programmausführung mit zwei Tabellen aussehen kann. Der Ablauf ähnelt demjenigen aus Bsp. 1.3, doch diesmal werden zwei hinzugezählt. Wollte man zwei beliebige Zahlen addieren, bräuchte man eine wesentlich größere Anzahl an Tabellen, ganz zu schweigen von komplexeren Berechnungen.

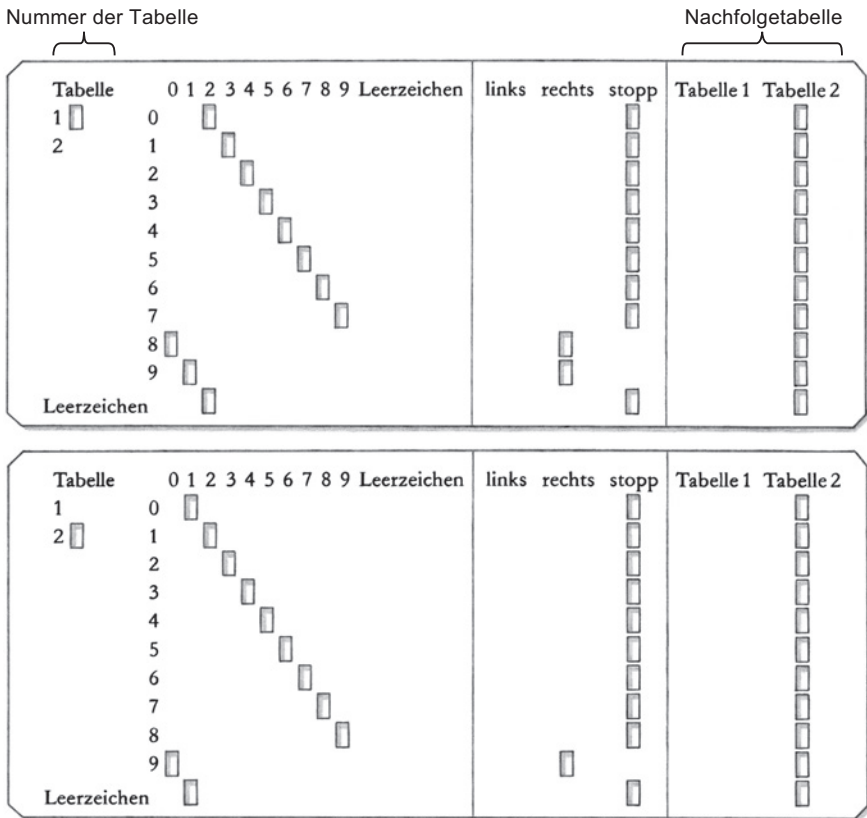


Abb. 1.19 Ein Programm mit zwei Tabellen zur Addition von 2. (© Susanne Staubli, Eckart Zitzler)

Beispiel 1.4: Zwei hinzuzählen mit einer Turing-Maschine

Programmieren wir die Turing-Maschine nun mit den Tabellen aus Abb. 1.19 und starten die Berechnung in derselben Ausgangssituation wie vorher (siehe Abb. 1.20).

Im ersten Schritt wird eine 9 gelesen, und gemäß der Konvention legt Tab. 1 die nächsten Aktionen fest. Der Tabelleneintrag zur Ziffer 9 besagt, dass die 9 durch eine 1 zu ersetzen, anschließend das Band um eine Zelle nach rechts zu verschieben und bei der nächsten Rechenoperation Tab. 2 zu verwenden ist. Wenn diese Aktion durchgeführt wurde, steht die Zahl 4991 auf dem Band.

In der nachfolgenden Rechenoperation wird wiederum eine 9 eingelesen, aber diesmal wird Tab. 2 konsultiert, um die Aktionen zu bestimmen. Diese lauten: die 9 durch eine 0 ersetzen, das Band nach rechts verschieben und auch bei der nächsten Operation Tab. 2 benutzen. Der Zwischenstand danach ist 4901.

Die nächsten zwei Rechenoperationen sind identisch zur dritten und vierten Operation im vorherigen Rechenbeispiel. Die letzte 9 wird durch eine 0 ersetzt und die 4 durch eine 5, beide Male gemäß Tab. 2. Wenn die Maschine schließlich stoppt, steht 5001 auf dem Band. Das Programm mit den zwei Tabellen hat eine Addition mit 2 bewirkt, wiederum ganz in Anlehnung an das Vorgehen beim handschriftlichen Addieren auf Papier.

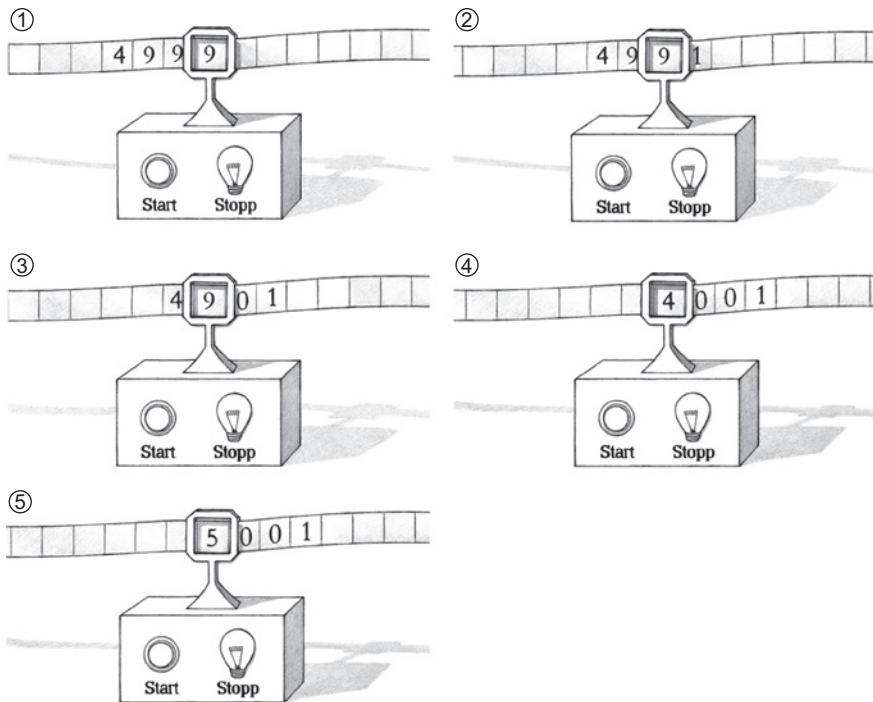


Abb. 1.20 Erweitertes Rechenbeispiel: Wie eine Turing-Maschine zu einer vorgegebenen Zahl zwei hinzuzählt. (© Susanne Staubli, Eckart Zitzler)

Bei einer Turing-Maschine ist ein Programm also über einen Satz an Tabellen beschrieben. Im Allgemeinen legt ein Programm eine Menge von konkreten Rechenoperationen oder Befehlen fest (die Tabellen) und darüber hinaus die Reihenfolge, in welcher diese auszuführen sind (bestimmt durch die Angabe der Nachfolgetabellen). Welche Befehle effektiv zur Ausführung kommen, hängt von den zu verarbeitenden Daten ab; im vorherigen Beispiel bestimmt die Anzahl der Ziffern auf dem Band, wie oft die durch Tab. 2 beschriebene Rechenoperation ausgeführt wird. Somit führt eine spezifische Programmausführung stets zu einer spezifischen Sequenz von Rechenoperationen, d. h., andere Eingabedaten können in einer anderen Befehlsabfolge resultieren.

Sinn und Unsinn von Turings Minimalcomputer

So – damit ist eigentlich alles Wesentliche zur Arbeitsweise einer Turing-Maschine gesagt. Gratulation! Sie kennen jetzt den einfachsten Computer der Welt und wissen, wie er funktioniert. Er kann Daten speichern, diese automatisch verarbeiten und ist darüber hinaus programmierbar, d. h., die Art und Weise, wie die Daten verarbeitet werden, ist flexibel einstellbar. Somit verfügt er über alle charakteristischen Merkmale, die wir einer vollwertigen Rechenmaschine zuschreiben. Obwohl er nur über einen einzigen, simplen Rechenmechanismus verfügt – ein Zeichen durch ein anderes ersetzen, das Band verschieben und ggf. die Tabelle wechseln –, steht er bzgl. seines Rechenvermögens einem beliebigen heutigen Computer in nichts nach (die Rechengeschwindigkeit ist eine andere Geschichte, aber davon ist hier nicht die Rede). Dabei umfasst Rechnen nicht nur rein mathematische Aufgaben, z. B. die Berechnung von Integralen oder die Lösung von Gleichungssystemen, sondern auch ganz praktische Probleme wie die Steuerung eines Mars-Roboters oder die Vorhersage von Wirbelstürmen (Abb. 1.21). Alles, was sich grundsätzlich im intuitiven Sinne berechnen lässt, kann auch von einer Turing-Maschine berechnet werden. Diese Aussage, die auch als Churchsche These bekannt ist, lässt sich nicht beweisen, doch die Informatikergilde ist sich einig darin, dass diese Vermutung zutrifft.

Wenn wir nun zurück zu unserer Ausgangsfrage kommen, nämlich „Was ist ein Computer?“, so könnte man salopp sagen: Ein Computer ist ein Gerät, das alles kann, was eine Turing-Maschine kann. Wer es gerne genauer hat, der kann auf folgende Definition eines Fachbuchs zur Didaktik der Informatik zurückgreifen (Schubert und Schwill 2011, S. 4):

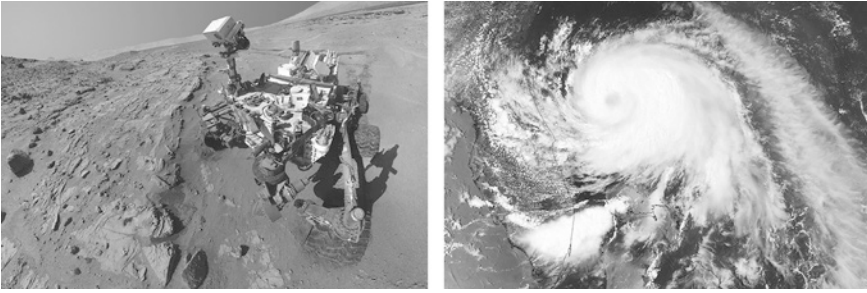


Abb. 1.21 Robotersteuerung und Wettersimulation – zwei exemplarische Anwendungen, bei denen die Berechnungen von einer Turing-Maschine durchgeführt werden könnten; die Fotos zeigen ein Selbstporträt des Mars-Roboters Curiosity (links) und die Satellitenaufnahme eines Hurrikans (rechts). (© NASA/JPL-Caltech/MSSS (links), NASA Goddard MODIS Rapid Response Team (rechts))

Ein **Computer** [...] ist ein **universell** einsetzbares Gerät zur automatischen Verarbeitung von Daten (einschl. deren Speicherung).

Auch in dieser Beschreibung finden wir die drei Stichworte Speicher, Automatik und Programmierbarkeit („universell einsetzbar“) wieder.

Schön und gut, aber was ist letztlich der Nutzen der Turing-Maschine? Die zwei Rechenbeispiele haben bereits aufgezeigt, wie aufwändig die Programmierung einer Turing-Maschine selbst für elementarste Aufgaben ist. Keine Informatikerin, kein Informatiker würde sich das antun wollen, es sei denn, im Studium zu Übungszwecken. Dann ist da noch die Frage, wie eine Turing-Maschine eigentlich bedient werden soll; denn von Ein- und Ausgabevorrichtungen war bislang keine Rede. Grundsätzlich könnte man schon Zusatzgeräte bauen, mit denen sich Zeichen auf das Band schreiben und Teile des Bandes auf einem Bildschirm anzeigen ließen. Aber so richtig praktisch würde das Ganze dann immer noch nicht, man denke nur an Foto- oder Videobearbeitung. Machen wir uns also nichts vor: Für den praktischen Einsatz ist eine Turing-Maschine kaum geeignet, dafür ist sie viel zu langsam und zu schwerfällig in der Bedienung – auch wenn es, wie wir gesehen haben, Hobbybastler gibt, die sie für Demonstrationszwecke tatsächlich bauen.

Der Nutzen ist eher theoretischer Natur. Weil sie so schön einfach ist und von allen unnötigen Details abstrahiert, eignet sie sich bestens, um das Wesen eines Computers zu illustrieren und zu studieren – die Turing-Maschine als des Pudels Kern, als Essenz eines Computers. Wie ich bereits angedeutet habe, basieren auch modernste Rechenmaschinen in ihrem Innersten auf äußerst einfachen Rechenoperationen, ähnlich

dem Verarbeitungsmechanismus einer Turing-Maschine; die Prinzipien sind also dieselben. Zudem ist Turings fiktive Maschine interessant für Informatikerinnen und Informatiker, die sich mit theoretischen Fragestellungen beschäftigen, und zwar als repräsentatives ComputermodeLL: Wenn sie zeigen wollen, ob sich ein Problem ganz allgemein mithilfe eines Computers berechnen lässt oder nicht, dann zeigen sie, dass dies mit einer Turing-Maschine geht bzw. nicht geht – und schon haben sie die Aussage auch für jeden beliebigen anderen Computer bewiesen.

Insgesamt hat Alan Turing mit seinem Gedankenexperiment und anderen Arbeiten die gesamte Computerwissenschaft so nachhaltig beeinflusst und geprägt, dass zu seinen Ehren der Turing-Award – quasi der Nobelpreis für Informatikerinnen und Informatiker – ins Leben gerufen wurde. Dass er sich in einer seiner Veröffentlichungen auch mit der Biologie beschäftigte und hier wichtige Impulse gab, ist hingegen weniger bekannt. Auch wir werden uns nun der Biologie zuwenden und am Beispiel einer Bakterie der Frage nachgehen, wie in der Natur „gerechnet“ wird.

Natürliche Informationsverarbeitung

Für die Natur sind die Grundprinzipien der automatisierten Informationsverarbeitung, wie sie Turing mit seinem Minimalcomputer skizziert hat, ein alter Hut. Jeder Organismus nimmt permanent Informationen aus seiner Um- und Innenwelt auf, wertet sie aus und merkt sich diese gegebenenfalls, um anschließend situativ darauf zu reagieren. Selbst das Konzept der Programmierbarkeit lässt sich wiederfinden, wenn wir die Variabilität des Erbguts mitberücksichtigen. Das in dieser Hinsicht ausgeklügeltste Lebewesen ist derzeit sicherlich der Mensch, dessen Gehirn enorme Informationsmengen bewältigt und speichert; aber auch Kleinstlebewesen, die nur aus einer einzigen Zelle bestehen, weisen ganz erstaunliche Steuerungsmechanismen auf. Somit ist die Thematik der selbstständig ablaufenden Informationsverarbeitung keinesfalls auf die Informatik beschränkt, stellt sie doch einen Kernaspekt des Lebens und der Evolution dar. Der Unterschied zwischen Biologie und Informatik liegt vielmehr in der Art der Systeme begründet, die untersucht werden: natürliche Organismen auf der einen Seite und vom Menschen für seine Zwecke kreierte Computer auf der anderen Seite.

Insofern scheint es naheliegend und kann aufschlussreich sein, die Informationsverarbeitung bei beiden Arten von Systemen zu vergleichen. Im Folgenden werden wir exemplarisch die Kolibakterie (Abb. 1.22), eines

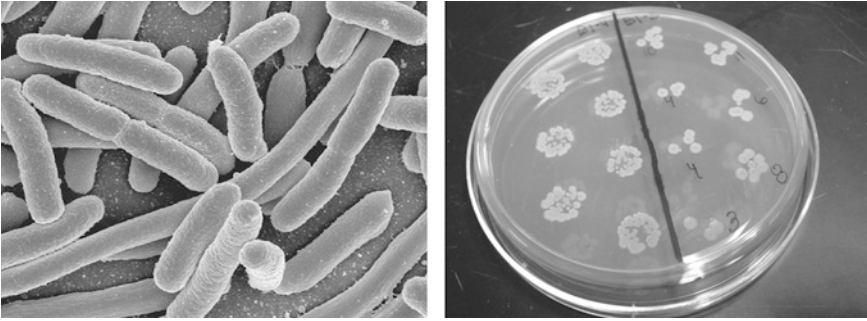


Abb. 1.22 Kolibakterien unter dem Elektronenmikroskop in 10.000-facher Vergrößerung als einzelne Zellen (links) und in einer Petrischale in Form von punktförmigen Kolonien, wie sie sich dem bloßem Auge darstellen (rechts). (© National Institute of Allergy and Infectious Diseases (links), NASA/Ames (rechts))

der am besten untersuchten Lebewesen überhaupt, der Turing-Maschine gegenüberstellen und – bei allen offensichtlichen Unterschieden – bemerkenswerte Gemeinsamkeiten entdecken. Das Interesse an diesem Bakterium kommt übrigens nicht von ungefähr. Obwohl es normalerweise als wertvoller Vitaminproduzent in unserem Darm lebt, gibt es einzelne Stämme, die für den menschlichen Organismus äußerst gefährlich sein können. Dieser Einzeller hat es also in sich!

Das Band des Lebens: die DNA

Die frappanteste Übereinstimmung ist das Vorhandensein eines Bands, das eine riesige Zeichenkette speichert und eine Schlüsselrolle bei der Informationsverarbeitung einnimmt. Das Band der Kolibakterie ist die berühmte DNA, allgemein bekannt als Trägerin des Erbguts. Man kann sie sich als enorm lange, verwirnte Kordel vorstellen, welche in einer Kapsel – dem stäbchenförmigen Bakterienkörper – aufbewahrt wird. Gemäß Abb. 1.23 scheint dieses Band nicht so akkurat aufgewickelt wie dasjenige der Turing-Maschine in Abb. 1.13; fast könnte man meinen, es sei schnell und unachtsam in die Kapsel hineingestopft worden. Doch das Gegenteil ist der Fall: All die Lebensvorgänge in der Zelle, in deren Zentrum die DNA steht, laufen mit höchster Präzision ab.

Erstaunlich ist auch der Aufbau dieses biologischen Langzeitspeichers, und es war eine Sensation, als 1953 zum ersten Mal seine chemische Struktur, die Doppelhelix, beschrieben wurde. An dieser Stelle soll uns jedoch einzig interessieren, dass es sich bei der DNA um ein Kettenmolekül

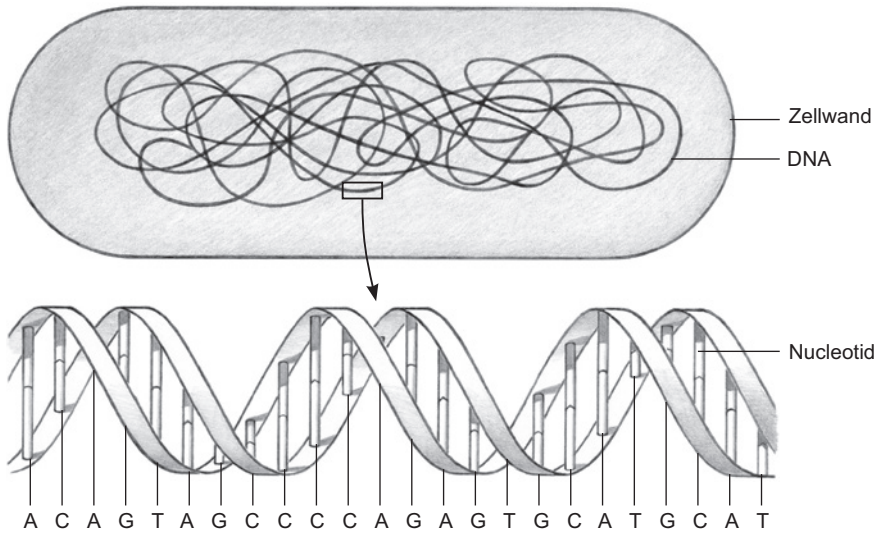


Abb. 1.23 Oben ist eine Kolibakterie im Querschnitt schematisiert; in der Mitte ist die DNA als langfädiges, verknäultes Molekül dargestellt. Ein Ausschnitt der DNA ist unten in einer detaillierten Zeichnung zu sehen, zusammen mit der codierten Zeichenfolge. (© Susanne Staubli, Eckart Zitzler)

handelt, das sich aus vier Arten von Kettengliedern zusammensetzt: den Nucleotide genannten Molekülen Adenin, Cytosin, Guanin und Thymin. Die Nucleotide sind wie auf einer Perlenschnur hintereinander aufgereiht, und da ihre Namen häufig durch die entsprechenden Anfangsbuchstaben abgekürzt werden, lässt sich die DNA als eine Zeichenfolge begreifen: das Buch des Lebens, verfasst in einer Alphabetschrift mit den vier Buchstaben A, C, G und T (Abb. 1.23). In der Kolibakterie umfasst der codierte Text mehr als 4 Mio. Zeichen, entsprechend lang ist das biologische Band. Die DNA ist ausgestreckt mehr als 200-mal länger als eine einzelne Kolibakterie und muss mehrfach gefaltet und dicht verpackt werden, um in das Innere der Zelle hineinzupassen. Die so erzielte Speicherdichte ist enorm. Um den in einem Gramm DNA codierten Text zu speichern, bräuchte man eine Milliarde mal eine Milliarde Lochkarten mit einem Gesamtgewicht von über 6 Mio. t.

Schließlich gibt es auch ein Pendant zum Lese- und Schreibkopf der Turing-Maschine: ein Molekül namens RNA-Polymerase, über das der Bandinhalt ausgelesen werden kann (Schreiben ist eine andere Geschichte, dazu später mehr). Die RNA-Polymerase lagert sich an der DNA an, tastet sie Nucleotid für Nucleotid bzw. Zeichen für Zeichen ab und überträgt

die gelesene Sequenz auf ein anderes Molekül. Es wird also eine Abschrift erstellt, allerdings nicht von der gesamten DNA, sondern nur von kleineren Abschnitten, Gene genannt. Anfang und Ende dieser Abschnitte sind jeweils durch spezielle Nucleotidabfolgen markiert, sodass die RNA-Polymerase weiß, von wo bis wo sich das abzulesende Textsegment erstreckt.

Welchem Zweck die resultierenden Textschnipsel dienen? Im Wesentlichen sind es Konstruktionspläne, anhand derer die Kolibakterie in einem komplizierten Zusammenspiel von über 50 molekularen Maschinen Proteine herstellt, d. h. Werkzeuge und Baumaterial für den Lebensunterhalt (Bsp. 1.5). Proteine sind essenziell für das Funktionieren des Organismus, denn diese Moleküle sind an nahezu allen Vorgängen in der Zelle beteiligt und regeln diese – kein Wunder, machen sie gesamt- haft doch mehr als die Hälfte des Trockengewichts einer Kolibakterie aus. Entsprechend groß sind ihre Formenvielfalt – vom Stab über das Netz bis zur Hohlkugel – und ihr Anwendungsspektrum. Als Enzyme können Proteine Stoffe verbinden, zerlegen und umwandeln, also chemische Reaktionen in Gang setzen; als Transportmittel bewegen sie; als Trägerelemente stabilisieren sie; und als Teamplayer können sie molekulare Miniaturmaschinen bilden, von der Pumpe zum Motor bis zur komplexen Baumaschine – es scheint nichts zu geben, was es nicht gibt. Es sind folglich die Proteine, über die die Bakterienzelle die Lebensvorgänge in ihr steuert.

Beispiel 1.5: Wie ein Protein hergestellt wird

Abb. 1.24 skizziert den Ablauf, wie aus dem auf der DNA gespeicherten Bauplan ein Protein hergestellt wird. Die RNA-Polymerase übt die Funktion eines Lesekopfes aus, indem sie an der DNA entlangfährt und die gespeicherten Informationen auf ein anderes Molekül – die Abschrift – überträgt. Aus der Abschrift wird anschließend in hier nicht weiter erläuterten Verarbeitungsschritten das Protein zusammengesetzt, dessen Aussehen je nach Bauplan stark variieren kann.

Damit haben wir bereits einen Teil des Informationsverarbeitungsapparats einer Kolibakterie kennengelernt, nämlich die DNA, die die einzelnen Verarbeitungsschritte und ergo den gesamten Verarbeitungsprozess beschreibt, sowie die Proteine, die die Verarbeitungsschritte effektiv umsetzen. Mit den Begrifflichkeiten der Biologie könnte man es auch so ausdrücken, dass die DNA das Verhaltensrepertoire definiert und die Proteine konkrete Reaktionen realisieren. Ein wichtiger Aspekt fehlt dabei allerdings noch: die zeitliche Steuerung. Ein Verarbeitungsvorgang im Sinne einer

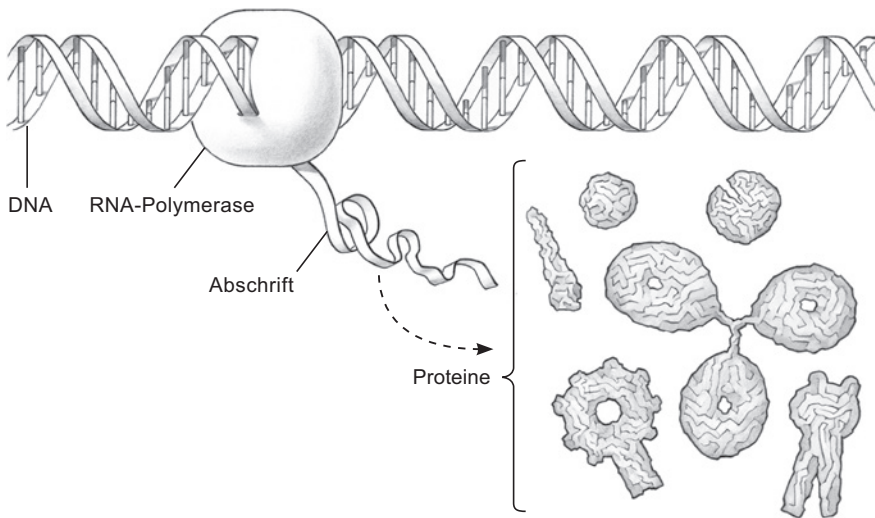


Abb. 1.24 Von der DNA zum Protein, in Anlehnung an Zeichnungen von David S. Goodsell (2010). (© Susanne Staubli, Eckart Zitzler)

komplexen Reaktion ist zum einen situationsabhängig und zum anderen zusammengesetzt aus einzelnen Teilschritten, die in einer bestimmten Reihenfolge ausgeführt werden, also aufeinander abgestimmt sein müssen. Daher dürfen nicht alle Proteine in der Zelle gleichzeitig operieren – nicht auszumalen, wenn alle möglichen Reaktionen simultan ablaufen würden. Entscheidend ist, dass sie gezielt und nach Bedarf eingesetzt werden. Wie entscheidet die Kolibakterie also, wann welches Protein zum Einsatz kommt, und wie wird diese Entscheidung in die Tat umgesetzt?

Proteine beeinflussen den DNA-Ablesevorgang

Es ist kaum verwunderlich, dass wiederum Proteine bei der Regulation der Proteinproduktion eine wesentliche Rolle spielen, indem sie sich an spezifische Stellen der DNA andocken und damit verhindern, erschweren, fördern oder ermöglichen, dass bestimmte DNA-Abschnitte abgelesen werden können. Wir werden den Mechanismus an einem konkreten Vorgang studieren: wie eine Kolibakterie Nahrung in Form von Milchzucker aufnimmt und verwertet. Dazu ist es nützlich zu wissen, dass die Kolibakterie zum festen Bestandteil unserer Darmflora gehört und dort wichtige Funktionen ausfüllt, sei es, dass sie Nährstoffe spaltet, Vitamine bildet oder für die Abwehr

von Krankheitserregern sorgt. Sie selbst lebt von dem, was wir uns mit dem Essen zuführen – bevorzugt von Traubenzucker, aber wenn der nicht verfügbar ist, kann es eben auch Milchzucker sein (Abb. 1.25). Letzteren kann die Kolibakterie allerdings nicht direkt verwerten, sie muss ihn zunächst einmal ins Zellinnere befördern und dort in seine Bestandteile zerlegen. Für beide Aufgaben gibt es spezialisierte Proteine: ein Transport- und ein Schneideprotein.

Die zwei Proteine werden allerdings nur produziert, wenn sie auch wirklich gebraucht werden. Die Kolibakterie muss ihre Werkzeuge ökonomisch und gezielt einsetzen, daher werden auch nicht mehr benötigte Proteine durch andere Proteine abgebaut – die Bausteine der Proteine, die Aminosäuren, bleiben so im Spiel. Die Produktion muss sich also an- und ausschalten lassen, und das wird mittels eines extra dafür konzipierten Schalterproteins erreicht. Dieses Schalterprotein ist so beschaffen, dass es sich an einer ganz bestimmten Stelle an der DNA anlagern kann – genau dort, wo die Konstruktionspläne für die besagten zwei Proteine abgelegt sind. Im Normalzustand ist der Schalter auf „aus“ gelegt, d. h., das Schalterprotein ist an der DNA angedockt (Abb. 1.26 links). Die Folge ist, dass die RNA-Polymerase, der Lesekopf, nicht oder nur erschwert an der DNA entlangfahren und die Abschrift erstellen kann. Die Produktion von Transport- und Schneideprotein ist folglich stark eingeschränkt. Erst wenn Milchzuckermoleküle mit dem Schalterprotein in Berührung kommen, verändert letzteres seine Form, löst sich von der DNA und gibt so die Produktion von Transport- und Schneideprotein frei; auf diese Weise wird der Schalter umgelegt (Abb. 1.26 rechts).

Dieses Beispiel illustriert, wie eine Kolibakterie Informationen aus ihrer Umwelt verarbeitet und daraus ein Verhalten „berechnet“, wenn man so sagen will. Vereinfacht ausgedrückt: Ein Signal („Milchzucker vorhanden“)



Abb. 1.25 Kolibakterien ernähren sich u. a. von verschiedenen Zuckern, wie sie in Trauben oder Milchprodukten enthalten sind. (© Eckart Zitzler)

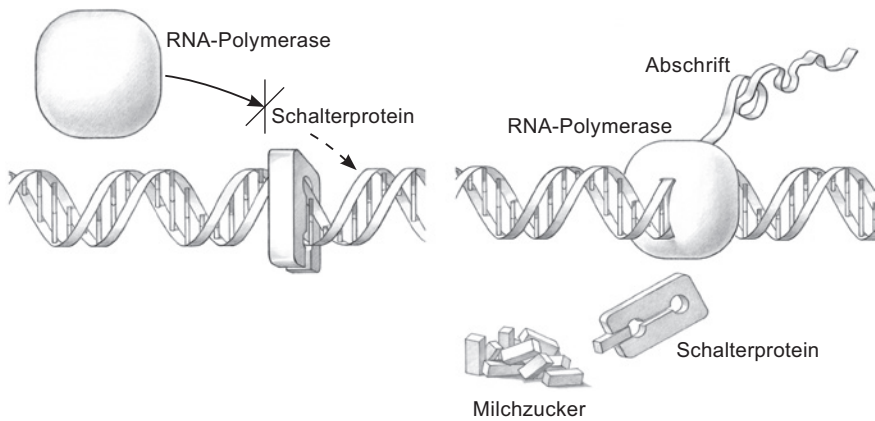


Abb. 1.26 Zusammenspiel von Schalterprotein und Milchzuckermolekülen. (© Susanne Staubli, Eckart Zitzler)

trifft ein, ein Protein reagiert darauf, womit die Herstellung anderer Proteine angestoßen und eine Antwort („Milchzucker verwerten“) provoziert wird. Im Detail ist das alles viel komplizierter, z. B. kann ein Schalterprotein auch ein anderes Schalterprotein aktivieren, womit sich vielfach verästelte Kaskaden solcher Schaltvorgänge sowie Rückkopplungen ergeben können. Für den Moment hingegen wollen wir es bei dieser oberflächlichen Betrachtung belassen.

Sind Kolibakterien Turing-Maschinen?

Jetzt, nachdem wir die groben Züge ihres Verarbeitungsmechanismus kennengelernt haben, können wir die Kolibakterie der Turing-Maschine gegenüberstellen und aus der Perspektive der Informatik beleuchten. Um den Vergleich zu vereinfachen, betrachten wir ein leicht modifiziertes Computermode. Bei dieser Variante werden die Tabellen nicht über einen Lochkartenleser, sondern über ein Zusatzband eingespeist. Die Beschreibung der Rechenoperationen ist also in einem geeigneten Format auf diesem Speichermedium abzulegen, bevor die Turing-Maschine gestartet wird. Während der Programmausführung schaut die Steuereinheit dann jeweils auf dem Zusatzband nach, um den nächsten Berechnungsschritt zu bestimmen. Anm. 1.2 verdeutlicht die Idee.

übernimmt damit die gleiche Rolle, die die Steuereinheit bei der Turing-Maschine innehat. Die Konstruktionspläne für die Proteine, die als Gene auf der DNA festgeschrieben sind, korrespondieren mit den Tabellen, die die Rechenoperationen der Turing-Maschine beschreiben; vereinfacht ausgedrückt: **Gene = Tabellen**. Folglich enthält die DNA – analog dem Zusatzband – das Programm und definiert das Spektrum der zur Verfügung stehenden Befehle, ihr kommt die Funktion eines Programmspeichers zu; sprich: **DNA = Zusatzband**. Dann sind da noch die Stoffe und Signale, die die Kolibakterie pausenlos aufnimmt und verarbeitet; ihnen stehen die Zahlen und ganz allgemein die Daten gegenüber, die die Turing-Maschine bearbeitet, also: **Stoffe/Signale = Daten**. Dabei erfüllt der Bakterienkörper die Aufgaben des Bands, weil er als Reservoir für die Daten, also als Datenspeicher fungiert. Es ließen sich über die in Abb. 1.28 gezeigten Parallelen hinaus noch viele weitere aufführen, z. B. zwischen den Schalterproteinen (die durch das Andocken an die DNA beeinflussen, welche Gene überhaupt abgelesen werden) und der gerade aktiven Tabelle bei der Turing-Maschine; doch die erwähnten Punkten mögen hier genügen.

Natürlich ist die Darstellung simplifiziert. Trotzdem haben die Entsprechungen etwas für sich, denn tatsächlich lassen sich bei der Kolibakterie die drei beschriebenen Charakteristika eines Computers wiederfinden: Speicherfähigkeit, selbsttätige Ablaufsteuerung und – im übertragenen Sinn – Programmierbarkeit. Letztere Eigenschaft ist erfüllt, weil die DNA veränderbar ist und dadurch die Art und Weise der Informationsverarbeitung modifiziert werden kann. Das Schreiben erfolgt

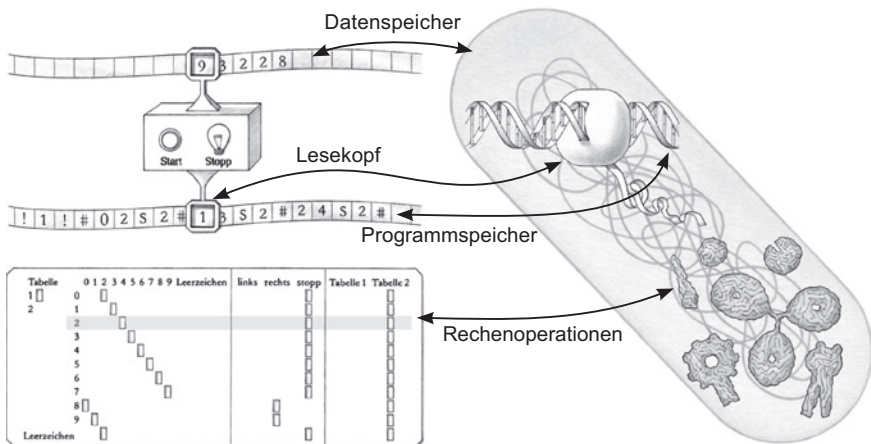


Abb. 1.28 Analogien in der Informationsverarbeitung zwischen Turing-Maschine und Kolibakterie. (© Susanne Staubli, Eckart Zitzler)

allerdings nicht über einen Lese- und Schreibkopf, sondern über andere Mechanismen wie Mutationen. So gesehen stellt sich durchaus die Frage, ob unser Darmbewohner im Grunde genommen nicht eine Art erweiterte Turing-Maschine repräsentiert. So interessant solche Analogien auch sind, sie haben immer ihre Grenzen. Die Tücke steckt bekanntlich im Detail.

Wundersam vertrackt wird es beispielsweise beim Begriff des Programms. Bei einer Turing-Maschine verstehen wir darunter eine Folge von einzelnen, nach dem gleichen Schema aufgebauten Rechenschritten, die bei gleicher Eingabe stets dasselbe Ergebnis produziert. Bei einer Kolibakterie ist es viel schwieriger darzulegen, was eigentlich das Programm ist: Am ehesten lässt es sich als ein mit der Umwelt wechselwirkendes Geflecht vieler verschiedener Verarbeitungsaktivitäten beschreiben, welches erstaunlich präzise, aber niemals exakt gleich reagiert. Auch bei der Konstruktion wartet die Kolibakterie mit so einigen Besonderheiten auf. Beispielsweise ist das Pendant zur Steuereinheit, die Gesamtheit an Proteinen, selbst Teil der Programmausführung, da ihre Baupläne auf der DNA gespeichert und abgelesen werden. Eine Maschine, die sich selbst erschafft? Das erinnert an das Motiv der zwei sich gegenseitig zeichnenden Hände (Abb. 1.29), ursprünglich 1948 vom Holländer M. C. Escher erdacht und seitdem vielfach variiert.

Damit ist klar: Eine Kolibakterie lässt sich nicht einfach als ein Computer in anderer Gestalt betrachten, zu unterschiedlich sind die beiden Systeme in ihrer Wesensart. Zumal sich der Aufbau einer Bakterie nicht an den Belangen des Menschen, z. B. Übersichtlichkeit, Handhabbarkeit und Beherrschbarkeit, orientiert, sondern sich an der Anpassungs- und

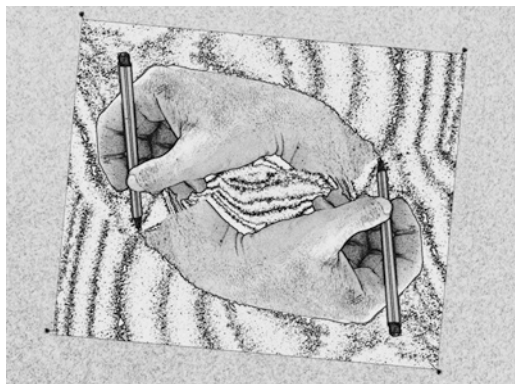


Abb. 1.29 Zwei sich zeichnende Hände als Metapher für das intrazelluläre Wechselspiel zwischen DNA und Protein, hier als Fotocollage in Szene gesetzt. (© Eckart Zitzler)

Überlebensfähigkeit des Organismus ausrichtet – das macht die Biologie ja so faszinierend. Vielleicht ist das auch gar nicht das Überraschende und Entscheidende an dieser Gegenüberstellung. Erstaunlich ist vielmehr, wie viele Querbezüge und Ähnlichkeiten wir feststellen können, **obwohl** Turing-Maschine und Kolibakterie auf den ersten Blick rein gar nichts miteinander zu tun haben. Denken wir nur an das Band als Speichermedium. Wohlgermerkt hat Alan Turing seine Ideen gut 15 Jahre vor der Entdeckung der DNA-Struktur entwickelt. Die Übereinstimmungen deuten darauf hin, dass gewisse Grundprinzipien der automatischen Informationsverarbeitung generell sind, obschon wir gesehen haben, dass bei Verallgemeinerungen Vorsicht angebracht ist. Gerade deshalb ist es stimulierend und aufschlussreich, die Dinge einmal aus der Perspektive der anderen Disziplin zu betrachten; es fördert das Verstehen und Durchdringen auf beiden Seiten. So ein Perspektivenwechsel kann schließlich dazu führen, dass eigenständige Forschungsgebiete entstehen, die beide Welten miteinander verbinden.

Rechnen mit DNA

DNA Computing ist so ein Forschungsgebiet. Wie die englische Bezeichnung nahelegt, dreht sich alles darum, auf der Basis von DNA Berechnungen durchzuführen, quasi eine Art DNA-basierten molekularen Computer zu bauen. Dabei hat man sich weniger einen Computer aus Fleisch und Knochen oder eine riesige rechnende Zelle vorzustellen, sondern das Ziel ist vielmehr, Kernprozesse der Informationsverarbeitung – wie in der Kolibakterie – mittels chemischer Vorgänge zu realisieren und dabei die DNA – anders als in der Kolibakterie – als Datenspeicher zu benutzen.

Eine verwegene Idee, doch wenn wir uns die obige Diskussion vor Augen führen, nicht abwegig. Die Natur stellt ja die Werkzeuge, mit der DNA gezielt manipuliert werden kann, größtenteils selbst zur Verfügung: Proteine. Mit gewissen lassen sich DNA-Moleküle kopieren und vervielfältigen, andere können DNA-Moleküle an bestimmten Stellen zerteilen, und wiederum andere ermöglichen es, DNA-Moleküle spezifisch aneinanderzuketten und zu einem Strang zu verbinden. Daneben gibt es die Biotechnologie. Deren Methoden erlauben es beispielsweise, DNA-Moleküle gemäß einer vorgegebenen A-C-G-T-Sequenz synthetisch herzustellen, den in einem DNA-Molekül gespeicherten Text auszulesen oder DNA-Moleküle nach Größe zu sortieren und entsprechend zu selektieren. Setzt man diese Hilfsmittel geschickt ein, dann lassen sich einfache Operationen auf molekularen Zeichenketten implementieren – und schließlich Berechnungen durchführen, denn auch die Turing-Maschine rechnet ja

einzig mittels elementarer Zeichenoperationen und besitzt trotzdem die gleiche Mächtigkeit wie ein moderner Computer.

Der Beweis, dass diese Idee Hand und Fuß hat, wurde 1994 von Leonard Adleman erbracht, einem amerikanischen Informatiker (Abb. 1.30). Indem Adleman ein Verfahren vorstellte, wie sich ein bekanntes informatisches Problem, das Hamilton-Pfad-Problem, im Prinzip mit den oben aufgeführten Hilfsmitteln – DNA, Enzymen und biotechnologischen Methoden – lösen lässt, wurde DNA Computing Realität.

Das Hamilton-Pfad-Problem – wir betrachten hier eine spezielle Variante – ist schnell erklärt. Auf einem Stadtplanausschnitt ist eine Route von einem Start- zu einem Zielpunkt zu ermitteln, die alle auf dem Ausschnitt ersichtlichen Kreuzungen genau einmal quert. Diese Aufgabenstellung kann sich z. B. ergeben, wenn an den Kreuzungen Abfallbehälter stationiert sind und die Müllabfuhr diese in einer einzigen Tour leeren möchte. Einfach zu erklären heißt jedoch nicht einfach zu lösen – die Routenwahl kann zu einer echten Herausforderung werden, insbesondere wenn viele Einbahnstraßen die Fahrtrichtungen einschränken. Wie ist nun Adleman dieses Problem angegangen? Mit roher Gewalt, könnte man sagen, denn die verwendete Strategie ist in Informatikerkreisen unter dem englischen Ausdruck *brute force* bekannt.

Zunächst einmal hat er sich überlegt, wie eine Route auf der Basis des A-C-G-T-Alphabets beschrieben werden kann. Dazu hat er die Kreuzungen mit eindeutigen Namen versehen, z. B. AC oder AG. Die Bezeichnungen für die Fahrbahnen hat er anschließend aus diesen abgeleitet, und zwar durch Aneinanderreihung der Namen der zwei verbundenen Kreuzungen. Die Fahrbahn, die von Kreuzung AC zu Kreuzung AG führt, heißt dann ACAG, wie in Abb. 1.31 illustriert. In diesem Fall ist eine Route nichts



Abb. 1.30 Leonard Adleman, der Begründer des DNA Computing. (© Camille Büsser)

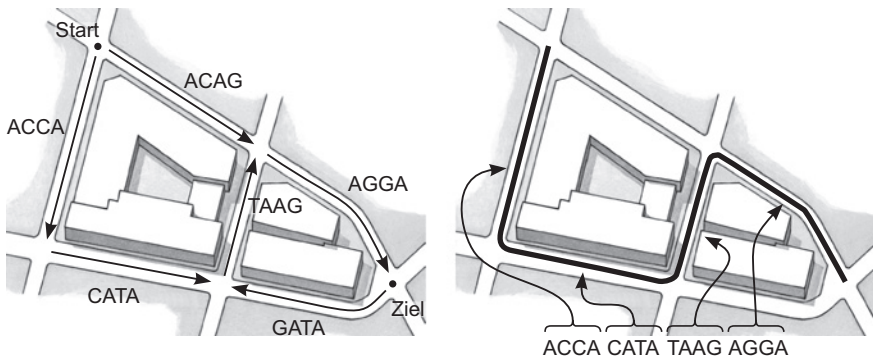


Abb. 1.31 Ein Stadtplanausschnitt mit fünf Kreuzungen und sechs Einbahnstraßen: Auf der linken Seite lässt sich erkennen, wie die Kreuzungen und die erlaubten Fahrtrichtungen benannt werden können; die rechte Seite zeigt das Beispiel einer korrekten Route und ihrer Beschreibung als A-C-G-T-Zeichenkette vom Start- zum Zielpunkt. (© Susanne Staubli, Eckart Zitzler)

anderes als eine Folge von Fahrbahnbezeichnungen, d. h. A-C-G-T-Zeichenketten, und lässt sich entsprechend mittels eines DNA-Moleküls repräsentieren. Doch wie kommt man zu diesem DNA-Molekül und vorzugsweise zu einem, das die gesuchte Route verkörpert?

Das Prinzip an sich ist simpel. Wir stellen für jede Fahrbahn ein entsprechendes DNA-Molekül als Namensträger her, vervielfältigen die resultierenden DNA-Schnipsel in enormen Mengen, lassen nun beliebige Kombinationen aus diesen DNA-Schnipseln entstehen und sortieren anschließend alle DNA-Stränge, die keine gültige Route repräsentieren, aus. Bsp. 1.6 erläutert die einzelnen Arbeitsschritte.

Beispiel 1.6: Routen finden mittels DNA Computing

In Abb. 1.32 sehen Sie den „Berechnungsprozess“ – es ist ja eigentlich ein chemischer Verarbeitungsprozess – grafisch dargestellt. Die Ausgangsbasis bilden die Namen der vorhandenen Fahrbahnen zwischen den Kreuzungen, die synthetisch in Form von DNA-Schnipseln hergestellt werden und die elementaren Lösungsbausteine repräsentieren. In einem ersten Verarbeitungsschritt werden diese Bausteine in großen Mengen vervielfältigt. Im zweiten Schritt werden die DNA-Moleküle zu längeren DNA-Strängen zusammengeklebt – zufällig, je nachdem, welches Molekül gerade auf welches trifft, aber immer so, dass die zusammengefügte Teilrouten an einer Kreuzung ineinander übergehen, sich also zu einem Weg ergänzen. Der dritte Verarbeitungsvorgang umfasst mehrere Filterungsstufen, die zum Ziel haben, ungültige Routen auszusortieren. Da muss z. B. getestet werden, ob alle Kreuzungsnamen **mindestens** einmal vorkommen, und gleichzeitig muss sichergestellt sein, dass jeder Kreuzungsname

höchstens einmal vorkommt. Das Filtern hat den Effekt, dass alle DNA-Moleküle, die zum Schluss im Reagenzglas übrig bleiben, gültige Routen verkörpern. Entsprechend genügt es im vierten Schritt, einen der verbleibenden DNA-Stränge in die dazugehörige Buchstabensequenz zurückzuübersetzen; Nucleotid für Nucleotid wird das Molekül ausgelesen und in die codierte A-C-G-T-Zeichenkette übersetzt, welche die gesuchte Lösung beschreibt.

Das Ganze setzt natürlich voraus, dass beim Kombinieren der Schnipsel tatsächlich eine korrekte Route generiert wird; hundertprozentige Sicherheit gibt es hierbei nicht, nur eine sehr hohe Wahrscheinlichkeit. Dieser Ansatz, effektiv alle möglichen – korrekten und unsinnigen – Routen zu generieren und dann die meisten davon wegzwerfen, hat tatsächlich etwas Brachiales an sich, doch die Winzigkeit der Moleküle und die massive Parallelität bei den chemischen Prozessen machen dies möglich, wenn die Anzahl der Kreuzungen nicht zu groß ist. Das also kann Rechnen im Reagenzglas heißen.

Zugegeben: Das Ganze ist ziemlich aufwändig und erinnert noch nicht wirklich an einen universell einsetzbaren, bedienfreundlichen Computer – doch wie gesagt, es war die Geburt eines Forschungsgebiets. Natürlich stellt sich die Frage, was man sich vom Rechnen mit DNA-Molekülen verspricht. Die reine Neugierde spielt in der Wissenschaft immer eine Rolle,

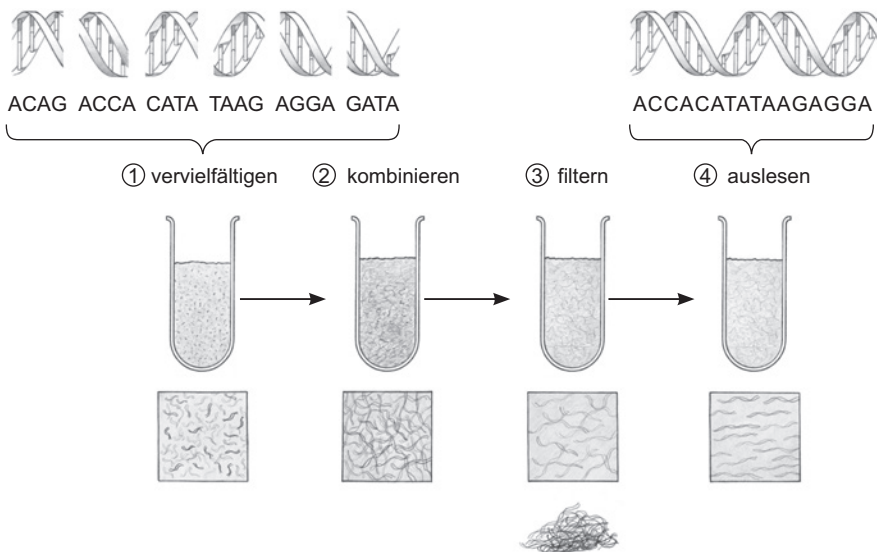


Abb. 1.32 Wie sich mit DNA rechnen lässt, illustriert am Beispiel des Hamilton-Pfad-Problems. (© Susanne Staubli, Eckart Zitzler)

aber letztlich steckt wesentlich mehr dahinter. Der große Vorteil eines wirklichen DNA-Computers gegenüber seinem elektronischen Bruder wäre seine Fähigkeit, eine immense Anzahl von Rechenoperationen auf kleinstem Raum und bei sehr geringem Energieaufwand parallel ausführen zu können. Tatsächlich haben sich Forscher Gedanken darüber gemacht, wie sich eine Turing-Maschine und abgespeckte Varianten davon auf der Basis von DNA, Proteinen und weiteren Molekülen bauen ließen – zunächst in der Theorie. Adlemans Brute-force-Ansatz erwies sich jedoch als nicht praktikabel: Zur Lösung realer Probleme bräuchte man eine unvorstellbare Menge an Molekülen. Trotzdem gehen die Arbeiten weiter. Knapp 20 Jahre nach Adlemans Versuchen gelang es Forschenden, DNA als Datenspeicher zu verwenden und eine Art biologischen Transistor zu bauen. Noch stellt DNA Computing hauptsächlich eine Ideen-Spielwiese dar, denn konkrete praktische Anwendungen haben sich bislang nicht daraus ergeben. Doch der Ideen gibt es viele. Wir werden sehen, wohin die Forschungsarbeiten effektiv führen und ob wir jemals eine organische Miniaturrechenmaschine in unseren Händen halten werden.

Fazit

Der Ausgangspunkt für dieses Kapitel war die simple Frage „Was ist ein Computer?“, und von da aus ging es im Schnelldurchlauf von der Geschichte des Computers über das grundlegende Berechnungsmodell der Theoretischen Informatik zu den Grundlagen der Molekularbiologie, inklusive Synthese dieser Themen. Wir haben gesehen: Ein Computer ist weit mehr als ein Rechner, er stellt ganz allgemein ein künstliches System dar, mit dessen Hilfe sich die Verarbeitung von Informationen automatisieren lässt. Charakteristisch ist, dass er Daten speichern sowie diese gemäß einer vorgegebenen Abfolge von Operationen manipulieren kann und dass er es ermöglicht, die Verarbeitungsvorschrift selbst flexibel einzustellen. Im weitesten Sinne weisen auch Lebewesen diese Eigenschaften auf, mit dem Unterschied, dass es die Natur ist, die diese Systeme erschafft. Daran lässt sich erkennen, dass grundlegende Prinzipien der automatischen Informationsverarbeitung universell sind und bereits Milliarden von Jahren vor dem Zeitpunkt erfunden wurden, als der erste Computer die Weltbühne betrat. Insofern haben viele informatische Fragestellungen einen Bezug zur Biologie und umgekehrt; eigenständige Forschungsgebiete, die beide Welten miteinander verbinden, unterstreichen dies.

Zum Nach- und Weiterlesen

- Goodsell, D. S. (2010). *Wie Zellen funktionieren: Wirtschaft und Produktion in der molekularen Welt* (2. Aufl.). Heidelberg: Spektrum Akademischer Verlag.
- Hofstadter, D. R. (1992). *Gödel, Escher, Bach – ein Endloses Geflochtenes Band*. München: Deutscher Taschenbuch.
- Lippe, W.-M. (2013a). *Die Geschichte der Rechenautomaten: Von der Himmelscheibe von Nebra bis zu den ersten Rechenmaschinen*. Berlin: Springer Vieweg.
- Lippe, W.-M. (2013b). *Die Geschichte der Rechenautomaten: Von mechanischen Chiffriergeräten bis zu den ersten programmierbaren Rechnern*. Berlin: Springer Vieweg.
- Padua, S. (2015). *The thrilling adventures of lovelace and babbage: The (Mostly) true story of the first computer*. New York: Pantheon Books.
- Schubert, S., & Schwill, A. (2011). *Didaktik der Informatik* (2. Aufl.). Heidelberg: Spektrum Akademischer.

Bildquellen

- Abb. 1.3: NASA, <http://spaceflight.nasa.gov/gallery/images/shuttle/sts-101/html/jsc2000e10522.html> (links, abgerufen am 11.04.2017); NASA, http://www.nasa.gov/mission_pages/station/multimedia/S.135e011867.html (rechts, abgerufen am 11.04.2017)
- Abb. 1.7: Reinhard Atzbach, <http://www.rechenwerkzeug.de/stephenson.htm> (abgerufen am 11.04.2017)
- Abb. 1.11: U. S. Army, <http://ftp.arl.mil/ftp/historic-computers/> (abgerufen am 22.08.2014)
- Abb. 1.13: Michael Davey, <http://www.aturingmachine.com/> (abgerufen am 11.04.2017)
- Abb. 1.18: NASA, <https://www.flickr.com/photos/nasacommons/9467782802/in/set-72157634982467137> (links, abgerufen am 11.04.2017)
- Abb. 1.21: NASA/JPL-Caltech/MSSS, <http://photojournal.jpl.nasa.gov/catalog/PIA18390> (links, abgerufen am 11.04.2017); NASA Goddard MODIS Rapid Response Team, http://www.nasa.gov/sites/default/files/arthur.a2014185.1520.2km_2.jpg (rechts, abgerufen am 11.04.2017)
- Abb. 1.22: National Institute of Allergy and Infectious Diseases, http://www.niaid.nih.gov/topics/biodefenserelated/biodefense/publicmedia/Pages/image_library.aspx (links, abgerufen am 22.08.2014); NASA/Ames, http://www.nasa.gov/centers/ames/multimedia/images/2006/genebox_prt.htm (rechts, abgerufen am 11.04.2017)

Dem Computer ins Hirn geschaut

Informatik entdecken, verstehen und querdenken

Zitzler, E.

2017, XVI, 460 S. 271 Abb. Book + eBook., Softcover

ISBN: 978-3-662-53665-0