Waldemar Dege

EDV - Maschinelles Rechnen

1971 Urania-Verlag Leipzig / Jena / Berlin
MSB: Nr. 57
Abschrift und LaTex-Satz: 2023 https://mathematikalpha.de

Inhaltsverzeichnis

1 Die neue Maschine

1.1 Der große Aufwand

Mit ein wenig Humor kann man sich den Weg, den die Entwicklung der menschlichen Gesellschaft genommen hat, rechts und links von Maschinenwracks gesäumt denken.
Am Anfang liegen sie als unscheinbare Steine am Weg. Sehr bald finden wir die ersten Werkzeuge, wenig später das Rad und »gleich« danach die ersten muskelgetriebenen Maschinen. Gehen wir weiter, sehen wir die Reste der alten Dampfmaschine, den ersten Stromerzeuger, das erste Auto, das Flugzeug in der Ausführung schwerfälliger Doppeldecker und als Überschallmaschine. Damit sind wir aber schon in der Gegenwart.

Unser Weg ist inzwischen eine breite Schnellstraße geworden, und da entdecken wir am Rand dieser Straße ein seltsames Gewirr von Drähten, Röhren und sonstigen elektrischen und elektronischen Bauteilen, was gar nicht recht in das Bild hineinpasst, das wir uns bisher von einer Maschine gemacht haben:
Wir vermissen Hebel, Räder und sonstige bewegte Teile, die nun einmal zu einer Maschine gehören. Auch sehen wir nicht, was diese Maschine bearbeitet oder hergestellt haben könnte. Ein wenig nachdenklich stehen wir vor dem ersten Modell einer programmgesteuerten elektronischen Rechenmaschine.

Von den ersten Stufen der Menschheitsgeschichte bis zur Gesellschaftsordnung des Sozialismus unserer Tage zeichnet die Menschen das Bemühen aus, sich Werkzeuge zu schaffen, die es ihnen ermöglichen, ihre Bedürfnisse immer besser zu befriedigen. Das ständige Ringen des Menschen um die Vervollkommnung der Produktionsinstrumente führte nach objektiven Gesetzen der gesellschaftlichen Entwicklung von den Anfängen menschlicher Einflussnahme auf die Umwelt über die Dampfmaschine, dem typischen Werkzeug der industriellen Revolution, bis zum Computer, den man getrost als ein Produktionsinstrument der wissenschaftlich-technischen Revolution der Gegenwart bezeichnen kann.

Wodurch unterscheidet sich dieses Instrument von den herkömmlichen Maschinen, die wir kennen?
Die ersten Maschinen, die sich der Mensch gebaut hatte, verarbeiteten irgendwelche Stoffe oder Materialien; sie gaben ihnen neue Formen oder verwandelten diese Materialien in andere Stoffe. Als Beispiele mögen die Töpferscheibe und der Webstuhl genannt sein.

Mit der Dampfmaschine traten dann die Energiemaschinen auf den Plan. Sie wandelten Energie aus einer Form in die andere um. Wir finden sie heute als riesige Turbinen und Generatoren in modernen Großkraftwerken.
Die neue Maschine, die im folgenden untersucht werden soll und die durch den elektronischen Rechenautomaten verkörpert wird, werden wir als informationsverarbeitende Maschine verstehen lernen. Wir werden erkennen, dass die Information eine ähnlich fassbare Größe ist wie die Energie.

Informationen werden aber immer dort benötigt, wo gewisse Steuerprozesse ablaufen, sei es im Gehirn des Tieres oder des Menschen, in einem Betrieb, einem Kombinat oder in der weitverzweigten Volkswirtschaft. Diese Tatsache deutet darauf hin, dass das wichtigste Anwendungsgebiet der informationsverarbeitenden Maschinen in der theoretischen und praktischen Beherrschung industrieller und gesellschaftlicher Steuerungsprozesse liegen wird.

Von einem anderen Standpunkt aus gesehen, zeigen uns die neuen Maschinen einen weiteren Unterschied zu den bisherigen Produktionsinstrumenten. Alle vorhergehenden Maschinen verstärkten die physischen Kräfte des Menschen.
Das Rad erhöhte die Fortbewegungsgeschwindigkeit, die Werkzeugmaschine vervielfachte die Fertigkeit der menschlichen Hand, Dinge zu verformen und zu gestalten, die Kraftwerke stellten schließlich eine überdimensionale Vergrößerung der Muskelkraft dar. In eben diesem Sinne schicken sich die neuen Maschinen an, die geistigen Fähigkeiten des Menschen zu erweitern.

Wenn man bedenkt, in welchem Ausmaß sie das heute schon tun, obwohl sie erst seit etwa 25 Jahren existieren, darf man von ihrer weiteren Entwicklung einen gewaltigen Beitrag zum Fortschritt der gesellschaftlichen Produktivkräfte erwarten.

In einem zeichnet sich diese Maschine gegenüber allen bisherigen Maschinen sehr deutlich aus: Sie verlangt von ihrem Bediener ein Wissen, das die alten Maschinen in solcher Tiefe und Breite nie gefordert haben. Wenn man zum Verständnis ihrer Arbeitsweise und ihrer Einsatzmöglichkeiten gelangen will, muss man sich notgedrungen durch den nicht immer süßen Brei verschiedener Wissensgebiete durchessen.

Wir müssen erfahren, was man unter dem Begriff der Programmsteuerung versteht, was ein Algorithmus ist, warum diese Maschinen auch logische Maschinen heißen, wieso sie rechnen können und auf welche Weise sie das tun. Wir müssen wissen, dass es Analogrechner und Digitalrechner gibt, und wir müssen uns mit den Grundlagen der Programmierung befassen.

Des weiteren muss uns klar werden, dass zu einem sinnvollen, nutzbringenden Einsatz der Anlagen Kenntnisse der Operationsforschung, der Kybernetik und der marxistisch-leninistischen Organisationswissenschaft benötigt werden. Schließlich verlangen ihr Betrieb und ihre Wartung umfangreiche technisch-physikalische und elektronische Kenntnisse.

Im Vergleich zu anderen Maschinen ist die Vorbereitungszeit für den Einsatz elektronischer Rechenautomaten außerordentlich lang.
Ehe der geringste Nutzen durch sie erzielt werden kann, ist eine Vielzahl zum Teil komplizierter und langwieriger Arbeiten zu erledigen, die sich über Jahre erstrecken können. Der Einsatz informationsverarbeitender Maschinen ist also mit hohen gesellschaftlichen Aufwendungen verbunden.

1.2 Scheu vor Robotern unbegründet

An dieser Stelle kann man nun fragen, ob sich ein derartiger Aufwand zur Beherrschung dieser Automaten lohnt, zumal man vielleicht davon gehört haben mag, dass sie angeblich menschenfeindliche Roboter seien, die bei »falscher Programmierung« ihre Konstrukteure verschlingen.
Bedenkt man nun, mit welcher stürmischen Geschwindigkeit sich die neuartigen Automaten vermehren, so ist es durchaus verständlich, dass die Frage gestellt wird, ob die Roboter gefürchtet werden müssen. Mit anderen Worten:
Wie fühlt sich der Mensch in unserer Zeit, die gewissermaßen mit Automaten möbliert ist? Fühlt er sich in seiner Wohnung heimisch, oder erdrücken ihn die sonderbaren Möbelstücke?

Die Gestaltung des entwickelten gesellschaftlichen Systems des Sozialismus in der DDR unter den Bedingungen der wissenschaftlich-technischen Revolution verlangt die Wissenschaft als Produktivkraft.
Der Mensch in unserer Gesellschaftsordnung muss nicht nur mit der Wissenschaft leben und mit ihr »auskommen«, er muss sie beherrschen lernen und sie wie ein Werkzeug zu seinem Nutzen handhaben.

Auch unter kapitalistischen Verhältnissen ist die wissenschaftlichtechnische Revolution wirksam und beeinflusst das materielle, soziale und geistige Leben der Menschen. Aus der Perspektivlosigkeit dieser Gesellschaft entspringt die Idee, dass die modernen Automaten dem Menschen nicht nur die Existenzbedingungen rauben, sondern sich auch mehr und mehr zu Beherrschern des Menschen aufschwingen.
Ein Automat könnte sogar eine Regierung ersetzen, und man benötigte nur einen Portier, der darauf achtet, dass niemand den Stecker aus der Steckdose zieht, damit die Maschine außer Betrieb setzt und so die Regierung zum Rücktritt zwingt.

Es besteht kein Zweifel daran, dass der Mensch mit der weiteren Entwicklung informationsverarbeitender Maschinen und ihrer Nutzung ständig mehr Macht über die Gesetze von Natur und Gesellschaft gewinnt; aber es kommt immer darauf an, für welche Interessen und für welche Ziele die Errungenschaften der Wissenschaft genutzt werden.

Stets erweisen sich die angeblichen Nachteile des automatisierten Zeitalters als Nachteile einer bestimmten Gesellschaftsstruktur im automatisierten Zeitalter. Jedes Werkzeug dient demjenigen, der es besitzt und verwendet. Auch die programmgesteuerten Rechenautomaten sind Produktionsinstrumente, die unter kapitalistischen Bedingungen der kleinen, aber mächtigen Schicht Besitzender gehören, die am deutlichsten durch die Aktionäre der großen, international verflochtenen Konzerne verkörpert werden.

Auf dem »Computermarkt« ist die Kapitalkonzentration und die Machtanmaßung einiger weniger Großunternehmen besonders erkennbar.
Die informationsverarbeitenden Maschinen dienen wie alle Maschinen in dieser Gesellschaftsordnung den Interessen der Unternehmer und sind Mittel zur Vergrößerung ihres Profits. Das spüren insbesondere die untergeordneten Angestellten und mit aller Härte die Arbeiter, wenn ihre Arbeitsplätze durch die Anschaffung programmgesteuerter Maschinen überflüssig werden.

Die Automaten sind dafür nicht verantwortlich zu machen. Ebenso sind es nicht die Computer, die z. B. jeden Einwohner Westdeutschlands zu einer leicht überwachbaren Nummer in der Bundesdatenbank des dortigen Innenministeriums degradieren.
Unter sozialistischen Bedingungen kann es eine Furcht vor der künftigen weitgehenden Automatisierung nicht geben, weil die Ziele der Wissenschaften mit den Bedürfnissen der werktätigen Schichten übereinstimmen. Wir haben uns die Aufgabe gestellt, unsere Kräfte und Hilfsmittel, zu denen die Rechenautomaten gehören, so einzusetzen, dass wir ständig den größtmöglichen Zuwachs an unserem gesellschaftlichen Reichtum, an unserem Nationaleinkommen erzielen. Dabei wird der einzelne in der Gesellschaft nicht vergessen.
Langfristige Qualifizierungsmaßnahmen bereiten ihn auf die Übernahme neuer und verantwortungsvoller Aufgaben vor; in ausführlichen Gesprächen wird seine Neigung für andere Arbeiten festgestellt. Eine Tendenz tritt dabei deutlich zutage:
An die Stelle des körperlich schwer arbeitenden tritt immer mehr der qualifizierte, Automaten bedienende Mensch. An die Stelle von Routinetätigkeiten tritt schöpferische Arbeit. Viele alte Berufe verschwinden völlig, viele Berufe entstehen neu.

Neben den Berufen, die direkt an den Rechenautomaten gebunden sind, wie Programmierer, Operateur oder Wartungsingenieur, entstanden auch Berufsbilder am Rande der Rechenmaschinen. Der Problemanalytiker sucht und formuliert Probleme, die einer elektrischen Lösung zugänglich gemacht werden sollen. Der Systemorganisator entwirft komplexe Lösungen für umfangreiche, auf die Rechenanlage zu übertragende Aufgabengebiete, vor allem im Bereich der sozialistischen Wirtschaft.
Der technische Rechner, der Probeberechnungen, kleine Programmierungsarbeiten und Auswertungen vorzunehmen hat, erhält in der DDR bereits einen Facharbeiterbrief. Daneben kann man auf einer Fachschule zum Datenverarbeiter ausgebildet werden.

Dass bei uns in vielen Rechenzentren bereits mathematisch begabte Schüler der Oberstufe zusätzlich mit den Problemen der Rechenautomaten vertraut gemacht werden, wo sie ohne jede Scheu auf den Maschinen selbständig kleine Aufgaben lösen, belegt, wie wenig Grund zur Furcht vor den »Elektronengehirnen« besteht.

Es ist geradezu verblüffend, wie vor allem junge Menschen von den neuen Maschinen angezogen werden. In verschiedenen Betrieben der DDR wurden die Vorbereitungsarbeiten für den Einsatz der Datenverarbeitungsanlagen zu Jugendobjekten erklärt.

Uns reizt ein Gegenstand, der Gelegenheit gibt, neue Erkenntnisse zu sammeln und anzuwenden, unsere Kräfte voll einzusetzen und die Anforderungen an uns ständig zu erhöhen. Wir setzen den Rechenautomaten dort ein, wo er am meisten nützt, und diktieren ihm die Aufgaben, die am dringlichsten zu lösen sind. Wenn er Ergebnisse liefert, mit denen wir nicht einverstanden sein können, so haben wir die Aufgabe falsch formuliert. Der Automat zwingt uns nicht, unsinnige Ergebnisse anzuwenden.

1.3 Wer ist Herr, und wer ist Knecht?

Ähnlich wie mit dem »Fürchten« verhält es sich mit dem Problem, das in dem Wörtchen »Bedienen« zum Ausdruck kommt. Man trifft in Ausführungen über Rechenautomaten verschiedentlich auf den Ausdruck »der den Automaten bedienende Mensch«.
Entwürdigt sich der Mensch, wenn er den Automaten bedient? Dient er wirklich dem Automaten oder wiederum den Interessen einer Gruppe von Menschen, denen der Automat gehört? Oder dient er sich selbst, wenn unter sozialistischen Verhältnissen die Produktionsmittel gesellschaftliches Eigentum geworden sind?

Der Automat schafft keine zusätzliche, spezifisch automatenbedingte Entwürdigungsform des Menschen. Er setzt höchstens die Reihe der Entwürdigungen des Menschen in der kapitalistischen Gesellschaftsordnung fort.
Unter sozialistischen Bedingungen steht das Problem der »Unterdrückung des Menschen« durch den Automaten nicht. Hier beginnt vielmehr eine Zeit, in der wir immer leistungsfähigere Automaten in nie gekanntem Ausmaß zu unserem Nutzen bauen und beherrschen werden.

Oft scheint es, als ob der Mensch selbst dieser Entwicklung hinderlich im Wege stehe. Als Beispiel soll hier die deutsche Sprache angeführt sein. Wir sprechen aus historischen Gründen keineswegs so, wie wir schreiben. Niemand hört einen Unterschied in der Aussprache des i in den Worten »wir«, »ihr«, »sie«, und doch erfährt dieser Laut drei verschiedene Schreibweisen.
Man sieht ein, dass dies bei der Vollautomatisierung des Vorgangs Diktieren-Schreiben große Schwierigkeiten bereiten wird. Aus diesem Grunde gibt es Bestrebungen, eine Lautschrift der deutschen Sprache zu entwickeln, bei der die Entsprechung zwischen Laut und Schriftzeichen umkehrbar eindeutig ist. Wir erhalten eine solche Lautschrift, indem wir gewissen Buchstaben eine neue Bedeutung geben.

Alt	Neu
C	Sch
Q	NG
V	offenes O
X	CH
Y	offenes Ö
Z	stimmhaftes S

Alle anderen Buchstaben behalten ihre frühere Aussprache. Dehnungen werden prinzipiell durch doppelte Vokale ausgedrückt. Alle Wörter sollen mit kleinem Anfangsbuchstaben geschrieben werden.

ain in deer lautcrift ferfasster täkst wirkt zumindest auf deen eersten blikk ungewoont. die fertraute und in fiilen jaarhunderten entctandenä vrtografii wird mit ainem clag abgecafft.

Unwillkürlich wehrt sich jeder dagegen, auf solcherlei Art zu schreiben und erweist sich so als ein hemmendes Element bei der weiteren Durchführung der Automatisierung.
Demnach wäre der Mensch ein unvollkommener Automat, der unnötige Mehrkosten bei der Vollautomatisierung des gesellschaftlichen Lebens verursacht? Es scheint, dass wir diese Frage bejahen müssen, doch sollten wir das Folgende bedenken.

Unsere Sprache ist historisch entstanden und hat sich auf der Grundlage der gesellschaftlichen Veränderungen als das wichtigste Kommunikationsmittel entwickelt. Die ABC-Schützen, die sich in der Schule in ihrem Gebrauch üben, erlernen das Produkt einer vieltausenjährigen Kulturgeschichte. Die weltverändernden Gedanken von Karl Marx wurden in eben dieser »nichtautomatengerechten« Schrift festgehalten. Hinderte das ihre Durchsetzung?
Der Mensch ist in der Tat kein vollkommener Automat. Er hat ein Bewusstsein seiner Herkunft, seiner Entwicklung. Der Mensch macht sich ein Bild von seiner Zukunft, von der er offensichtlich keine Informationen erhalten kann, weil die Zukunft keine gesellschaftliche Praxis, keine Quelle von verarbeitbaren Daten ist. Der Mensch hat Bedürfnisse, die historisch bestimmt und sozial determiniert sind und die den Inhalt seiner Ziele ausmachen.

Der informationsverarbeitende Automat ist dem Menschen behilflich, indem er dessen Fähigkeiten und Fertigkeiten gewaltig vergrößert; aber welche Bedürfnisse der Mensch hat, für deren Befriedigung er überhaupt tätig ist, kann der Automat von sich aus weder wissen noch erraten.

Zweifellos erfordert die Einführung moderner Datenverarbeitungsanlagen in vielen gesellschaftlichen Bereichen ein neues Durchdenken altgewohnter Tatsachen und eine wissenschaftlich exakte, zum großen Teil mathematische Betrachtungsweise von Zusammenhängen. Auf das Beispiel von der deutschen Sprache angewandt, bedeutet dies, dass bei einer Rechtschreibreform die Aspekte der Automatisierung nicht ohne Einfluss bleiben können, dass sie aber nicht die Automatenreife der technischen Lautschrift erzwingen werden.
Damit stehen wir wieder vor der Frage, ob sich der Mensch den Automaten unterordnet. Bis jetzt haben wir diese Frage verneint, nun müssen wir sie anscheinend mit »ja« beantworten. Wir geben also zu, dass sich der Mensch Dingen unterwirft, die er selbst geschaffen hat.

In gewisser Weise aber tut er das schon jahrhundertelang. Wir gehen gewöhnlich auf Straßen und nicht über Dächer. In den Wohnungen benutzen wir größtenteils die Tür als Ein- und Ausgang und nicht das Fenster. Der Automat legt uns also keine prinzipielle neue Art von Beschränkungen auf. Wenn er Bedingungen stellt, die ihm ein sinnvolles Arbeiten ermöglichen, so stellt er sie letztlich einzig und allein zum Nutzen seines Erfinders, des Menschen; dieser muss freilich die gesellschaftlichen Bedingungen so einrichten, dass er in den vollen Genuss aller Vorteile gelangen kann, die durch die modernen Rechenautomaten möglich geworden sind.

1.4 Der große Nutzen

In der DDR gilt unsere Sorge der schnellstmöglichen Verbreitung und der maximalen Auslastung dieser Anlagen, weil sie einen unschätzbaren Beitrag zur weiteren Entwicklung der Volkswirtschaft leisten.

Der materielle Produktionsprozess ist zunehmend komplizierter geworden: Genügte in einem kleinen Handwerksbetrieb ein grober gedanklicher Entwurf zur Anfertigung des Produktes, müssen in einem Großbetrieb des Maschinenbaus viele Detailzeichnungen, Berechnungen und Hunderte von ausführlichen Arbeitspapieren geschaffen werden, ehe mit der Fertigung der Einzelteile für das entstehende Enderzeugnis begonnen werden kann.
Darüber hinaus ist ein exaktes Wissen über den Stand der Produktion, über die ökonomische Situation des Werkes und über die Kosten, die jedes Produkt hervorruft, für eine wissenschaftliche Planungs- und Leitungstätigkeit unerlässlich.

Diese Tatsachen sowie die zunehmende Verflechtung der Volkswirtschaft stellen die Leiter großer volkseigener Betriebe und Kombinate vor Aufgaben, die nur noch mit Hilfe der elektronischen Datenverarbeitung gelöst werden können.

Während früher die wichtigsten Verlustquellen eines Betriebes in den Werkstätten lagen, müssen sie heute in schlechter Produktionsorganisation, mangelhafter Planungstätigkeit oder falschen Leitungsentscheidungen gesucht werden. Umgekehrt ist natürlich das erzielte Betriebsergebnis in seiner Gesamtheit um so größer, je höher das Niveau von Planung und Leitung ist. Die Informationsverarbeitung wird damit zu einer der wichtigsten Voraussetzungen für die weitere schnelle Entwicklung unserer Volkswirtschaft.

Bis zur Gegenwart galten der Grad der Elektrifizierung und der Grad der Mechanisierung der Produktion als die wichtigsten Kennziffern, die das ökonomisch-technische Entwicklungsniveau eines Landes charakterisierten.
Heute muss man eine weitere Kenngröße beachten, wenn man dieses ökonomisch-technische Niveau real einschätzen will: den Grad der Einführung der elektronischen

Rechentechnik in die Planungs- und Leitungstätigkeit. Es ist sogar möglich, dass diese Kennziffer in naher Zukunft in ihrer Bedeutung an die erste Stelle rücken wird.

Die schnelle und wirkungsvolle Einführung der elektronischen Datenverarbeitung ist damit zu einer wichtigen Aufgabe im Wettbewerb mit dem kapitalistischen Wirtschaftssystem geworden. Die Regierung der DDR hat das klar erkannt: Gegenwärtig gelangt eine hohe Anzahl von Datenverarbeitungsanlagen zum Einsatz.
Neben modernen Großanlagen aus der UdSSR und anderen Ländern handelt es sich vor allem um die in der DDR entwickelten und produzierten mittleren Datenverarbeitungsanlagen vom Typ Robotron 300.

Die Maschinen werden in erster Linie an den Ursprungsorten unseres Nationaleinkommens, in den Betrieben, installiert. Dort werden sich ihre Aufgaben nicht in der Mechanisierung von Abrechnungsarbeiten beschränken. Sie müssen vielmehr den Planungs- und Leitungsprozess der Betriebe verbessern und modernen Technologien zum Durchbruch verhelfen; sie müssen letzten Endes für eine Erhöhung der Betriebsergebnisse sorgen.
Die Rechenanlagen sollen also - wenn man ein wenig übertreibt - nicht rechnen, sondern in der Hand der Wirtschaftsfunktionäre Mittel zur Planung und Leitung des Betriebsgeschehens sein.

Elektronische Datenverarbeitungsanlagen werden in der DDR aber nicht nur in den Produktionsbetrieben aufgestellt. Technisch-wissenschaftliche Großrechner werden an Schwerpunkten der Forschung in Hochschulen und Instituten ihre Arbeit genauso aufnehmen wie mittlere Datenverarbeitungsanlagen in Krankenhäusern. Staatliche Rechenzentralen werden die Entscheidungen in der Planung der gesamten Volkswirtschaft auf ein noch höheres Niveau heben.

Mit der zunehmenden Verflechtung der gesamten Volkswirtschaft, mit der weitergehenden Arbeitsteilung wachsen die Anforderungen an die Leitungs-, Informations- und Kontrollprozesse. Damit ergibt sich zwangsläufig die Notwendigkeit, auch immer mehr und bessere Datenverarbeitungsanlagen einzusetzen.
Je höher der Automatisierungsgrad eines Landes, je höher sein industrielles, technisches Niveau ist, um so höher wird der Bedarf an programmgesteuerten Anlagen sein, und immer stehen hinter jedem Einsatz programmgesteuerter Anlagen gewaltige Steigerungen der Produktivität und neue Möglichkeiten der Entfaltung sozialistischer Persönlichkeiten.

Wir hatten am Anfang auf den großen Aufwand hingewiesen, den die Rechenautomaten für ihren rationellen Einsatz erfordern. Sicher kann man nun den großen Nutzen ahnen, den diese Maschinen in sich bergen:
Sie werden in alle Lebensbereiche eindringen und dazu beitragen, dass das große Vorhaben des entwickelten gesellschaftlichen Systems des Sozialismus in der DDR noch schneller verwirklicht werden kann, dass sich Wissenschaft, Wirtschaft und Kultur noch stürmischer entwickeln werden als bisher.

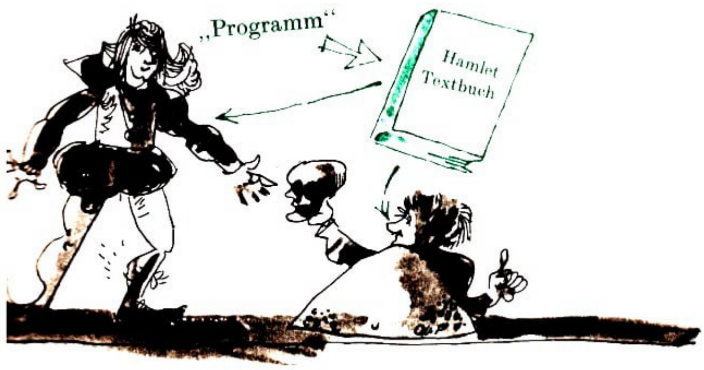
Auf Grund welcher Eigenschaften und Fähigkeiten der elektronischen Datenverarbei-

tungsanlagen diese Erwartung zu Recht besteht und wie sie erfüllt werden kann, soll nun in groben Zügen angedeutet werden.
Wir wollen uns zunächst mit dem wichtigen Begriff der Programmsteuerung befassen.

2 Über die Bedeutung von Vorschriften

2.1 Ein Theaterbesuch

Wenn wir im Theater sitzen und ein Stück von Shakespeare oder Schiller verfolgen, so ist uns kaum bewusst, dass der Handlungsablauf bereits in allen Einzelheiten festgelegt ist. Nach einem bestimmten Programm, das in der Regel mit dem Manuskript des Autors übereinstimmt, verkörpern die Schauspieler ihre Rollen.
Das »Programm« des Textbuches steuert die Bewegungen der Spieler. Zweierlei soll dabei als wichtig hervorgehoben werden.



Einmal muss der Schauspieler Text und Bewegungsfolge auswendig wissen; sie müssen in seinem Gedächtnis gespeichert sein. Zum anderen ist ein Eingriff des Publikums in dem Sinne, dass ein anderer Verlauf des Stückes gewünscht wird, während der Aufführung nicht üblich.
Aus dieser Sicht zeigt uns das Theater deutlich automatenhafte Züge, und es wundert uns daher nicht, wenn wir erfahren, dass es vor allem im 18. Jahrhundert regelrechte Theaterautomaten gab, auf denen zwar sehr kurzdauernde, aber immerhin vollständige Aufführungen abliefen.

In Wien gibt es ein einzigartiges Museum, in dem viele erhalten gebliebene Kunstautomaten aufs sorgfältigste bewahrt und gepflegt werden. Großen Publikumserfolg hat im Wiener Museum zum Beispiel ein Theaterautomat, auf dem eine »sportliche« Veranstaltung zu sehen ist. Er entstammt der Werkstatt der Familie Kraus und wird Ritterspieluhr genannt.
In einem Raum von etwa 10 Kubikdezimetern wird ein vollständiges Ritterturnier mit vielen Beteiligten, mit Siegern und Verlierern vorgeführt. Man stürzt vom Pferd, man verliert Helme, es zerbrechen Lanzen. Nach der Vorstellung genügt es, das Uhrwerk erneut aufzuziehen, um die ganze Herrlichkeit noch einmal zu erleben.

Bei diesem Beispiel wollen wir die Wiederholbarkeit des Vorgangs als wichtig im Gedächtnis behalten. Es liefen gewisse Ereignisse und Handlungen nach bestimmten, festen Vorschriften oder, wie wir auch sagen wollen, nach einem vorgegebenen »Programm« ab.

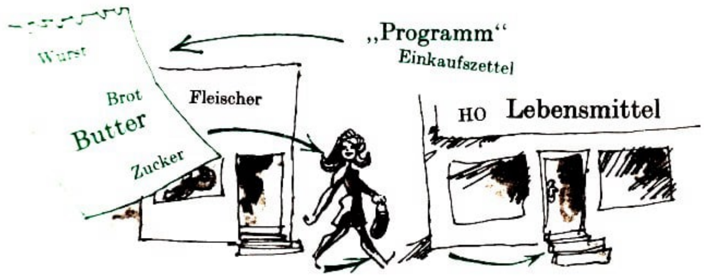
2.2 Alles verläuft programmgemäß

In der Umgangssprache formulieren wir den Ausdruck, »es ist programmgemäß verlaufen«, wenn wir sagen wollen, dass etwas ganz nach unseren Erwartungen gelungen ist. In diesem Falle hatten wir das »Programm« des Geschehenen zuvor in unserer Vorstellung ausgearbeitet, d. h. eine Folge von Handlungen im voraus festgelegt. Danach führten wir dann hintereinander alle diese Handlungen bis zum gewünschten Ergebnis aus.

Einen guten Teil des Bedeutungsinhaltes des in diesem Sinne verstandenen Wortes Programm wollen wir nun in den neuen Begriff »Programmsteuerung« übernehmen.

Was eine Steuerung ist, scheint uns bekannt zu sein. Denken wir beispielsweise an den Flug einer unbemannten Sonde zu unserem Nachbarplaneten, der Venus, so leuchtet uns unmittelbar ein, dass dies, vom Antriebsmechanismus der Rakete her gesehen, ein bloßes Befolgen der auf der Erde gegebenen Kursbefehle bedeutet:
Der Flugkörper wird ferngelenkt oder ferngesteuert. Wenn wir festhalten, dass die Steuerung im wesentlichen eine Ausführung irgendwelcher Vorschriften bedeutet und dass ein Programm eine Sammlung solcher Vorschriften ist, so haben wir fürs erste genug zusammengetragen, um eine vorläufige Vorstellung vom Begriff Programmsteuerung zu erhalten. Wir wollen unter diesem Ausdruck eine Steuerung gewisser Handlungsabläufe durch ein bestimmtes, vorgegebenes Programm verstehen.

Das Programm muss dabei immer auf irgendein entsprechend geeignetes Medium geschrieben sein. Es muss gespeichert vorliegen.
Die Speicher selbst können aus unterschiedlichen Stoffen besehen. Das Schreiben auf diese Speicher kann in einigen Fällen Durchaus wörtlich genommen werden.



Wenn jemand viele Besorgungen zu erledigen hat, wird er sie zweckmäßigerweise auf einen Zettel notieren. Wenn er dann die entsprechenden Stellen aufsucht oder die aufgeschriebenen Waren einkauft, steht er in der Befehlsgewalt des auf dem Papier stehenden Programms. Hierbei ist offenbar ein anderer Speicher, das Gehirn, verschmäht worden, da er für diese Zwecke als unsicher erschien. Wir werden später noch weitere Speicher kennenlernen und dabei feststellen, dass sie die materielle Voraussetzung für die Programmsteuerung darstellen.
Zunächst sei aber noch einiges mehr getan, um die Programmsteuerung als den wichtigsten Begriff der elektronischen Datenverarbeitung zu erkennen.

Ein gewöhnlicher Einkaufszettel lässt uns viele Freiheiten. Er schreibt gewöhnlich nichts über die Reihenfolge der Besorgungen vor und verbietet uns nicht, Dinge zu kaufen,

die nicht auf ihm verzeichnet sind. Wir können ihn aber wesentlich ausführlicher und strenger gestalten, wenn wir auf ihm die Reihenfolge der Läden vermerken und angeben, was wir tun müssen, wenn ein gewünschter Artikel nicht vorhanden ist oder wenn ein aufzusuchendes Geschäft geschlossen hat.
Führen wir diesen Gedanken weiter fort, so gelangen wir schließlich zu einem detaillierten Einkaufsplan, der mit zahlreichen einfachen Entscheidungen versehen ist, eine Fülle einzelner Anweisungen enthält und der so einer gedachten Einkaufsmaschine übergeben und von dieser Punkt für Punkt erfüllt werden könnte.

Wenn wir im folgenden weiter von Programmen sprechen, so sollen darunter Vorschriften gemeint sein, die in diesem Sinne eindeutig und genau sind: Sie sollen bei der Ausführung keine Willkürlichkeiten, keine Abweichungen vom »Text« des Programms zulassen.

2.3 Eine neue Art zu lernen

Gegenwärtig kann man viel vom programmierten Lehrbuch hören. Was verbirgt sich hinter diesem Ausdruck?
Das bisher übliche Lehrbuch reihte im wesentlichen den zu vermittelnden Stoff hintereinander auf. Es setzte also stets einen »genormten« Schüler voraus, der die Lehrsätze Zeile um Zeile gleichmäßig aufnahm und der nichts vom Vorangegangenen vergaß. Wir wissen, dass es einen solchen Schüler nur selten gibt.

Das programmierte Lehrbuch berücksichtigt diese Tatsache, indem es Testfragen stellt, Wiederholungen anordnet, zusätzliche Aufgaben erteilt und sich so der Lerngeschwindigkeit des Lesers besser anpasst. Aber immer wird der Schüler durch den Text des Buches geleitet und von diesem Text zu Wiederholungen und zum Nachdenken gezwungen.

Wir wollen das in stark vereinfachter Weise demonstrieren:

(1) Glauben Sie, dass Ihnen der Begriff der Programmsteuerung einleuchtet? Wenn nein, so lesen Sie bitte jetzt noch einmal die Abschnitte »Ein Theaterbesuch« und »Alles verläuft programmgemäß« dieses Kapitels und danach den Punkt (2) dieser Unterweisung. Wenn Sie dagegen mit »Ja« antworten, so fahren Sie bitte mit dem Punkt (3) fort.

(2) Haben Sie nach einem zweiten Lesen der angeführten Abschnitte die Frage (1) nicht positiv beantwortet, so können Sie das vorliegende Buch getrost als schlecht bezeichnen und das Lesen in demselben einstellen.

(3) Ist bisher eine exakte Definition des Begriffs Programmsteuerung gegeben worden? Wenn Sie diese Frage bejahen, so verfahren Sie bitte nach der Anweisung des Punktes (2). Wenn Sie dagegen richtig festgestellt haben, dass bis jetzt keine solche Definition gegeben wurde, so lesen Sie bitte ohne Programmsteuerung weiter.

Fassen wir unsere noch unvollkommenen Ansichten über diesen Begriff zusammen, so können wir die Programmsteuerung einen Zustand nennen, bei dem gewisse, eindeutig festgelegte Operationen in der Reihenfolge, in der sie auf irgendeiner Liste verzeichnet sind, ausgeführt werden.

2.4 Über Gebrauchsanweisungen und Rechenvorschriften

Im folgenden wollen wir diese Liste und die darauf befindlichen Anweisungen oder Befehle näher betrachten.
Eine Gebrauchsanweisung für einen Staubsauger muss so klar abgefasst sein, dass sie jeden Käufer eines solchen Gerätes in die Lage versetzt, dieses zweckgerecht verwenden zu können. Eine derartige Anweisung kann bei komplizierten Geräten sehr lang sein.

Als Beispiel sei die Bauanleitung für ein Fernsehgerät genannt. Aber nie kann diese Bauanleitung unendlich lang sein, und immer zielen derartige Vorschriften auf eine Möglichkeit, etwas in Gang zu setzen oder etwas herzustellen. Sie erfüllen das, es sei noch einmal betont, indem sie alle Einzelheiten, die zur Erreichung dieses Zieles nötig sind, in der richtigen Reihenfolge und in der nötigen Klarheit aufzählen.

Lückenlos lassen sich in ein solches Schema auch alle Rechenvorschriften einfügen. Sie haben das Ziel, etwas zu berechnen.
Jeder Mensch, der die Einzelanweisungen einer solchen Vorschrift versteht und etwas Ausdauer besitzt, ist fähig, eine derartige Aufgabe zu lösen. Die Rechenvorschrift muss natürlich genügend klar abgefasst und mit allen möglichen Alternativen bis in die letzte Einzelheit geplant sein. Weiter wird vorausgesetzt, dass die Liste der Einzelanweisungen aus endlich vielen Befehlen besteht.
Jede in diesem Sinne aufzufassende Vorschrift, deren Ausführung bis in die letzte Einzelheit festgelegt ist, soll ein Algorithmus genannt werden.

Es gibt abbrechende und nicht abbrechende Algorithmen. Ein abbrechender Algorithmus liegt zum Beispiel vor, wenn die Aufgabe gestellt wird, alle ganzen Zahlen von 1 bis 100 niederzuschreiben.
Nach der Niederschrift der Zahl 100 ist die Aufgabe gelöst, und der Algorithmus bricht ab.
Im folgenden sei ein nicht abbrechender Algorithmus angegeben:

1. Setze für den Ausdruck y den Wert 1.
2. Bilde den Wert $x = 0{,}5\left(y + \dfrac{4}{y}\right)$.
3. Setze den eben errechneten Wert von x dem Ausdruck y gleich!
4. Fahre mit Punkt 2 dieser Vorschrift fort.

Es ist leicht einzusehen, dass das Verfahren kein Ende finden kann. Wenn wir uns der Mühe aussetzen und diesen Algorithmus abarbeiten, so stellen wir fest, dass sich die laufenden x-Werte immer mehr der Zahl 2 nähern, ohne sie jemals zu erreichen.
Offensichtlich können wir auf diese Weise die Zahl 2 nie »berechnen«:

	1.Durchlauf	2.Durchlauf	3.Durchlauf	4.Durchlauf	5.Durchlauf
y	1,0	2,50	2,050	2,0010	...
$\frac{4}{y}$	4,0	1,60	1,952	...	...
$4+\frac{4}{y}$	5,0	4,10	4,002	...	...
$x=4+\frac{4}{y}$	2,5	2,05	2,001	...	...

Uns drängt sich die Frage auf, ob es nicht noch mehr Dinge gibt, die sich in dem beschriebenen Sinne nicht berechnen lassen, ob nicht die meisten Dinge »an und für sich« unberechenbar sind.

2.5 Das rettende Epsilon

Wir wollen den Begriff »berechenbar« exakt definieren.
Eine Funktion $f(x)$ soll berechenbar heißen, wenn für alle x, für die diese Funktion erklärt ist, zur Ermittlung des Funktionswertes $f(x)$ ein und derselbe abbrechende Algorithmus angegeben werden kann.
So ist zum Beispiel die Funktion

$$y = 3x + 8$$

in der x alle ganzen Zahlen durchläuft, berechenbar, da der folgende Algorithmus abbricht:

1. Bilde das Dreifache von x.
2. Zähle 8 zum vorigen Ergebnis dazu.

Dagegen ist die Funktion $y = \sqrt{x}$, die für jeden ganzzahligen x-Wert seine Quadratwurzel liefert, nicht berechenbar, weil zur Berechnung von $\sqrt{x}$ kein abbrechender Algorithmus angegeben werden kann.
Die einfachen und viel gebrauchten Funktionen $\sqrt{x}$, $\sin x$ oder e^x besitzen zu ihrer Bestimmung folgende Algorithmen:

1. Wähle einen geeigneten Ausgangswert y_0.
2. Setze diesen Ausgangswert dem Rechenwert y_{alt} gleich.
3. Berechne aus dem alten Wert von y, also aus y_{alt} nach einer speziellen Formel einen neuen Wert von y, der y_{neu} genannt sein soll.
4. Setze diesen neu gewonnenen Wert y_{neu} gleich y_{alt}.
5. Fahre mit Punkt 3 dieser Anweisung fort.

Diese Algorithmen brechen nicht ab. Die berechneten Werte y_{alt} und y_{neu} werden dem wahren Wert y zwar immer näher kommen, aber ihn nie erreichen. Einen solchen Vorgang haben wir bereits bei der scherzhaften Berechnung der Zahl 2 kennengelernt.

Es zeigt sich nun, dass es weit mehr Funktionen gibt, die nicht berechenbar sind, als solche, die man berechnen kann. Die nicht berechenbaren Funktionen lassen sich mit einem unendlich ausgedehnten Ozean vergleichen, aus dem die berechenbaren Funktionen als winzige unscheinbare Inseln herausragen. Es ist eine überaus seltene Ausnahme, eine berechenbare Funktion zu finden.

Hier scheint ein Widerspruch vorzuliegen. Wir wissen, dass die eben als unberechenbar

hingestellten elementaren Funktionen $\sqrt{x}$, $\sin x$ oder e^x in vielen technischen Berechnungen vorkommen.
Schon in der Schule haben wir das Wurzelziehen gelernt.
Bei näherer Betrachtung solcher Berechnungen zeigt es sich aber, dass nie ein abbrechender Algorithmus für die Berechnung der wahren Werte dieser Funktion verwendet wird, sondern immer ein etwas abgewandelter Algorithmus, der die Frage »Ist eine gewünschte Genauigkeit erreicht?« als entscheidende Ergänzung enthält und sich dadurch in einen abbrechenden Algorithmus zur Berechnung eines Näherungswertes verwandelt.

Diese Frage wird meist in folgender Form gestellt: »Ist die Differenz aus dem gegenwärtig berechneten und dem vorhergehenden Näherungswert kleiner als eine gewisse vorgegebene Zahl ε ?«
Diese Zahl Epsilon erweist sich als Retter in der Not. Wird die Frage mit »Ja« beantwortet, so findet das Verfahren sein Ende.

Kein Mensch wird je die Werte aller Dezimalstellen der berühmten Zahl π erfahren. Was er kennt, sind nur Näherungswerte, von denen der bekannteste, $\pi = 3{,}14$, mit zwei Stellen angegeben ist.
Ebenso ließe sich die Zahl mit einer Genauigkeit von 100 oder gar 100000 Stellen niederschreiben, aber nie in ihrer gesamten Ziffernfolge.

Die Frage der Genauigkeit spielt bei den modernen Ziffernrechnern eine wichtige Rolle. Wenn wir dort Funktionen berechnen, so sind die Ergebnisse selten exakt. Es sind fast immer Näherungen, weil der Automat nur über eine beschränkte Stellenzahl für die Darstellung von Zahlen verfügt. Freilich erweisen sich diese 8 bis 12 Stellen für die Praxis als völlig ausreichend.

2.6 Sind Entscheidungen schwieriger als Berechnungen ?

Ein Rechenautomat kann, wenn wir es sehr kurz sagen wollen, nur das berechnen, was berechenbar ist. Berechenbar aber ist alles das, was auf einem abbrechenden Algorithmus beruht. Damit können wir auch erklären, wie die Begriffe Programmsteuerung, Programm und Algorithmus zusammenhängen:
Ein Programm ist nichts weiter als die der Maschine »verständliche« Niederschrift eines abbrechenden Algorithmus. Ein solches Programm steuert die Arbeit der Maschine, indem Schritt für Schritt die Einzelanweisungen des Algorithmus von der Rechenanlage befolgt werden. Während dieser Arbeit befindet sich die Anlage im Zustand der Programmsteuerung.

Wir wissen aber, dass ein Rechenautomat nicht nur Rechnungen ausführt, sondern auch Übersetzungen anfertigt und Produktionsprozesse steuert. Werden hier »höhere« geistige Eigenschaften des Automaten benötigt?

Jeder ist geneigt, diese Frage zu bejahen. Wir vermuten, dass diesen Tätigkeiten kompliziertere als beim einfachen Rechnen benötigte Entscheidungen zugrunde liegen. Dabei stellen wir uns unter einer Entscheidung einen auf hoher geistiger Stufe ablaufenden Prozess vor.

Ein Werkleiter habe zum Beispiel die Frage zu beantworten, ob sein Betrieb einen plötzlich eintreffenden Auftrag annehmen kann. Der Auftrag sei von großer Wichtigkeit für die gesamte Volkswirtschaft.
Das ist gewiss eine schwerwiegende Entscheidung. Aber wie wird sie vorbereitet? Trifft sie der Leiter gewissermaßen aus dem Gefühl heraus?

Der Werkleiter wird zunächst seinem Technischen Direktor oder seinem Produktionsdirektor die Frage vorlegen, wann und wieviel freie Maschinenkapazität zur Verfügung steht. Unter Heranziehung der entsprechenden Mitarbeiter werden die Planunterlagen durchgesehen, der bisherige Produktionsstand berücksichtigt und erwogen, ob etwa andere, unwichtigere Aufträge zugunsten des soeben eingetroffenen zurückgestellt werden können.
Ergeben sich aus dem Umfang der zu diesem Auftrag anfallenden Arbeiten Überlastungen von Maschinen oder Mitarbeitern, so muss geprüft werden, ob Teilaufträge an andere Betriebe weitergeleitet werden können.
Eine Fülle von Einzelentscheidungen und Berechnungen bereiten die endgültige Entscheidung des Werkleiters vor.

Wir haben das Gefühl, dass einer Entscheidung - wenn sie objektiv sein soll - ein Algorithmus zugrunde liegen müsste.

2.7 Was ist eine Entscheidung ?

Wir wollen dem umgangssprachlichen Wort »Entscheidung« einen neuen Sinn geben. Ähnlich wie das Wort »berechenbar« soll es eine ganz bestimmte, definierte Bedeutung erhalten. Zu diesem Zweck müssen vorher einige weitere Begriffe erklärt werden.

Wir wollen unter einem Alphabet eine endliche Anzahl beliebiger Zeichen verstehen und dieses Alphabet mit A bezeichnen. Wir verallgemeinern gewissermaßen das uns allen bekannte Alphabet mit den Buchstaben A bis Z. Auch die Ziffern 0 bis 9 bilden jetzt beispielsweise ein Alphabet.

Unter einem Wort, das aus einem Alphabet A gebildet werden kann, soll eine Folge von Zeichen dieses Alphabets A verstanden werden. In diesem Sinne besteht der vorliegende Satz aus Wörtern, die aus dem Alphabet der deutschen Schriftsprache gebildet werden. Die Zahl 738 ist dagegen ein Wort über dem Alphabet der Ziffern 0 bis 9.

Eine beliebige Anzahl von Wörtern, die aus Zeichen eines gegebenen endlichen Alphabets A bestehen, soll schließlich eine Menge M von Wörtern über einem endlichen Alphabet A genannt werden.
Der Begriff der Menge ist einer der wichtigsten in der Mathematik. Wir bringen lediglich in Erinnerung, dass eine Menge M aus Elementen a besteht, die durch die Beziehung

$$a \in M$$

»a ist Element von M« oder »a ist enthalten in M« miteinander verknüpft werden. Der Charakter dieser Beziehung bestimmt letztlich die Menge selbst; man sagt: »a ist Element von M, genau dann, wenn...« und formuliert hinter diesem »genau dann, wenn« die Eigenschaften der Elemente a, die sie zur Mitgliedschaft in der Menge M berechtigen.

Vorgegeben sei nun eine Menge M4 von Wörtern über einem endlichen Alphabet A. Wir definieren:
Die Menge M ist entscheidbar, wenn es einen abbrechenden Algorithmus gibt, mit dessen Hilfe man feststellen kann, ob ein beliebiges Wort W über dem Alphabet A zur Menge M gehört oder nicht. Ein solcher Algorithmus soll als Entscheidungsverfahren bezeichnet werden.

Nehmen wir die Ziffern von 0 bis 9 als Alphabet A und die Menge aller geraden Zahlen 2, 4, 6,....als Menge M, so ist diese Menge entscheidbar, weil wir von jeder beliebigen ganzen Zahl x sagen können, ob sie gerade ist oder nicht. Der abbrechende Algorithmus, der uns das erlaubt, hat eine sehr einfache Gestalt:

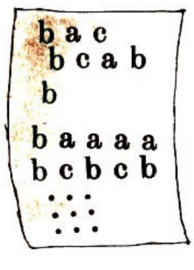
1. Setze den Wert 0 für die unbekannte Hilfsgröße y.
2. Addiere zu y die Zahl 2 und nenne das Ergebnis wieder y.
3. Ist y größer oder gleich der Zahl x?
Wenn nein, so fahre mit Punkt 2 dieser Anweisung fort. Wenn ja, gehe über zu Punkt 4.
4. Ist $x = y$, so gehört x zur Menge der geraden Zahlen. Ist dagegen y größer als x, so gehört x nicht zur Menge der geraden Zahlen.

Wir wollen mit Hilfe dieses Algorithmus entscheiden, ob die Zahl $x = 3$ gerade ist:

1	y	=0	0		
2	$y + 2$	$= y$	2	$\nearrow$4	
3	$y \geq 3$		nein$\nearrow$	ja	
2	$y = 3$			nein	$\rightarrow$ 3 ist ungerade

Vergleichen wir jetzt die Definitionen der Berechenbarkeit und der Entscheidbarkeit miteinander, so ist eine große Ähnlichkeit erkennbar. Sind beide Begriffe am Ende identisch?

Wir wollen uns eine Funktion $F(W)$ vorstellen, die die Eigenschaft hat, dass sie jedem Wort W aus einem Alphabet A eine Zahl y zuordnet. Diese Zuordnung soll so beschaffen sein, dass die Zahl y immer dann den Wert 1 annimmt, wenn das Wort W einer bestimmten Menge M von Wörtern desselben Alphabets angehört.



Beispielsweise sollen die drei Buchstaben a, b, c unser Alphabet bilden:

$$\{a, b, c\} = \text{ Alphabet } A$$

Die Menge M soll aus allen Wörtern des Alphabets A bestehen, die den Anfangsbuchstaben b haben. Diese Menge besteht also aus solchen Wörtern:

Die Funktion $y = F(W)$ soll so beschaffen sein, dass $y = F(W) = 1$ sobald das Wort W mit dem Buchstaben b beginnt; im anderen Fall sollen $y = F(W) = 0$ sein. Diese Funktion lässt sich leicht aufschreiben, wenn man den drei Buchstaben a, b, c folgende Zahlen x zuordnet

	a	b	c
$x =$	0	1	1

und sagt, dass für $F(W)$ der Wert x des Anfangsbuchstabens zu nehmen ist:

$$y = F(W) = x_{\text{Anfangsbuchstabe}}$$

Beispielsweise gilt für die aufgeschriebenen Wörter

Wort	W Anfangsbuchstabe	$F(W) = x_{\text{Anfangsbuchstabe}}$
a b c	a	0
b c a b a	b	1
c a a	c	0
c b	c	0

Die auf diese Weise konstruierte Funktion $F(W)$ ist berechenbar. Zu ihrer Berechnung ist der folgende abbrechende Algorithmus zu verwenden.

Algorithmus 1:
1. Betrachte den Anfangsbuchstaben des Wortes W.
2. Ist dieser ein b, so ist $y = 1$; in allen anderen Fällen ist $y = 0$.

Andererseits ist die Menge M aller Wörter, die mit b beginnen, entscheidbar im Sinne der vorher gegebenen Definition, weil sich sehr leicht ein Algorithmus finden lässt, der gestattet, von jedem Wort W des Alphabets A zu sagen, ob es der Menge M angehört oder nicht.

Algorithmus 2:
1. Betrachte den Anfangsbuchstaben des Wortes W.
2. Ist dieser ein b, so gehört W zur Menge M; in allen anderen Fällen gehört W nicht zur Menge M.
Man kann also in unserem Beispiel sagen, dass M entscheidbar und dass $F(W)$ berechenbar ist.

Wenn wir noch einmal die Algorithmen 1 und 2 ansehen, so müssen wir feststellen, dass beide im Grunde dasselbe aussagen. Das aber hat schwerwiegende Bedeutung. Daraus lässt sich nämlich der wichtige Satz herleiten, dass die Menge M dann und nur dann entscheidbar ist, wenn die oben definierte Funktion $F(W)$ berechenbar ist.
Entscheidbar ist also nur das - wenn man es sehr einfach sagen will -, was berechenbar ist.

Damit ist die große Bedeutung des Algorithmus gezeigt: Nur solche Berechnungs- und Entscheidungsverfahren, denen abbrechende Algorithmen zugrunde liegen, sind so klar und eindeutig formulierbar, dass sie von Maschinen abgearbeitet werden können.

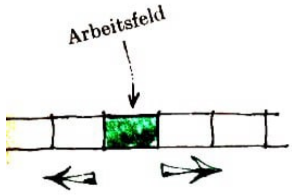
2.8 Eine Maschine erklärt einen Begriff

Im Jahre 1937 versuchte der englische Mathematiker Turing, den Begriff Algorithmus exakt zu fassen und definierte: Ein Algorithmus ist eine Rechenvorschrift, die auf einem speziellen Rechenautomaten ausgeführt werden kann. Diese so einfach formulierte Auffassung erwies sich für die Logik und für die Theorie der Automaten als überaus bedeutungsvoll.

Alan M. Turing, 1912 in London geboren, hatte eine Abhandlung über eine Klasse von mathematischen Problemen veröffentlicht, die nicht durch einen algorithmischen Prozess, d. h. durch eine endliche Folge von Rechenanweisungen gelöst werden können. Ein algorithmischer Prozess kann nach Turing von einem speziell für diesen Prozess gebauten Rechenautomaten ausgeführt werden. Ein solcher gedachter Rechenautomat, der nur in der Lage ist, einige wenige einfache Operationen auszuführen, erhielt nach seinem Erfinder den Namen Turingmaschine.

Die Turingmaschine ist also kein »universeller« Automat, sondern eine Rechenmaschine, die nur jeweils den gerade betrachteten Algorithmus abarbeitet. Damit ist das Problem, den Algorithmus exakt zu definieren, auf die Aufgabe zurückgeführt worden, einen speziellen Automaten genau zu erklären.

Die Konstruktion der Turingmaschine ist einfach. Die »Maschine« besteht im wesentlichen aus dem sogenannten Rechenband, das sowohl zur Eingabe und Ausgabe von Zahlen als auch zu deren Speicherung benutzt wird. Das Band können wir uns als einen langen schmalen, in einzelne Felder unterteilten Papierstreifen vorstellen, der nach beiden Seiten endlos weiterläuft.



Jedes Feld besitzt ein linkes und ein rechtes Nachbarfeld. Die Turingmaschine arbeitet in Arbeitstakten. Zu jedem Takt besitzt die Maschine ein einziges, von den übrigen Feldern unterschiedenes Arbeitsfeld. In jedem Takt kann eine der folgenden Operationen ausgeführt werden.

1. Das Beschreiben des Arbeitsfeldes. Dabei kann es sich sowohl um das Beschriften eines leeren Feldes als auch um das Überschreiben eines bereits gefüllten handeln.

2. Das Weiterrücken nach links. Das ist ein einfaches Verschieben des Bandes um ein Feld nach links, wobei dieses neue Feld zum Arbeitsfeld wird.

3. Das Weiterrücken nach rechts. Diese Operation unterscheidet sich von der vorhergehenden nur durch die Richtung.

4. Das Anhalten.

Diese vier Operationen bilden die einfachste Befehlsliste, die man sich denken kann.

Das Band ist Träger einer Inschrift, die aus Buchstaben eines endlichen Alphabets besteht. Jedes Feld darf höchstens von einem Zeichen beschrieben sein. Zu Beginn der Arbeit trägt das Band eine Anfangsinschrift.
Handelt es sich zum Beispiel um die Turingmaschine zur Berechnung der Quadratwurzel aus x, so ist es am Anfang der Wert x, aus dem die Wurzel gezogen werden soll. Die Tätigkeit der Maschine besteht nun darin, die Anfangsinschrift entsprechend dem Algorithmus in jedem Augenblick systematisch abzuändern und irgendwann anzuhalten. In diesem Augenblick befindet sich dann das Ergebnis auf dem Band.

Mit der Erklärung des Begriffs Algorithmus haben wir den ersten Schritt zum Verständnis programmgesteuerter Rechenmaschinen bereits getan. Programmgesteuerte Rechenautomaten sind prinzipiell in der Lage, beliebige abbrechende Algorithmen abzuarbeiten.
Sie sind aber nicht fähig, mehr als das zu leisten.

Diese Erkenntnis hat wichtige praktische Konsequenzen: Es ist sinnlos, ein Problem einer Rechenmaschine zur Lösung anzubieten, wenn dieses Problem nicht algorithmisch gefasst ist oder gar algorithmisch nicht fassbar ist. Wir dürfen von einer elektronischen Rechenanlage keine Wunderdinge erwarten.

Ein Betrieb, der eine elektronische Datenverarbeitungsanlage installiert hat, wird keinen Nutzen davon haben, wenn er nicht zuvor wenigstens einen Teil der Probleme, die er mit Hilfe dieser Maschine lösen will, in klarer, eindeutiger Form aufbereitet und Lösungsalgorithmen angegeben hat, die - in die »Sprache« der Maschine übersetzt - als »Programme« von ihr abgearbeitet werden können.
Um zu verstehen, wie dies im einzelnen erfolgt, muss man natürlich noch mehr wissen. Im nächsten Kapitel wollen wir uns nun der Logik zuwenden und herausfinden, warum die programmgesteuerten Rechner auch logische Maschinen genannt werden.

3 Der Geist in spanischen Stiefeln

3.1 Die lügenden Kreter

Alle Kreter lügen. Ich bin aus Kreta.
Die Folgerungen aus diesem bekannten Ausspruch scheinen einen Teufelskreis zu bilden und wecken ein tiefes Misstrauen gegen die formale Logik, obwohl gerade diese formale Logik den Teufelskreis an der richtigen Stelle zerschneiden kann.

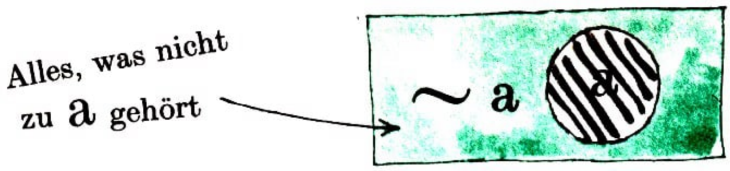
Den zitierten Ausspruch wollen wir als eine Aussage bezeichnen. Dabei wollen wir uns mit der Erklärung des Begriffes Aussage dahingehend begnügen, dass sie eine beliebige Bemerkung, eine Mitteilung oder ein Ausspruch über die objektiv existierende Welt sein soll.
Eine Aussage kann mit der Realität, über die sie etwas aussagt, übereinstimmen oder nicht. Im ersten Fall wollen wir sie wahr nennen und ihr den »Wahrheitswert« W zuordnen; im anderen Fall soll die Aussage falsch heißen und den Wahrheitswert F erhalten. Aussagen sollen im folgenden mit kleinen lateinischen Buchstaben bezeichnet werden.

Aus gegebenen Aussagen kann man durch logische Verknüpfungen neue Aussagen erhalten. Der einfachste Fall besteht in der Verneinung einer Aussage. Liegt eine Aussage a vor, so soll ihre Verneinung mit dem Symbol

$$\sim a$$

das »nicht a« gelesen wird, bezeichnet werden. Das kleine Zeichen »$\sim$« ist demnach der Geist, der stets verneint.

Ist also a die Aussage »es regnet«, so ist $\sim a$ die Aussage »es regnet nicht«.
Für die Verneinung und für alle später noch zu erklärenden logischen Verknüpfungen lassen sich einfache geometrische Modelle angeben. In der folgenden Abbildung sei der Kreis die Aussage a; dann ist die Verneinung dieser Aussage die übrige Fläche des Vierecks, also all das, was nicht zu a gehört:



Ist nun die Aussage a wahr, hat a also den Wahrheitswert W, so ist $\sim a$, d. h. die Verneinung von a, zwangsläufig falsch und umgekehrt, wie das die Tabelle ausdrückt. Andererseits ist $\sim a$ wahr, sobald a falsch ist.

a	$\sim a$
W	F
F	W

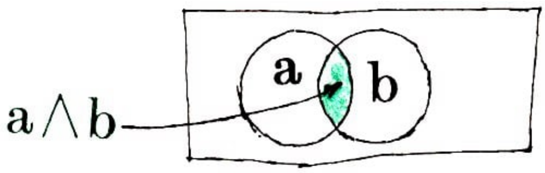
3.2 Es ist kalt, und (oder) es regnet

Sind zwei Aussagen a und b gegeben, so kann man die Aussage »a und b« definieren, die in Zeichen

$$a \wedge b$$

geschrieben wird. Diese logische Operation wird als Konjunktion oder als das logische »und« bezeichnet.

Nehmen wir als Aussage a den Satz »Es ist kalt« und für b die Bemerkung »es regnet«, so erhalten wir als Aussage $a \wedge b$ den Satz »Es ist kalt und es regnet«.
Diese Gesamtaussage kann nur an einem kalten und regnerischen Tage mit der Wirklichkeit übereinstimmen.
Im folgenden Bild sei der eine Kreis die Aussage a und der andere die Aussage b. Dann ist das den beiden Kreisen gemeinsame Flächenstück der Aussage »a und b« gleichzusetzen ($a \wedge b$).



Dieses gemeinsame Flächenstück umfasst also alles das, was sowohl zu a als auch zu b gehört.
Es kann $a \wedge b$ nur dann wahr sein, wenn beide ihrer Teile wahr sind. Sobald eine der Teilaussagen den Wahrheitswert F erhält, hat auch das Ergebnis den Wahrheitswert F. Dieses sei wieder an einer Tabelle dargestellt.

a	b	$a \wedge b$
W	W	W
W	F	F
F	W	F
F	F	F

Denken wir uns an die Stelle des Wahrheitswertes W die Zahl 1 und an Stelle von F die Zahl 0 gesetzt, so stellt die obige Tabelle nichts anderes dar als die Multiplikationstabelle dieser beiden Zahlen.

a	b	$a \cdot b$
1	1	1
1	0	0
0	1	0
0	0	0

Aus diesem Grunde wird die logische Konjunktion auch vielfach als logisches Produkt bezeichnet.

Zwei Aussagen a und b können durch das Wort »oder« verbunden werden und ergeben so die neue Aussage »a oder b«, was in Zeichen

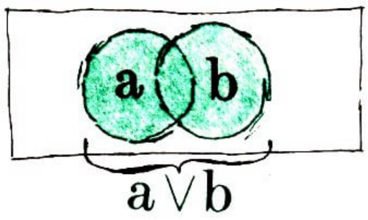
$$a \vee b$$

geschrieben wird. Diese Operation nennt man logische Disjunktion.

Verwenden wir für a und b wieder die Bedeutung »Es ist kalt« und »Es regnet«, so erhalten wir für $a \vee b$ die Aussage »Es ist kalt, oder es regnet«.
Eine solche Aussage macht man vielleicht am Morgen, noch ehe man aus dem Fenster gesehen hat, und meint damit soviel wie:
»Das Wetter ist nicht zum Baden geeignet. Vielleicht ist es kalt, vielleicht regnet es, oder vielleicht ist es kalt und es regnet dazu.«

Die Disjunktion $a \vee b$ ist nur dann falsch, wenn beide Teilaussagen den Wahrheitswert F tragen. In allen anderen Fällen ist sie wahr.
Man muss sich vor Augen halten, dass dieses mit dem Zeichen $\vee$ gemeinte logische »oder« nicht das ausschließliche »entweder - oder« der Umgangssprache ist. Am geometrischen Beispiel wird die Bedeutung dieses »oder« noch deutlicher:

Die beiden Kreise seien wieder den Aussagen a und b gleichgesetzt. Dann ist die neue Aussage »a oder b« das gesamte von beiden Kreisen gebildete Flächenstück. In der Abbildung ist dieses Flächenstück schraffiert



dargestellt; es enthält also alles das, was zu a und zu b gehört, wobei das Wort »und« im Sinne einer Vereinigung oder Summierung aufzufassen ist. »Dies oder das« kann also »beides« sein.

Da die Aussage $a \vee b$ nur dann falsch ist, wenn beide ihrer Teilaussagen falsch sind, lautet die dazugehörige Wertetabelle folgendermaßen:

a	b	$a \vee b$
W	W	W
W	F	W
F	W	W
F	F	F

In den drei Beziehungen W $\vee$ F = W, F $\vee$ W = W und F $\vee$ F = F offenbart sich der Charakter der Addition der beiden Zahlen 0 und 1. Aus diesem Grunde bezeichnet man die Disjunktion mitunter auch als logische Summe.

3.3 Wenn einmal keinmal wäre

Zwei Aussagen a und b lassen sich durch Verwendung der beiden Wörter »wenn« oder »so« zu der Aussage »wenn a, so b« oder »aus a folgt b« kombinieren, was symbolisch

$$a \to b$$

geschrieben wird. Diese Operation nennt man die logische Implikation.
Mit unseren alten Bedeutungen für a und b erhalten wir für $a \to b$ die Aussage »wenn es kalt ist, so regnet es«. Dieser Satz ist nur dann wahr, wenn es auch wirklich regnet. Unwillkürlich ist man geneigt, diesen Satz generell als falsch hinzustellen, weil die Wortverknüpfung »wenn - so« dies in der gewohnten Verwendung als logisch erscheinen lässt. Aber die hier durch den Pfeil definierte logische Verknüpfung hat einen etwas anderen Sinn, der aus folgender Tabelle entnommen werden kann.

Aussage a: Es ist kalt	Aussage b: Es regnet	Aussage: $a \to b$
kalt	Regen	wahr
kalt	niederschlagsfrei	falsch
warm	Regen	wahr
warm	niederschlagsfrei	wahr

Dasselbe spiegelt die folgende Tabelle wider:

a	b	$a \to b$
W	W	W
W	F	F
F	W	W
F	F	W

Diese Tabellen sind anfangs nicht ohne weiteres verständlich.
Wir wollen versuchen, mit der Wortverknüpfung »aus a folgt b« etwas tiefer in die Implikation einzudringen; wir nennen a Voraussetzung und b Folgerung.

Dann kann man sagen: Eine Aussage, in der aus einer richtigen Voraussetzung eine richtige Folgerung gezogen wird, ist richtig (1. Zeile unserer Tabellen).
Eine Aussage, in der dagegen aus einer wahren Voraussetzung eine falsche Folgerung gezogen wird, ist falsch (2. Zeile).

Die 3. Zeile sei an einem weiteren Beispiel erläutert.
Wir setzen für a eine falsche Aussage, beispielsweise den Satz »Einmal ist keinmal«, der ja offensichtlich »$1 = 0$« behauptet und daher falsch ist. Als Aussage b wählen wir die Tatsache »$2 = 2$«. Wir wollen zeigen, dass die Aussage $a \to b$ wahr ist. Dazu zählen wir auf beiden Seiten der »Gleichung«

$$1 = 0$$

die Zahl 2 dazu und erhalten

$$2 + 1 = 2$$

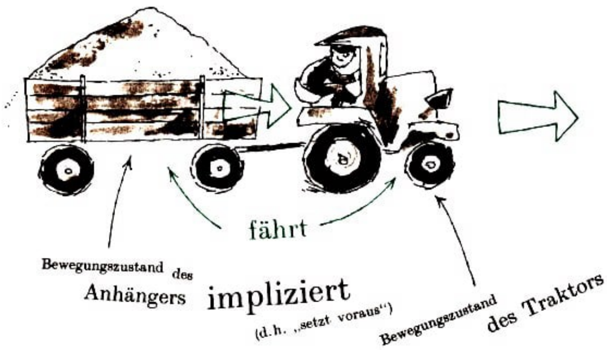
Da aber nach unserer Voraussetzung $1 = 0$ ist, können wir schreiben

$$2 = 2$$

Es beruht also auf Wahrheit, dass man aus einer falschen Voraussetzung einen gültigen Schluss ziehen kann.

Ebenso wahr ist auch, dass man aus einer falschen Voraussetzung etwas Falsches folgern kann (letzte Zeile in den Tabellen). Die Aussage a sei wieder die Behauptung »$1 = 0$« und die Aussage b die Behauptung »$2 = 0$«. Die Aussage $a \to b$ lautet also »Wenn $1 = 0$ ist, so ist auch $2 = 0$«.
Da sich die 2 aus der Operation $1+1 = 2$ ergibt und da jede 1 in dieser Operation nach der Voraussetzung 0 ist, so haben wir auch $0 + 0 = 2$ oder, was dasselbe ist $2 = 0$.
Wenn also einmal keinmal wäre, so wäre auch zweimal keinmal.

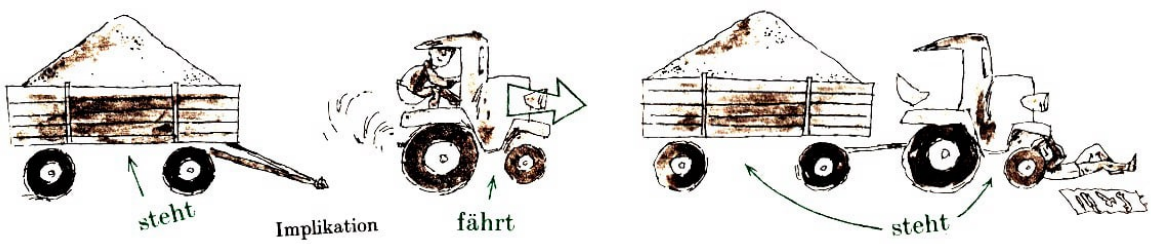
Wir wollen ein weiteres Beispiel für die Implikation mit Hilfe eines Traktors und eines schwer beladenen Anhängers konstruieren.



Dieser Anhänger soll nur vom Traktor bewegt werden können. Wir betrachten nun die beiden »Bewegungs«-Zustände »fahren« und »stehen«. Dann folgt doch zwangsläufig aus dem Bewegungszustand des Anhängers der Bewegungszustand des Traktors; umgekehrt können wir übrigens nicht behaupten, dass aus dem Bewegungszustand des Traktors der Bewegungszustand des Anhängers folgt, da der Traktor auch allein fahren kann. Wir betrachten also die Beziehung.

Bewegungszustand des Anhängers - Bewegungszustand des Traktors.

Diese Beziehung kann doch offensichtlich nur dann nicht mit der Wirklichkeit übereinstimmen, wenn der Traktor steht und vom Anhänger behauptet wird, dass er in Bewegung sei. Wir hatten ja vorausgesetzt, dass der Anhänger nicht ohne Traktor fahren kann.
Das aber ist das typische Verhalten der Implikation: Nur im Falle einer wahren Aussage a und einer falschen Aussage b ist die Implikation falsch.



Einfacher verständlich ist die logische Äquivalenz, die zwei Aussagen a und b durch die Wörter »genau dann« und »wenn« verbindet und zu der Aussage »a genau dann, wenn b« führt. Diese Aussage schreibt man symbolisch

$$a \leftrightarrow b$$

Die Äquivalenz ist in denjenigen Fällen wahr, in denen beide Teilaussagen denselben Wahrheitswert besitzen.

a	b	$a \leftrightarrow b$
W	W	W
W	F	F
F	W	F
F	F	W

Die Realisierung des Wunsches »Wo du bist, da will ich auch sein« kann damit durch die Aussage »Ich bin am Ort x genau dann, wenn du am Ort x bist« formuliert werden.

Die Schwierigkeiten, die beim Verständnis der Wertetabellen der logischen Operationen, insbesondere der Implikation, auftreten, rühren - wie gesagt - daher, dass für diese Operationen Bezeichnungen verwendet werden, die im gewöhnlichen Sprachgebrauch bereits einen wohldefinierten, gebräuchlichen Sinn besitzen. Wir haben es auch hier wieder mit einem Umbenennungsvorgang zu tun.

Sehr viel kürzer wäre folgende Einführung der logischen Verknüpfungen. Man betrachte Funktionen $f(a, b)$, die für die beiden Werte W und F erklärt sind und deren Funktionswerte wiederum nur W und F sein können. Offensichtlich sind 16 solcher Funktionen möglich,
Offensichtlich sind 16 solcher Funktionen möglich, was die nachstehende Tabelle zeigt.

	$a\ b$	$a\ b$	$a\ b$	$a\ b$		$a\ b$	$a\ b$	$a\ b$	$a\ b$
Nr.	W W	W F	F W	F F	Nr.	W W	W F	F W	F F
1	W	W	W	W	9	F	W	W	W
2	W	W	W	F	10	F	W	W	F
3	W	W	F	W	11	F	W	F	W
4	W	W	F	F	12	F	W	F	F
5	W	F	W	W	13	F	F	W	W
6	W	F	W	F	14	F	F	W	F
7	W	F	F	W	15	F	F	F	W
8	W	F	F	F	16	F	F	F	F

Die Funktion 8 nennen wir nun Konjunktion, die Funktion 2 Disjunktion, die Funktion 5 Implikation und die Funktion 7 Äquivalenz. Die übrigen Funktionen sollen hier nicht weiter interessieren.

3.4 Das Rechnen mit Gedanken

Ebenso wie man bei den Zahlen gewisse allgemeine Gesetze aufstellen kann, ist das auch in der Logik möglich. Wir erinnern uns, dass die Algebra Gesetze wie $a \cdot b = b \cdot a$ oder $(a + b) \cdot c = a \cdot c + b \cdot c$ enthielt.
Aus ähnlichen Gesetzen besteht die Algebra der Logik, in der die Größen a, b und c die Bedeutung von Aussagen haben. An die Stelle der Rechenoperationen treten die logischen Operationen. Auf diese Weise werden wir gewissermaßen in die Lage versetzt, mit Gedanken zu rechnen.

Wir haben bereits gelernt, zwei Aussagen a und b miteinander zu verknüpfen. Die neue Aussage, die entsteht, wollen wir mit c bezeichnen und dabei durch Klammern kenntlich machen, woraus sie entstanden ist. Wir schreiben also beispielsweise

$$c = (a \wedge b) \qquad \text{oder} \qquad d = (a \vee b \wedge c)$$

Für solche Klammerausdrücke gelten nun eine Reihe von Gesetzen der Algebra der Logik. Wir wollen eine ansehnliche Zahl dieser Gesetze zur Übersicht anführen. Sie stellen praktisch die Denkregeln des »gesunden Menschenverstandes« dar.

Die einfachsten Regeln sind die sogenannten Kommutativgesetze, die beispielsweise lauten

$$(a \vee b) = (b \vee a) \qquad , \qquad (a \wedge b) = (b \wedge a)$$

Die Kommutativgesetze besagen, dass es für die angeführten Operationen gleichgültig ist, in welcher Reihenfolge die Aussagen geschrieben werden. In der Algebra der Zahlen lauten die vergleichbaren Gesetze

$$(a + b) = (b + a) \qquad , \qquad a \cdot b = b \cdot a$$

In den Assoziativgesetzen werden bereits 3 Aussagen, von denen jeweils 2 in Klammern stehen, miteinander verknüpft, wie zum Beispiel

$$a \vee (b \vee c) = (a \vee b) \vee c \qquad , \qquad a \wedge (b \wedge c) = (a \wedge b) \wedge c$$

Diese Gesetze entsprechen den Rechenregeln

$$a + (b + c) = (a + b) + c \qquad , \qquad a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

Während die zwei vorangegangenen Gruppen nur eine einzige logische Verknüpfung enthielten und daher recht einfach waren, müssen wir uns bei den folgenden Distributivgesetzen ein wenig anstrengen, um ihre Gültigkeit einzusehen.

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \qquad , \qquad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

Sei zum Beispiel a die Aussage »Er ist groß« und b die Aussage »Er ist nachdenklich« und schließlich c »Er ist schweigsam«, so lautet die linke Seite des ersten Gesetzes »Er ist groß und nachdenklich oder schweigsam«.

Auf der rechten Seite steht dann: »Er ist groß und nachdenklich oder er ist groß und schweigsam«. Beide Seiten besagen offenbar dasselbe.

Sehr einfach sind dagegen die Reduktionsgesetze:

$$a \wedge a = a \qquad , \qquad a \vee a = a$$

Die Aussage »Das Haus ist schön und das Haus ist schön« ist völlig mit der einfacheren Aussage: »Das Haus ist schön« identisch. Geläufig ist uns sicher das folgende Gesetz der Verneinung

$$\sim\sim a = a$$

das die doppelte Verneinung mit der ursprünglichen Aussage gleichsetzt.

Ein weiteres Gesetz der Verneinung ist das berühmte »Platonsche Falschheitskriterium«:

$$(a \rightarrow \sim a) = \sim a$$

Es besagt, dass eine Aussage a nicht gelten kann, wenn sich aus dieser Aussage ihr Gegenteil $\sim a$ folgern lässt. Ähnlich lautet das »Euklidsche Wahrheitskriterium«, in dem die Aussage a gegenüber dem Falschheitskriterium an allen Stellen durch die Aussage $\sim a$ ersetzt ist:

$$(\sim a \rightarrow a) = a$$

Wenn man aus dem Gegenteil einer Aussage a folgern kann, dass diese Aussage gilt, so gilt sie tatsächlich. Das Euklidsche Wahrheitskriterium und das Platonsche Falschheitskriterium sind beliebte Schlussweisen in der Mathematik.

Ein einfaches Beispiel möge das verdeutlichen. Wir wählen als Aussage a die Behauptung, dass es eine größte ganze natürliche Zahl N gebe. Durch Addition einer 1 zu dieser Zahl N erhalten wir aber sofort die größere Zahl $N+1$ und haben damit aus der Aussage a ihr Gegenteil $\sim a$ gefolgert. Nach dem Platonschen Falschheitskriterium ist damit die Aussage; dass es eine größte ganze Zahl geben könne, widerlegt.
An dieser Stelle sollen noch zwei weitere Gesetze der Verneinung angeführt werden:

$$\sim (a \wedge b) = \sim a \vee \sim b \qquad , \qquad \sim (a \vee b) = \sim a \wedge \sim b$$

Die Verneinung der Aussage »Es ist sieben Uhr, und der Zug fährt ab« lautet demzufolge »Entweder ist es nicht sieben Uhr, oder der Zug fährt nicht ab«.
Verneint man dagegen eine Disjunktion, so erhält man eine Konjunktion der beiden verneinten Teilaussagen.

Von größter Bedeutung ist die Tatsache, dass sich alle logischen Verknüpfungen auf Konjunktionen und Negationen zurückführen lassen. Dies hat in der Algebra der Zahlen eine Parallele. Auch dort lassen sich alle Rechenoperationen auf wenige Grundrechenoperationen reduzieren. Stellvertretend sei die Auflösung der Implikation angegeben:

$$a \rightarrow b = \sim (a \wedge \sim b)$$

Diese Formel behauptet, »Aus a folgt b« ist gleichbedeutend mit »Es gilt nicht, dass a gilt und gleichzeitig b nicht gilt«.
Betrachten wir einmal den Satz »Wenn es Sommer ist, geht die Sonne früh auf«. Diese Implikation ist doch offensichtlich mit folgender Aussage identisch: »Es kann nicht sein, dass es Sommer ist und dass die Sonne nicht schon früh aufgeht.«

3.5 Nicht alle Kreter lügen

Die konsequente richtige Anwendung der logischen Gesetze kann nie zu Widersprüchen oder undurchsichtigen Spekulationen führen.
Die Befürchtung, die Logik führe infolge ihrer formalen Schlussmethoden zu falschen Ergebnissen, ist gänzlich unberechtigt.

Der Mann aus Kreta, der behauptet, dass alle seine Landsleute Lügner seien, verbindet die Aussage $a =$ »alle Kreter lügen« und die Aussage $b =$ »ich bin aus Kreta« zu der Aussage $c = a \wedge b$.
Nun folgt aber aus der Tatsache, dass er von dieser merkwürdigen Insel stammt, dass er lügt und infolgedessen auch seine Aussage c nicht zutrifft.

Wir haben also aus der Aussage c ihr Gegenteil gefolgert oder in Formeln den Schluss

$$c \to \sim c$$

gezogen. Nach dem Platonschen Falschheitskriterium ist aber dieser Ausdruck mit $\sim c$ identisch:

$$c \to \sim c = \sim c$$

Wir müssen also das Gegenteil von $c = (a \wedge b)$ bilden. Dazu verwenden wir das entsprechende Gesetz der Verneinung und erhalten

$$c = \sim (a \wedge b) = \sim a \vee \sim b$$

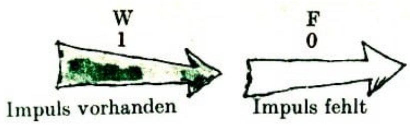
Das bedeutet in Worten: »Nicht alle Kreter lügen, oder ich bin nicht aus Kreta.« In einer anderen Form geschrieben, lautet dieser Sachverhalt: »Entweder gibt es zumindest einen Kreter, der nicht lügt, oder ich bin nicht aus Kreta.«
Der in die spanischen Stiefel der formalen Logik gezwängte Geist wird hierbei keineswegs verkrüppelt oder misshandelt.

Die Formalisierbarkeit der Logik erleichtert dem Menschen, logische Entscheidungen zu treffen, und entlastet ihn von unnötiger Geistesarbeit bei der Entschlüsselung und Vereinfachung komplizierter logischer Ausdrücke oder bei der Lösung verwickelter logischer Aufgaben.
Die formale Logik gibt uns aber vor allem auch die Mittel in die Hand, logische Probleme und Entscheidungsverfahren so aufzuschreiben, dass deren Lösung elektronischen Rechenautomaten anvertraut werden kann.

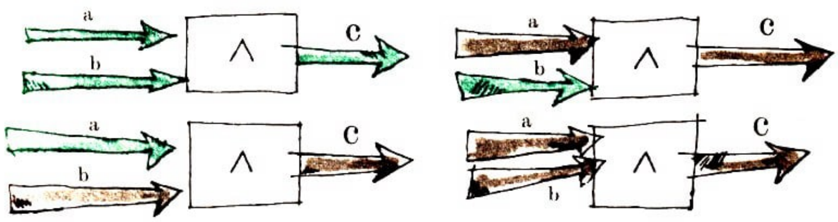
3.6 Nichts ist so praktisch wie eine gute Theorie

Die Formalisierbarkeit der Logik macht es möglich, Modelle ihrer Verknüpfungen als elektronische Schaltungen zu bauen und diese in beliebigen Automaten fest zu »verdrahten«. Solche Automaten sind perfekte Logiker.
Sie haben die Logik in Form von Schaltungen aus Röhren, Transistoren und Ferritkernen stets »griffbereit« zur Verfügung und sind durch ihren Konstrukteur gezwungen, diese Gesetze ständig und richtig anzuwenden.

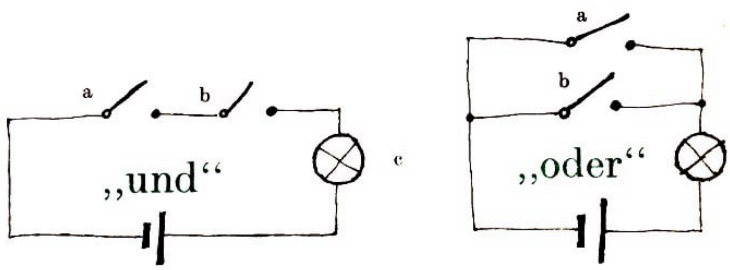
Es sei beispielsweise die Aufgabe gestellt, eine Schaltung für die logische Konjunktion $c = a \wedge b$ zu bauen. Eine solche Schaltung wird zwei Eingangsleitungen a und b sowie eine Ausgangsleitung c besitzen.
Die Eingangssignale und das resultierende Ausgangssignal sollen elektrische Impulse sein. Dabei soll dem Vorliegen eines Impulses der Wahrheitswert W und dem Fehlen eines Impulses der Wahrheitswert F zugeordnet werden. Dann ist eine elektrische Verknüpfung der Leitung so herzustellen, dass die Tabelle der Wahrheitswerte dieser logischen Operation in Form von elektrischen Impulsen realisiert wird.
Das Vorliegen bzw. das Fehlen eines Impulses in der Leitung wird durch geeignete Symbole dargestellt:



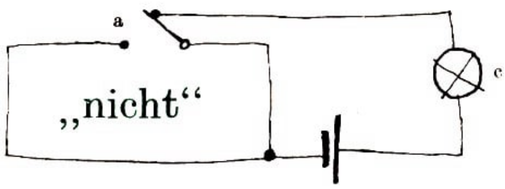
Damit ergibt sich für die logische »und«-Schaltung folgendes Gesamtverhalten:



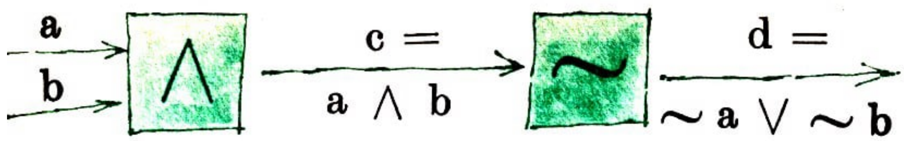
Dieses Verhalten spiegelt tatsächlich die Tabelle von Seite 26 wider. Die Schaltung liefert nur dann einen Ausgangsimpuls, wenn auf beiden Eingangsleitungen a und b ein Impuls vorliegt.
Mit zwei Schaltern, einer Glühbirne und einer Taschenlampenbatterie lassen sich ganz einfache Modelle der logischen Verknüpfungen »und« und »oder« bauen. In der ersten Schaltung brennt



die Glühbirne c dann und nur dann, wenn beide Schalter a und b geschlossen sind. Das ist genau das Verhalten der Konjunktion, b der logischen »und«-Funktion.
Im zweiten Fall liegt das Verhalten der logischen »oder«-Beziehung vor, weil die Glühbirne c immer dann aufleuchtet, wenn mindestens einer der beiden Schalter geschlossen ist.
Auf ähnliche Weise verdeutlicht man die Negation. Allerdings muss hier ein Zweipunktschalter verwendet werden. Steht er wie die Schalter der früheren Bilder in der Stellung »aus«, so schließt er gleichzeitig einen Stromkreis, der die Glühbirne zum Leuchten bringt.



In der umgekehrten Stellung »ein« dagegen leuchtet das Lämpchen nicht.



Die logischen Schaltungen, die im Rechenautomaten natürlich aus anderen Elementen bestehen, lassen sich in beliebiger Weise kombinieren. Führen wir zum Beispiel die Ausgangsleitung c der Konjunktionsschaltung auf den Eingang einer Negationsschaltung, so liefert deren Ausgangsleitung ein Signal d, das die logische Verknüpfung verwirklicht.

Elektronische Rechenautomaten benötigen zur Realisierung der Rechenoperationen oder zur Entschlüsselung von Befehlen eine Vielzahl komplizierter logischer Schaltkombinationen. Dabei ist man bestrebt, die Anzahl der Schaltelemente so niedrig wie möglich zu halten.
Man sucht also elektronische Schaltungen, die das geforderte logische Verhalten zeigen, aber aus einem Minimum an Schaltelementen bestehen. Diese Forderung ist gleichbedeutend mit der Suche nach einer minimalen Anzahl von Schaltungstypen.

Bei einer genaueren Durchsicht der angeführten logischen Gesetze zeigt es sich, dass alle logischen Operationen auf die Konjunktion, die Disjunktion und die Negation zurückzuführen sind. Daraus ergibt sich die wichtige technische Konsequenz, dass sich mit Hilfe der Konjunktionsschaltung, der Disjunktionsschaltung und der Negationsschaltung jedes beliebige logische Verhalten realisieren lässt.

Diese und ähnliche Fragen werden in der Schaltalgebra untersucht, die sich als höchst nützliches und konkretes Produkt aus der scheinbar so unnützen und »übertheoretischen« Logik entwickelt hat. Wie überall, so hat sich auch hier der Satz bestätigt, dass nichts praktischer sein kann als eine gute Theorie.

Die Bestrebungen, logische Maschinen zu bauen, sind sehr alt. Bereits im 13. Jahrhundert dachte Raimundus Lullus über den Bau einer solchen Maschine nach. Er sah

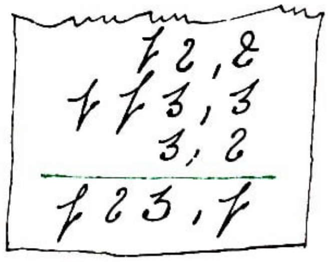
im Geist einen Apparat vor sich, der all die schönen Schlüsse der römischen Logiker nachmachen sollte. Seine Vorstellungen und Mittel waren damals unzulänglich.
Doch hat Lullus später durch die Entstehung der symbolischen Logik eine gewisse Rechtfertigung erfahren, wenn sich auch die Schaltalgebra zu seinen Bemühungen etwa so verhält wie die moderne Atomumwandlung zu den Versuchen der alchimistischen Goldmacher.

Unser Wissen über elektronische Rechenmaschinen hat bereits eine stattliche Höhe erreicht. Aber warum diese Maschinen eigentlich Rechenmaschinen heißen und wieso sie rechnen können, ist uns bis jetzt noch unklar geblieben. Es ist an der Zeit, einige Bemerkungen über Zahlen und Rechenoperationen folgen zu lassen.

4 Das Einmaleins mit der Eins

4.1 Geheimnisvolle Kassenzettel

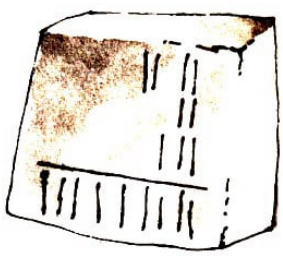
Unter all den Merkwürdigkeiten, die ein Raumschiff dereinst von einem entfernten bewohnten Sternsystem mitbringen wird, könnte sich vielleicht auch ein Kassenzettel befinden, der folgende Rechnung zeigt:



Bei der Übersetzung dieser Zeichen stehen wir vor einer Aufgabe, wie sie oft in illustrierten Zeitschriften vorkommt. Auf den ersten Blick fällt uns auf, dass nur vier Ziffern verwendet werden. Das kann natürlich bloßer Zufall sein, erweist sich aber hier als Schlüssel des Problems.

Es lässt sich zeigen, dass es vier solche Dezimalziffern, die, an die Stelle der unbekannten Zeichen gesetzt, die Aufgabe lösen, nicht gibt. Uns bleiben zwei Erklärungen für diese Erscheinung, wenn man das Vorliegen eines Rechenfehlers ausschließt. Entweder bedienen sich die Bewohner des fernen Sternsystems einer uns unbekannten Addition, oder sie benutzen nicht das Dezimalsystem. Die letzte Vermutung wollen wir für wahrscheinlicher als die erste halten.

Ehe wir dieses Rätsel vollends lösen, lassen wir jetzt ein weiteres Raumschiff zu einem anderen Sternsystem fliegen, auf dem ein recht primitiver Kassenzettel vorgefunden werde:



Offensichtlich ist die Entschlüsselung dieser Rechnung einfach:

$$\begin{aligned} IIII &= 4 \\ III &= 3 \\ II &= 2 \\ IIIIIIIII &= 9 \end{aligned}$$

Die Bewohner dieser Gegend im All benutzen ein Zahlensystem, bei dem jede Zahl durch die entsprechende Menge von Strichen dargestellt wird. Man kann sich leicht vorstellen, dass eine solche Zahlendarstellung unübersichtlich, fehleranfällig und sehr papierfüllend ist.

Bevor wir aber in ein gewiss berechtigtes Lächeln über diese rückständige Methode verfallen, müssen wir bedenken, dass dieselbe vor einigen hundert Jahren auch auf der Erde verbreitet war. Zumindest zeigen die ersten 3 Ziffern

I, II, III

des römischen Zahlensystems den Charakter der Reihendarstellung mit großer Deutlichkeit. Aber auch das Symbol

V

das aus dem Bild der Hand bei seitwärts gestrecktem Daumen hervorgeht, ist lediglich eine Abkürzung für 5 nebeneinander geschriebene Striche. Jedes römische Zahlensymbol steht stellvertretend für eine Reihe einzelner Einheiten. Die ersten 7 Grundsymbole haben folgende Bedeutung:

I = 1, V = 5, X = 10, L = 50, C = 100, D = 500, M = 1000

Das Nebeneinanderstehen von n gleichen Ziffern bedeutet das $n$fache dieser Ziffer, wie bei

XX = 20

Ein Ziffernpaar, bei dem die »kleinere« als erste erscheint, bedeutet die Differenz aus der »größeren« und der »kleineren« Ziffer. So ist

IV = V - I = 4

Steht die kleinere Ziffer rechts, so wird die Summe gebildet, was wir an

VI = V + I

kennen. Die Jahreszahl 1969 schreibt sich demnach in den ehrfurchtgebietenden römischen Symbolen als

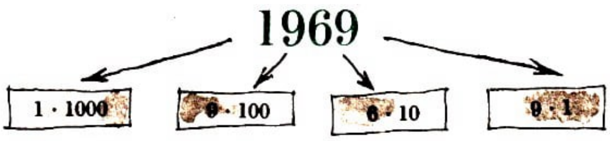
MCMLXIX

und ist eine Abkürzung für 1969 einzelne Jahre.

4.2 Jede Zahl ist Gründer eines Zahlensystems

Im 9. Jahrhundert lehrte in Bagdad der Perser al-Huwarizmi. Er verfasste eine Aufgabensammlung für Kaufleute und Testamentvollstrecker, in der er unter anderem verwickelte Erbteilungsprobleme behandelte.
In diesem Buch wurden bereits die uns heute so vertraut gewordenen arabischen Ziffern verwendet. Diese Zahlensymbole waren eine wichtige Voraussetzung für die weitere Entwicklung der Mathematik. Die lateinische Übersetzung des Namens al-Huwarizmi führte zum Ausdruck Algoritmi, woraus schließlich das Wort Algorithmus entstand.
Auch aus dem Namen der Aufgabensammlung des großen Rechenmeisters entstand in der Folgezeit ein Fachwort. Es war das Wort Algebra.

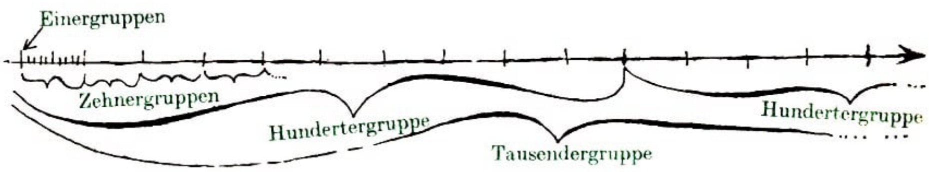
In diesem arabischen oder dezimalen System ist einiges anders als im römischen Zahlensystem. Die Dezimalzahl 1969 steht hier stellvertretend für einen Tausender, 9 Hunderter, 6 Zehner und 9 Einer.



Im Gegensatz zur Reihendarstellung des römischen Zahlensystems liegt hier eine Gruppendarstellung vor. Alle Potenzen von 10

$$...0{,}01 = 10^{-2}, 0{,}1 = 10^{-1}, 1 = 10^0, 10 = 10^1, 100 = 10^2, ...$$

grenzen Gruppen innerhalb der Zahlengeraden ab. Die Zahl 124 besteht zum Beispiel aus einer Hundertergruppe, 2 Zehnergruppen und 4 Einergruppen.



Das Dezimalsystem ist nur ein System aus einer Vielzahl möglicher Systeme. Eine außerordentliche Gewöhnung an die Grundzahl 10 spiegelt lediglich vor, dass diese Wahl besonders sinnvoll ist. In Wirklichkeit kann jede ganze positive Zahl mit nicht geringerer Berechtigung als die 10 zur Grundzahl erhoben werden.
Nennen wir diese Zahl g, so wird das auf ihr aufgebaute Zahlensystem aus den Ziffern

0, 1, 2, 3, ..., g-1

bestehen. Eine beliebige n-stellige ganze Zahl hat dann die Form

$$a_n, a_{n-1}, ..., a_2, a_1, a_0$$

wobei die verschiedenen a_i die einzelnen Ziffern, d. h. Zahlen zwischen 0 und $g - 1$ sind. Der Wert dieser Zahl ergibt sich zu

$$a_n \cdot g^n + a_{n-1} \cdot g^{n-1} + ... + a_2 \cdot g^2 + a_1 \cdot g^1 + a_0 \cdot g^0$$

Das Zehnersystem ordnet sich in dieses Schema mit $g = 10$ ein. Daraus folgt, dass die Ziffern dieses Systems die Zahlen 0 bis 9 sind.
Die Zahl 124 besteht zum Beispiel aus den Ziffern 1, 2, 4 und hat den Wert

$$1 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0 = 124$$

In der Tabelle seien einige Zahlensysteme an Hand ihrer ersten 16 ganzen Zahlen vorgestellt:

In dieser Tabelle stehen oben die jeweiligen Grundzahlen, so dass die linke Spalte herkömmliche Dezimalzahlen enthält.
Beim System mit der Grundzahl 14 müssen wir neue Bezeichnungen einführen, weil für diesen Fall 13 Ziffern benötigt werden, d. h. 13 Zahlen, die nur aus je einem Symbol bestehen, während das Zehnersystem lediglich 9 Ziffern kennt. Die dezimale Zahl 10 besteht also im System mit der Grundzahl 14 aus einer einzigen Ziffer und wurde deswegen als A geschrieben.

10	2	4	5	8	14
1	1	1	1	1	1
2	10	2	2	2	2
3	11	3	3	3	3
4	100	10	4	4	4
5	101	11	10	5	5
6	110	12	11	6	6
7	111	13	12	7	7
8	1000	20	13	10	8
9	1001	21	14	11	9
10	1010	22	20	12	A
11	1011	23	21	13	B
12	1100	30	22	14	C
13	1101	31	23	15	D
14	1110	32	24	16	10
15	1111	33	30	17	11
16	10000	100	31	20	12

Beim Lesen von Zahlen, die keine Dezimalzahlen sind, hat man anfangs einige Schwierigkeiten. Wir sind immer wieder geneigt, sie als Dezimalzahlen anzusehen.
Wenn wir von der Zahl 23 sagen, dass sie eine Zahl aus dem System mit der Grundzahl 5 darstellen soll, so ist sie nicht als »dreiundzwanzig« oder, was dasselbe ist, als »2 Zehner und 3 Einer« zu lesen, sondern als »2 Fünfer und 3 Einer«. Die Zahl 101 des Zweiersystems ist nicht die Zahl »Hundert und eins« oder »ein Hunderter, kein Zehner und ein Einer«, sondern die Zahl »ein Vierer, kein Zweier und ein Einer« und entspricht also der Dezimalzahl 5.

Angesichts der Tatsache, dass jede ganze positive Zahl größer als Null zur Grundzahl eines Zahlensystems erhoben werden kann, ist es berechtigt, die Verbreitung des Zehnersystems auf der Erde eine historische Zufälligkeit zu nennen.
Wenn wir durchaus nach einer »sinnvollen« Ursache suchen, so bietet sich lediglich die Tatsache an, dass der Mensch zehn Finger besitzt. In diesem Sinne könnte man jederzeit die Wette eingehen, dass wir das Vierersystem eingeführt hätten, wenn wir an jeder Hand nur Daumen und Zeigefinger besäßen.

4.3 Wenn wir nur Daumen und Zeigefinger hätten

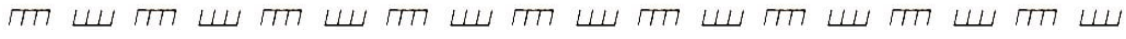
Wir wollen einmal versuchen, die »Entdeckung des Vierersystems« nachzuerleben: Versetzen wir uns zunächst in jene Epoche zurück, in der die Zahlensymbole noch nicht bekannt waren. Wollte man damals eine Menge bezeichnen, so war man gezwungen, Kieselsteine auf den Tisch zu legen, Striche in die Wand zu ritzen oder Finger in die Höhe zu heben.
Nach einiger Zeit dürfte sich herausgestellt haben, dass es der Mengenangabe »///« relativ gleichgültig war, welche konkreten Gegenstände sich hinter ihr verbargen. Wenn man zu /// Schafen / Schaf dazustellte oder zu /// Äpfeln / Apfel dazulegte oder schließlich zu /// Häusern / Haus hinzubaute, so ergab das Ergebnis immer eine Angabe von ////.

Damit war als Abstraktion all der Mengen, die gleichviel Elemente besaßen, der Begriff der Zahl entstanden. Nun entdeckte man noch, dass auch die Angaben »kein Schaf«, »kein Apfel« und »kein Haus« in dieses Schema passten, und hatte die Null erfunden.

Wir nehmen jetzt an, dass unsere Entdecker des Zahlenbegriffs an jeder Hand lediglich // Finger besaßen, was bei // Händen eine Gesamtzahl von //// ausmachte.
Bei der Betrachtung einer großen Zahl, etwa der Zahl

///
Einheiten

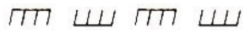
hat man sicherlich je vier Striche durch einen Querstrich verbunden:

 ///

Anschließend zählte man die Gruppen aus und erhielt

////////////////// ← Gruppen

Als Rest verblieben /// Einheiten. Danach fasste man je //// Gruppen zu einer Obergruppe zusammen

 //

und kam auf

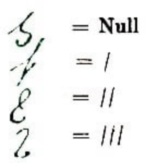
//// Obergruppen

Als Rest blieben hier // Gruppen. Schließlich fasste man je //// Obergruppen zu einer Hauptgruppe zusammen

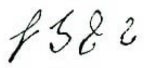
und stellte fest, dass so genau / Hauptgruppe entstand, ohne dass eine Obergruppe als Rest blieb.

Die ursprüngliche Zahl lautete demnach in der Viererschreibweise / Hauptgruppe und keine Obergruppe und // Gruppen und /// Einheiten.
Bei einer in dieser Weise vorgenommenen Gruppeneinteilung konnte in keiner Stufe ein Rest entstehen, der größer als /// war. Es genügte also, jeder der vier Zahlen Null, /, // und /// ein Symbol zuzuordnen, um die unschöne Strichschreibweise zu vermeiden.

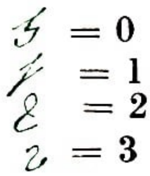
Wir identifizieren nun die Erfinder dieses Zahlensystems mit den Bewohnern jenes Sternsystems, von dem ein Raumschiff den im ersten Abschnitt dieses Kapitels gezeigten Kassenzettel mitbrachte, und unterstellen, dass die dortigen Bewohner folgende Symbole entwickelt hatten:



Mit diesen Symbolen schreibt sich unsere Zahl in der einfachen Ziffernschreibweise



Man sieht dieser Zahl an, dass sie von einem fernen Stern kommt. Ungewohnt, fremd und nichtssagend stehen die Ziffern auf dem Papier. Wir sollten aber bedenken, dass auch unser Zehnersystem in ähnlicher Weise entstanden und uns nur durch Gewöhnung vertraut geworden ist.
Die fremdartigen Symbole haben also folgende Bedeutung:



Wenn wir damit den Kassenzettel übersetzen, erhalten wir eine recht merkwürdige Rechnung:

$$\begin{array}{r} 13{,}2 \\ 110{,}0 \\ 0{,}3 \\ \hline 130{,}1 \end{array}$$

Auf den ersten Blick könnten wir vermuten, dass hier mehrere grobe Rechenfehler vorliegen. Doch dürfen wir nicht vergessen, dass diese Zahlen zwar mit Dezimalziffern geschrieben sind, dass sie aber Zahlen des Vierersystems darstellen.
Addieren wir 3 und 2, so ergibt sich zunächst 5. Diese Ziffer existiert jedoch im Vierersystem nicht. Mit der 5 ist ja die erste Vierergruppe um 1 überschritten. Wir schreiben daher

$$\begin{array}{r} 13{,}2 \\ 110{,}0 \\ 0_1{,}3 \\ \hline 130{,}1 \end{array}$$

und merken durch eine kleine 1 an, dass zu der Summe aus den 3 Ziffern 3, 0 und 0 noch die 1 aus der ersten Überschreitung dazugenommen werden muss. So ergibt $3+0+0+1=4$. Aber auch die Ziffer 4 existiert im Vierersystem nicht, da ja 4 Einheiten eine Gruppe bilden. Es ergibt sich also

$$\begin{array}{r} 13{,}2 \\ 110{,}0 \\ {}_10{,}3 \\ \hline 130{,}1 \end{array}$$

wobei die kleingeschriebene 1 wieder darauf hindeutet, dass sie bei der Addition der nächsten Spalte berücksichtigt werden muss. So entsteht $1+1+1=3$. Hier liegt keine Überschreitung einer Vierergruppe vor. Wir können die Addition zu Ende führen und erhalten wirklich

$$\begin{array}{r} 13{,}2 \\ 110{,}0 \\ 0{,}3 \\ \hline 130{,}1 \end{array}$$

Was anfangs ein Irrtum schien, erweist sich so als System. Abschließend wollen wir uns noch die leicht nachprüfbaren Tabellen für die Addition und Multiplikation der Zahlen 0 bis 3 im Vierersystem ansehen.

+	0	1	2	3
0	0	1	2	3
1	1	2	3	10
2	2	3	10	11
3	3	10	11	12

·	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	10	12
3	0	3	12	21

Will man wissen, welches Ergebnis beispielsweise aus der Addition $3+2$ im Vierersystem entsteht, sucht man in der linken Spalte die Zahl 3 und in der oberen Zeile die Zahl 2. Im Kreuzungspunkt dieser Zeile mit der entsprechenden Spalte steht dann das Ergebnis 11.
Genauso verfährt man in der Multiplikationstabelle.

Man hüte sich aber davor, die im Vierersystem geschriebene Zahl 11 als »elf« zu sprechen. Diese Zahl lautet »1 Vierer und 1 Einer« und hat daher den Wert »fünf«.

4.4 Ein Zahlensystem für Lernfaule

Ein weiteres Zahlensystem, das unbedingt eine nähere Untersuchung erfordert, ist das Dualsystem. Es beruht auf der Grundzahl 2 und kennt daher nur die Ziffern 0 und 1, was zur Folge hat, dass die Tabellen für die Addition und die Multiplikation in diesem Zweiersystem ein Höchstmaß an Einfachheit erreichen:

Additionstabelle

+	0	1
0	0	1
1	1	10

Multipikationstabelle

·	0	1
0	0	0
1	0	1

Auf einem Stern, dessen Bewohner das Dualsystem verwenden, beschränkt sich das Erlernen des kleinen Einmaleins auf die Aneignung der Tatsache, dass

$$0 \cdot 0 = 0, \quad 1 \cdot 0 = 0, \quad 1 \cdot 1 = 1$$

ist.
Wie einfach das auch sein mag, so hat doch das Dualsystem einen ernsthaften Nachteil gegenüber dem Dezimalsystem. Er besteht in der größeren Länge der Dualzahlen. Das zieht gesetzmäßig eine größere Fehlerwahrscheinlichkeit beim Rechnen nach sich und erfordert daher größere Aufmerksamkeit.
Sollten wir also Wesen finden, die sich nur des Dualsystems bedienen, so wären diese zwangsläufig sehr gewissenhaft, genau und ausdauernd. Unwillkürlich denken wir dabei an Pedanten, aber auch und vor allem an Automaten.

An zwei Beispielen sollen das Rechnen im Dualsystem und im Dezimalsystem gegenübergestellt werden:

$$
\begin{array}{r}
11010111 \\
+\ 1101 \\
\hline
11100100
\end{array}
\qquad
\begin{array}{r}
215 \\
+\ 13 \\
\hline
228
\end{array}
\qquad
\begin{array}{r}
11010111 \cdot 1101 \\
\hline
11010111 \\
00000000 \\
11010111 \\
11010111 \\
\hline
101011101011
\end{array}
\qquad
\begin{array}{r}
215 \cdot 13 \\
\hline
215 \\
645 \\
\hline
2795
\end{array}
$$

Während die Addition der beiden Zahlen des ersten Beispiels im Dualsystem noch ohne große Mühe gelingt, müssen wir bei der Addition der vier Zahlen, die bei der Multiplikation entstanden, unsere volle Aufmerksamkeit daran setzen, keinen Übertrag zu vergessen

In der nachstehenden Tabelle sind alle Dualzahlen enthalten, die den ersten 12 Potenzen der Grundzahl 2 entsprechen.

Dualzahl	Potenz von 2	Wert
1	2^0	1
10	2^1	2
100	2^2	4
1000	2^3	8
10000	2^4	16
100000	2^5	32
1000000	2^6	64
10000000	2^7	128
100000000	2^8	256
1000000000	2^9	512
10000000000	2^{10}	1024
100000000000	2^{11}	2048
1000000000000	2^{12}	4096

Wir sehen, dass bei diesen Dualzahlen die Anzahl der Nullen mit dem Exponenten der entsprechenden Zweierpotenz übereinstimmt. Mit Hilfe dieser Tabelle lassen sich die Ergebnisse der im Dualsystem vorgenommenen Berechnungen leicht kontrollieren. So bedeutet

$$1101 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13$$

oder

$$10101110101 = 1{\cdot}2^{11}+0{\cdot}2^{10}+1{\cdot}2^9+0{\cdot}2^8+1{\cdot}2^7+1{\cdot}2^6+1{\cdot}2^5+0{\cdot}2^4+1{\cdot}2^3+0{\cdot}2^2+1{\cdot}2^1+1{\cdot}2^0 = 2795$$

Zur besseren Unterscheidung von Dual- und Dezimalzahlen versuchte man die duale 1 dadurch kenntlich zu machen, dass man sie auf den Kopf stellte. und damit z. B. für die 9 im Zweiersystem schrieb

$$\rotatebox[origin=c]{180}{1}00\rotatebox[origin=c]{180}{1}$$

Später hat dann wahrscheinlich Undeutlichkeit oder Flüchtigkeit aus der verkehrten 1 ein regelrechtes L hervorgebracht. So jedenfalls kann man die Entstehung der heute überaus gebräuchlichen Darstellung der dualen 1 erklären. Damit schreibt sich also die 9 im Dualsystem

$$9 = \mathsf{LOOL}$$

4.5 Die Übersetzung von Zahlen

Das Dualsystem ist hervorragend geeignet, in elektronischen Rechenmaschinen verwendet zu werden. Es erfordert dort Schaltelemente, die lediglich zwei voneinander verschiedene elektrische Zustände besitzen müssen und denen man die Ziffern 0 und 1 zuordnen kann.

Da der Mensch vom Dezimalsystem kaum ablassen wird und andererseits das Dualsystem für den Automaten große Vorteile besitzt, muss man nach Übersetzungsmöglichkeiten von einem System in das andere suchen, die sowohl vom Menschen als auch vom Automaten gehandhabt werden können. Wir benötigen einen Algorithmus für die Übersetzung von Zahlen aus einem Ziffernsystem in ein anderes.

Für die Übertragung von Dezimalzahlen ins Dualsystem gibt es ein recht einfaches Verfahren. Die zu übersetzende Zahl sei 215. Wir dividieren diese Zahl durch 2, führen aber die Operation nicht bis zum Ende durch, sondern brechen sie ab, sobald ein Rest entsteht, der kleiner als 2 ist:

$$\begin{array}{l} 215 : 2 = 107 \\ \underline{2} \\ \;\;15 \\ \;\;\underline{14} \\ \;\;\;\;1 \end{array}$$

Diesen Rest wollen wir im Gedächtnis behalten, Nun nehmen wir den ganzen Anteil 107, der sich bei der Division ergab, und dividieren ihn durch 2, wobei wir die Operation wieder beenden, sobald ein Rest entsteht, der kleiner als 2 ist:

$$\begin{array}{l} 107 : 2 = 53 \\ \underline{10} \\ \;\;\;\;7 \\ \;\;\;\;\underline{6} \\ \;\;\;\;1 \end{array}$$

Auch diesen Rest halten wir fest. Mit der Zahl 53 geschieht wieder dasselbe. Wir dividieren sie durch 2 und markieren den Rest. Wir verfahren in dieser Weise weiter, bis ein ganzer Anteil entsteht, der seinerseits kleiner als 2 ist:

$$\begin{array}{rl} 53 : 2 = 26 & \text{Rest} = 1 \\ 26 : 2 = 13 & \text{Rest} = 0 \\ 13 : 2 = 6 & \text{Rest} = 1 \\ 6 : 2 = 3 & \text{Rest} = 0 \\ 3 : 2 = 1 & \text{Rest} = 1 \end{array}$$

Schreiben wir jetzt alle Reste - beginnend mit dem letzten ganzen Anteil, der kleiner als 2 ist - in der umgekehrten Reihenfolge, in der sie entstanden sind, nieder, so erhalten wir die Darstellung der Dezimalzahl 215 in der Zweierschreibweise:

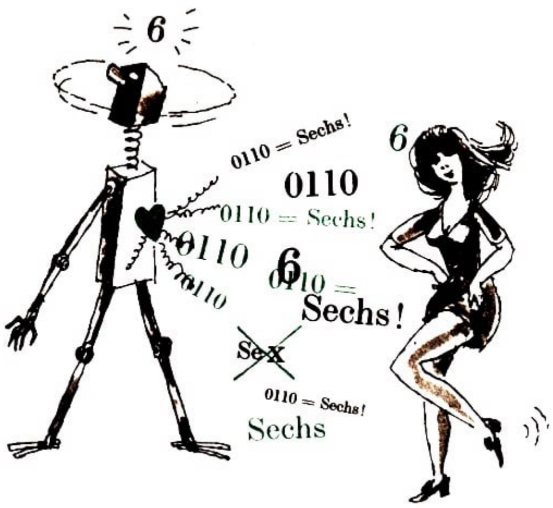
$$11010111$$

Dieser abbrechende Algorithmus lässt sich sinngemäß auf zwei beliebige Zahlensysteme ausdehnen, wenn nur beachtet wird, dass die Divisionen auch wirklich in dem Zahlensystem ausgeführt werden, in dem die umzuwandelnde Zahl geschrieben ist. Des weiteren kann der Algorithmus auf die Umwandlung von echten Brüchen übertragen werden, wenn man an die Stelle der Division die Multiplikation setzt und die Rolle der Reste den Zahlen überträgt; die bei der Multiplikation vor dem Komma entstehen.
Solche oder ähnliche Algorithmen werden im Rechenautomaten zur Umwandlung von Dezimalzahlen in Dualzahlen und umgekehrt zur Übersetzung von Dualzahlen in die dezimale Form verwendet.

Vergleichen wir diese Übersetzung von Zahlen mit der Übersetzung von Wörtern, so ist eine gewisse Ähnlichkeit erkennbar. Die Bedeutung der Zahl liegt letzten Endes in der Anzahl der Einheiten, die sie angibt. Die Formulierung dieser Bedeutung in verschiedenen Sprachen, d. h. in verschiedenen Zahlensystemen, entspricht dabei den verschiedenen Wörtern, die wir in zwei Sprachen für denselben Gegenstand verwenden. Dass wir dabei im Dualsystem dieselben Ziffern 0 und 1 benutzen, die wir vom Dezimalsystem kennen, ist bei der Übersetzung von Texten dem Fall vergleichbar, dass beide Sprachen auf demselben Alphabet, zum Beispiel dem lateinischen, aufgebaut sind.

Sprache 1	Sprache 2	Bedeutung
часы	Uhr	
101	5	/////

Beim Lernen einer Fremdsprache kann es vorkommen, dass wir neben Wörtern der alten Sprache gleichzeitig Wörter der zu erlernenden Sprache in der Unterhaltung verwenden. Eine ähnliche Erscheinung ist beim Gespräch des Menschen mit dem Rechenautomaten zu beobachten. Zwischen der dualen Sprache des Automaten und der dezimalen Redeweise des Menschen hat sich gewissermaßen ein Kompromiss eingebürgert.



Die einzelnen Ziffern einer Dezimalzahl werden dem Automaten nicht in dezimaler Form, sondern in dualer Darstellung eingegeben. Jede Dezimalziffer wird dabei aber streng für sich in eine Dualzahl umgewandelt, so dass der ursprüngliche Charakter der Dezimalzahl erhalten bleibt.

Da wir zur Niederschrift der größten einstelligen Dezimalzahl 9 insgesamt 4 Dualstellen benötigen, wurde diese Anzahl auch für die Darstellung aller anderen Ziffern beibehalten. Die Dezimalzahl 215 schreibt sich in diesem »dezimaldualen« System demnach in der Form

$$\underbrace{0010}_{2}\ \underbrace{0001}_{1}\ \underbrace{0101}_{5}$$

Die Gruppen zu jeweils 4 Dualziffern haben den Namen Tetraden erhalten. Die Dezimalziffern 0 bis 9 schreiben sich in Form von Tetraden folgendermaßen:

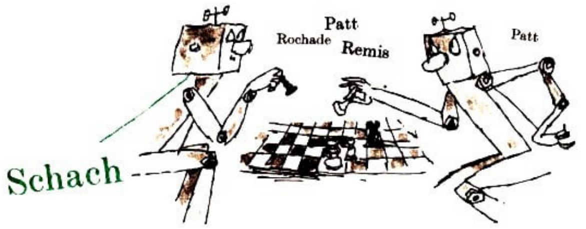
Tetrade	Dezimalziffer	Tetrade	Dezimalziffer
0000	0	0001	1
0010	2	0011	3
0100	4	0101	5
0110	6	0111	7
1000	8	1001	9

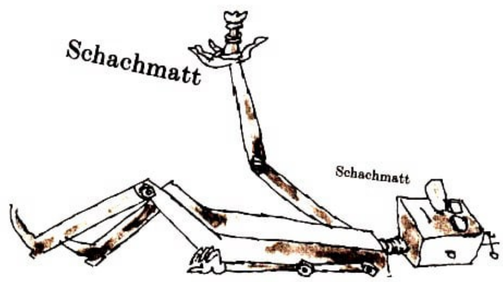
Man erkennt, dass eine Tetrade durch die Dezimalzahl 9 nicht ausgeschöpft wird. Die größtmögliche Dezimalzahl, die in einer Tetrade untergebracht werden könnte, ist die 15, die ja die Dualdarstellung

$$1111 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 15$$

besitzt.

In diesem Kapitel haben wir diejenigen Eigenschaften der Zahlen untersucht, die ihrer Darstellung und Verarbeitung in Rechenautomaten zugrunde liegen. Wir sind gewohnt, von Maschinen zu sagen, dass sie irgend etwas bearbeiten oder herstellen.
In diesem Sinne lassen sich die elektronischen Rechenautomaten als Maschinen ansehen, die Zahlen bearbeiten und Berechnungen »herstellen«.





Wir wissen aber auch, dass Automaten Übersetzungen ausführen und Schach spielen können. Offenbar ist es also nicht nur die Zahl, die den Rechenmaschinen als Rohmaterial dient, sondern ein »allgemeinerer Stoff«. Im folgenden wird sich herausstellen, dass Rechenautomaten vor allem informationsverarbeitende Maschinen sind.

5 Eine unsichtbare Ware

5.1 Der Zufall wird Gesetzen unterworfen

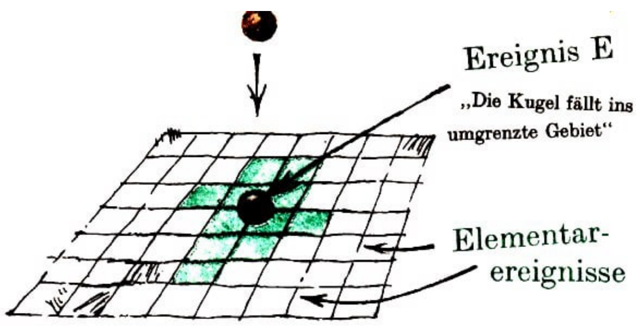
Der Begriff der Information soll eine ähnliche Bedeutungswandlung erfahren wie die Begriffe Berechenbarkeit und Entscheidbarkeit. Allerdings ist hier eine größere Vorbereitung nötig. Wir gehen dabei von folgender Überlegung aus:

Ein Gewinn an Information bedeutet gleichzeitig einen Verlust an Unkenntnis oder Ungewissheit. Die Ungewissheit findet sich aber gerade dort, wo der Zufall waltet, wo also Ereignisse nicht mit Sicherheit, sondern mit einer gewissen Wahrscheinlichkeit eintreten. Dem Begriff Wahrscheinlichkeit müssen wir uns daher zunächst zuwenden.

Er soll aber hier nicht an dem sonst üblichen Würfel demonstriert werden, sondern an einer anderen, etwas komplizierteren Vorrichtung, von der aber anzunehmen ist, dass sie die Wahrscheinlichkeit besser verständlich macht als der Würfel.
Es sei dies eine völlig ebene Stahlplatte, die in einzelne gleichgroße Felder geteilt und von einer genügend hohen Umrandung umgeben ist.

Auf diese Platte lassen wir eine kleine Stahlkugel fallen, die in irgendeinem Feld zur Ruhe kommen wird. Bleibt diese Kugel zufälligerweise auf einem waagerechten Strich liegen, so sei sie dem unteren, von diesem Strich begrenzten Feld zugeteilt; bleibt sie auf einer senkrechten Teilung liegen, so soll sie dem anschließenden rechten Feld zugeteilt werden. Auf diese Weise erreichen wir, dass die Kugel immer eindeutig einem einzigen Feld zugeordnet werden kann.

Den Vorgang, »die Kugel bleibt im Feld x liegen«, nennen wir ein Elementarereignis. Diese Bezeichnung wählen wir auch für die einzelnen Felder der Platte. Offensichtlich stellt dann die gesamte Platte die Menge aller möglichen Elementarereignisse dar.

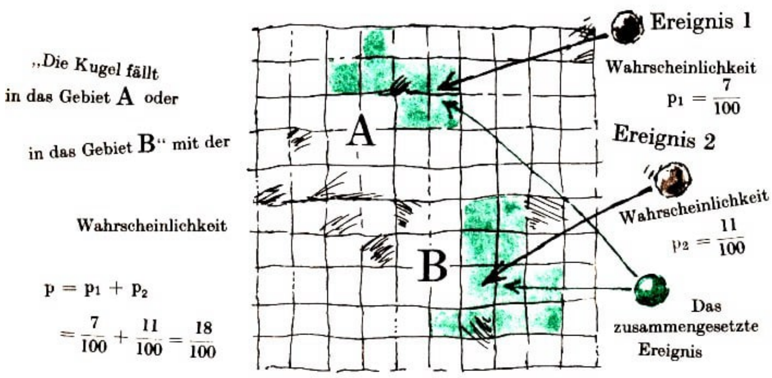


Eine beliebige, aus Feldergrenzen zusammengesetzte Figur soll ein Ereignis genannt werden. Wenn wir vor einem Wurf der Kugel eine solche beliebige Fläche abgrenzen, so ist die Aussicht, dass sich die Kugel nach dem Wurf darin befindet, um so größer, je mehr Einzelquadrate die Fläche umschließt. Wir benutzen demnach die Anzahl der Einzelquadrate als Maß für die Erwartung, die Kugel innerhalb des gewählten Gebietes zu finden.
Das größtmögliche Maß ist hierbei die Gesamtzahl der Quadrate unserer Platte. Beziehen wir den Flächeninhalt, der von einer beliebigen Umrandung gebildet wird, auf

die Gesamtzahl der vorhandenen Quadrate, so erhalten wir eine Maßzahl p, die zwischen 0 und 1 liegt und die wir als Wahrscheinlichkeit dafür bezeichnen, dass die Kugel innerhalb der vorgegebenen Umrandung liegen bleibt:

$$p = \frac{\text{Anzahl der Quadrate der schraffierten Fläche}}{\text{Gesamtanzahl der Quadrate}}$$

Dass die Kugel überhaupt auf die Platte fällt, gilt als sicher; für diesen Fall ergibt sich die Wahrscheinlichkeit $p = 1$.
Dagegen soll es unmöglich sein, dass die Kugel nach dem Wurf nicht auf der Platte zu finden ist. Für dieses Ereignis erhält man also $p = 0$.



Zeichnen wir zwei Figuren auf die Platte, die sich nicht gegenseitig durchsetzen, so erhalten wir für die Wahrscheinlichkeit p, dass die Kugel in das Teilgebiet 1 oder in das Teilgebiet 2 fällt, die Summe der beiden Teilwahrscheinlichkeiten p_1 und p_2.
Der berühmte sowjetische Mathematiker Kolmogorow benutzte die gezeigten Eigenschaften der Wahrscheinlichkeit zu ihrer axiomatischen Festlegung.

Gegeben sei eine Menge E, zum Beispiel die Menge der Quadrate unserer Platte, die wir die Menge der Elementarereignisse nennen wollen. Jede Menge E, die aus Elementarereignissen besteht, bezeichnen wir als Ereignis. Eine Maßzahl $P(A)$ der Menge A heißt Wahrscheinlichkeit des Ereignisses A, wenn folgende Eigenschaften gelten.

1. $P(A)$ ist eine Zahl, die zwischen 0 und 1 liegt.
2. Die Menge E aller Elementarereignisse heißt das sichere Ereignis. Die Wahrscheinlichkeit $P(E)$ des sicheren Ereignisses ist 1.
3. Die Wahrscheinlichkeit dafür, dass das Ereignis A oder das Ereignis B eintritt, ist gleich der Summe $P(A) + P(B)$ der Einzelwahrscheinlichkeiten, falls die beiden Ereignisse A und B keine Elementarereignisse gemeinsam haben.

Auf Grund dieser Axiome lässt sich beweisen, dass die Wahrscheinlichkeit des unmöglichen Ereignisses, d. h. eines Ereignisses, das nicht ein einziges Elementarereignis besitzt, 0 sein muss.

Wir erinnern uns vielleicht solcher allgemeiner Erklärungen der Wahrscheinlichkeit wie: »Die Wahrscheinlichkeit ist das Verhältnis der Anzahl der günstigen zur Anzahl der insgesamt möglichen Fälle.«

Kolmogorow hat mit seinem Axiomensystem dem Begriff der Wahrscheinlichkeit eine exakte wissenschaftliche Grundlage gegeben und ihn von allen subjektiven Formulierungen befreit. Auf das Axiomensystem stützen sich alle Theorien und Gesetze der Wahrscheinlichkeitsrechnung und der Statistik.

5.2 Von Lottozahlen, die an der Reihe sind

In einem Gefäß mögen sich 5 weiße und 5 schwarze Kugeln befinden. Es liegen dann 10 Elementarereignisse vor, die darin bestehen, eine beliebige Kugel herauszugreifen. Wir interessieren uns für die Wahrscheinlichkeit des Ereignisses w, dass eine weiße Kugel herausgenommen wird.

$$P_{\text{weiß}} = \frac{5}{10} = \frac{1}{2} \qquad P_{\text{schwarz}} = \frac{5}{10} = \frac{1}{2}$$

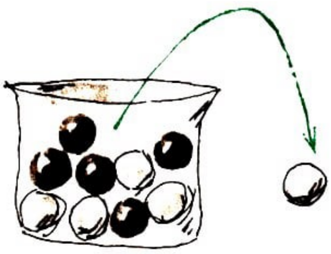
Offensichtlich setzt sich dieses Ereignis aus 5 Elementarereignissen zusammen; die Wahrscheinlichkeit $P(w)$ ist demnach

$$P(w) = \frac{5}{10} = \frac{1}{2}$$

Dasselbe gilt für das Ereignis s, eine schwarze Kugel zu erhalten.

$$P(s) = \frac{5}{10} = \frac{1}{2}$$

Nun soll angenommen werden, dass wir bereits eine weiße Kugel herausgegriffen hätten. Es seien also nur noch 4 weiße Kugeln im Gefäß.



$$P_{\text{weiß}} = \frac{4}{9} \qquad P_{\text{schwarz}} = \frac{5}{9}$$

Die Wahrscheinlichkeit für das Herausgreifen einer weißen Kugel hat sich gegenüber dem ersten Fall geändert und beträgt jetzt 4

$$P_w(w) = \frac{4}{9} \approx 0{,}4444$$

Eine schwarze Kugel wird in diesem Zustand mit der Wahrscheinlichkeit

$$P_w(s) = \frac{5}{9} \approx 0{,}5555$$

gezogen. Das kleine w hinter dem Buchstaben P soll darauf hindeuten, dass zuvor eine weiße Kugel herausgenommen wurde. Wir nennen die Wahrscheinlichkeit $P_w(w)$ die bedingte Wahrscheinlichkeit für das Ziehen einer weißen Kugel unter der Voraussetzung, dass vorher eine weiße Kugel herausgegriffen wurde.
Die Wahrscheinlichkeit $P_w(s)$ ist die bedingte Wahrscheinlichkeit für das Ziehen einer schwarzen Kugel unter der Voraussetzung, dass vorher eine weiße Kugel gezogen wurde.
Wir nennen allgemein die Wahrscheinlichkeit

$$P_B(A)$$

die bedingte Wahrscheinlichkeit für das Eintreffen des Ereignisses A vorausgesetzt, dass zuvor das Ereignis B eingetreten ist.

Hätten wir die weiße Kugel in das Gefäß zurückgelegt, so wären die Wahrscheinlichkeiten $P_w(w)$ und $P_w(s)$ beide unverändert gleich 0,5 geblieben. Diese Tatsache lassen alle Glückstheoretiker unbeachtet, die mit Hilfe der Häufigkeit der aufgetretenen Lottozahlen solche Zahlen suchen, die bei der nächsten Ziehung eigentlich auftreten müssten.
Dabei vergessen sie, dass die Kugeln nach einer abgeschlossenen Ziehung wieder in das Gefäß zurückgelegt werden, dass es sich also nicht um bedingte, sondern um unbedingte Wahrscheinlichkeiten handelt.

Im Zahlenlotto ist die Wahrscheinlichkeit für das Herausgreifen einer bestimmten Zahl, unter der Voraussetzung, dass diese Zahl in der Vergangenheit noch nie gezogen wurde, genauso groß wie die Wahrscheinlichkeit für die Ziehung einer anderen Zahl, die in der Vergangenheit schon oft aufgetreten sein mag, und beträgt immer

$$P = \frac{1}{90}$$

Eine Voraussage der »richtigen« Zahlen im Lotto gelänge uns nur dann mit völliger Bestimmtheit, wenn die an der Glücksmaschine stehende Person die Fähigkeit besäße, jede gewünschte Zahl herauszufinden, und uns mitteilte, welche Zahlen sie auszuwählen gedenkt.
Erst dann wäre sämtliche Ungewissheit über den Ausgang des Experiments von vornherein beseitigt.

Jeder wird zugeben, dass die Ungewissheit im Fall einer ehrlichen Prophezeiung der richtigen Lottozahlen außerordentlich groß ist, und den Grund dafür in der ungeheuren Vielfalt der Auswahlmöglichkeiten sehen. Die Ungewissheit, die über der Entscheidung liegt, ob die erste Person, die wir am Morgen auf der Straße erblicken, eine Frau oder ein Mann ist, scheint wesentlich geringer, weil die Anzahl der Möglichkeiten kleiner ist.

5.3 Die Ungewissheit erhält ein Maß

Ein in Zahlen ausdrückbares Maß für die Unbestimmtheit müsste mit der Anzahl der Möglichkeiten wachsen. Das ist unsere erste Forderung an das gesuchte Maß.

Zu einer weiteren Bedingung führt uns folgende Überlegung. Es soll einmal eine positive

ganze Zahl unter 10 erraten und zum anderen die Augenzahl vorausgesagt werden, die ein gewöhnlicher Würfel nach einem Wurf anzeigt. Diese beiden Experimente haben nichts miteinander gemeinsam, außer dass sie gleichzeitig erfolgen sollen.
Es ist daher angebracht, die Ungewissheit des Experiments »Raten und Würfeln« als Summe der Ungewissheiten der Einzelexperimente aufzufassen.

Schließlich können wir noch feststellen, dass die Ungewissheit von den Wahrscheinlichkeiten der einzelnen Möglichkeiten abhängen muss.
Dazu betrachten wir wieder unser Gefäß mit den 10 Kugeln. Offensichtlich liegt bei einem Inhalt von 5 weißen und 5 schwarzen Kugeln eine größere Ungewissheit vor als bei 9 weißen Kugeln und einer schwarzen. Im letzten Fall sind wir »fast sicher«, eine weiße Kugel zu ziehen. Wir vermuten daher, dass die Ungewissheit im Fall der Gleichwahrscheinlichkeit von Ereignissen ihren jeweils größtmöglichen Wert annimmt.

Formuliert man die gezeigten Eigenschaften der Ungewissheit zu mathematischen Bedingungen, so lässt sich zeigen, dass es nur eine einzige Funktion gibt, die alle Forderungen erfüllt. Die Funktion ist recht einfach und enthält nur die Wahrscheinlichkeiten für die verschiedenen Ausgänge eines Experimentes sowie die Logarithmen dieser Wahrscheinlichkeiten. Sie lautet für einen Versuch mit zwei Ausgängen

$$H(\alpha) = -P_1 \log P_1 - P_2 \log P_2$$

Darin sind P_1 und P_2 die Wahrscheinlichkeiten der einzelnen Versuchsausgänge α_1 und α_2 eines Experiments α. Die Größe $H(\alpha)$ wird absolute oder unbedingte Entropie genannt. Sie ist das quantitative Maß der Ungewissheit, die vor der Ausführung eines Experimentes über dessen Ausgang herrscht.

Betrachten wir noch einmal das Gefäß mit 10 Kugeln, von denen eine gewisse Anzahl n weiß und der Rest $10-n$ schwarz sind, so besteht das Experiment α in der Ziehung einer Kugel. Die möglichen Ausgänge α_1 und α_2 sagen, dass die gezogene Kugel entweder weiß oder schwarz ist.
Für die Wahrscheinlichkeit, eine weiße Kugel zu ziehen, erhält man

$$P(w) = P_1 = \frac{n}{10}$$

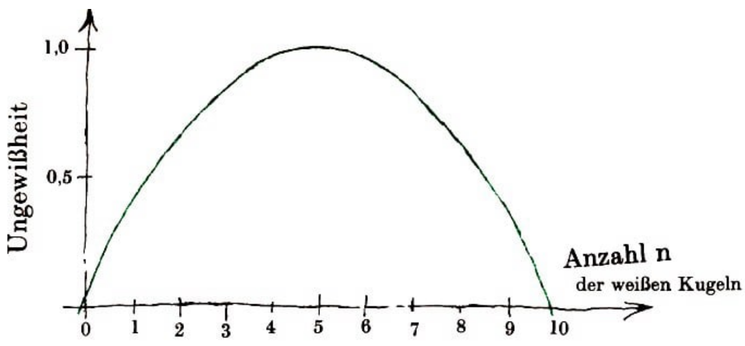
Für die Wahrscheinlichkeit, eine schwarze Kugel herauszugreifen, gilt

$$P(s) = P_2 = \frac{10-n}{n} = 1 - P_1$$

Das Maß der Ungewissheit für dieses Experiment, seine Entropie $H(\alpha)$, wäre demnach

$$\begin{aligned} H(\alpha) &= -P_1 \log P_1 - P_2 \log P_2 = -P_1 \log P_1 - (1-P_1)\log(1-P_1) \\ &= -\frac{n}{10}\log\frac{n}{10} - \left(1-\frac{n}{10}\right)\log\left(1-\frac{n}{10}\right) \end{aligned}$$

Berechnen wir diese Funktion für eine ausreichende Anzahl von Punkten, so erhalten wir eine Kurve, die die Ungewissheit für die richtige Voraussage einer gezogenen Kugel angibt.



Für $n = 0$, d.h. beim Vorliegen lediglich schwarzer Kugeln, nimmt sie den Wert 0 an. Bei gleicher Zahl von weißen und schwarzen Kugeln $n = 5$ erreicht sie ihren höchsten Wert. Für $n = 10$, beim Vorhandensein ausschließlich weißer Kugeln, wird sie wieder 0. Die Ungewissheit über den Ausgang der Ziehung ist also dann am größten, wenn sich gleichviel weiße und schwarze Kugeln im Gefäß befinden.

Jedes Maß verlangt eine Maßeinheit. Wir wissen, dass der Logarithmus einer Zahl derjenige Wert ist, der als Potenz der Basiszahl die Ausgangszahl ergibt. Es ist

$$y = 10^x \quad x = \lg y \quad y = 2^x \quad x = \operatorname{ld} y \quad y = a^x \quad x = \log_a y$$

Wählt man in der Formulierung der Entropie den Logarithmus zur Basis 2, so erhält man eine Darstellung, die sich von Schreibweisen mit anderen Grundzahlen nur quantitativ, aber nicht im Charakter unterscheidet.
Die Wahl der Basis der Logarithmen spielt hier eine ähnliche Rolle wie die Auswahl der Maßeinheiten Zentimeter, Meter oder Kilometer bei der Angabe einer Länge.

Als Maßeinheit der Entropie soll die Ungewissheit des Experimentes mit zwei gleichwahrscheinlichen Ausgängen dienen. Dann gilt

$$H = -\frac{1}{2}\operatorname{ld}\frac{1}{2} - \frac{1}{2}\operatorname{ld}\frac{1}{2} = -\operatorname{ld}\frac{1}{2} = \operatorname{ld} 2 = 1$$

da ja $2^1 = 2$ und damit $1 = \operatorname{ld} 2$ ist. Diese Menge an Ungewissheit ist die Einheit der Ungewissheit und trägt die Bezeichnung 1 bit.

Das Wörtchen bit ist eine Abkürzung des englischen binary digit und bedeutet soviel wie Zweierzahl. Wir werden später noch auf die Zusammenhänge mit den Dualzahlen zurückkommen.
Die Unbestimmtheit ist damit in eine höchst bestimmte, fassbare Größe verwandelt, was uns ermöglicht, diese Größe für jeden Versuch, dessen Ausgänge von Zufall abhängen, zahlenmäßig genau anzugeben.

5.4 Informationen gewinnen heißt Ungewissheit verlieren

Die Untersuchung der Ungewissheit unterstreicht den bereits anfangs geäußerten Gedanken, dass zur Erreichung der völligen Gewissheit über das Eintreten eines Ereignisses soviel »Information« erforderlich ist, wie Ungewissheit vorliegt.

Mit anderen Worten können wir sagen, dass nach der Realisierung eines Experimentes soviel »Information« gewonnen wird, wie Ungewissheit über den Ausgang des Experimentes vor dessen Ausführung vorgefunden wurde.

Vorläufig müssen wir aber den Begriff »Information« in Anführungszeichen setzen, da er noch nicht genau erklärt ist. Wir ahnen aber bereits, dass zwischen der Information und der Ungewissheit oder Entropie ein sehr inniger Zusammenhang bestehen muss.

Die genaue Definition der Information macht eine Erklärung der sogenannten bedingten Entropie erforderlich.
Wir kennen schon die bedingte Wahrscheinlichkeit $P_B(A)$ für das Eintreten des Ereignisses A unter der Voraussetzung, dass zuvor das Ereignis B eingetreten ist. Die bedingte Entropie wird nun ähnlich erklärt wie die absolute Entropie, nur mit dem Unterschied, dass anstelle der absoluten Wahrscheinlichkeiten bedingte Wahrscheinlichkeiten auftreten.

In einem Gefäß mögen sich wieder 5 weiße und 5 schwarze Kugeln befinden. Die Entropie des Versuches α, der darin besteht, eine Kugel herauszugreifen, ist dann wie wir bereits wissen

$$H(\alpha) = -P(w)\,\mathrm{ld}\,P(w) - P(s)\,\mathrm{ld}\,P(s)$$

Wir betrachten wieder den Fall, dass schon eine weiße Kugel gezogen, aber nicht zurückgelegt wurde. Dann ist die bedingte Wahrscheinlichkeit für das Ziehen einer weißen Kugel

$$P_w(w) = \frac{4}{9}$$

und für das Ziehen einer schwarzen Kugel

$$P_w(s) = \frac{5}{9}$$

Die Ungewissheit über den Ausgang der Ziehung hat sich durch die vorherige Ziehung verändert. Sie ist zu einer bedingten Ungewissheit geworden und wird jetzt mit den bedingten Wahrscheinlichkeiten folgendermaßen berechnet:

$$H_w(\alpha) = -P_w(w)\,\mathrm{ld}\,P_w(w) - P_w(s)\,\mathrm{ld}\,P_w(s) = -\frac{4}{9}\,\mathrm{ld}\,\frac{4}{9} - \frac{5}{9}\,\mathrm{ld}\,\frac{5}{9}$$

Diesen Ausdruck formen wir nach den Rechengesetzen für die Logarithmen um und erhalten

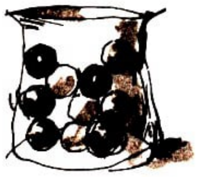
$$H_w(\alpha) = -\frac{4}{9}(\mathrm{ld}\,4 - \mathrm{ld}\,9) - \frac{5}{9}(\mathrm{ld}\,5 - \mathrm{ld}\,9) = \mathrm{ld}\,9 - \frac{4}{9}\cdot\mathrm{ld}\,4 - \frac{5}{9}\cdot\mathrm{ld}\,5$$

Die Werte für die Logarithmen zur Basis 2 entnehmen wir der Tabelle und errechnen schließlich den Wert

$$H_w(\alpha) = 0{,}9911 \text{ bit}$$

x	$\mathrm{ld}\,x$	x	$\mathrm{ld}\,x$	x	$\mathrm{ld}\,x$	x	$\mathrm{ld}\,x$
1	0,0000	2	1,0000	3	1,5850	4	2,0000
5	2,3219	6	2,5849	7	2,8073	8	3,0000
9	3,1699	10	3,3219				

Das ist also die bedingte Entropie für den Versuch, die Farbe einer Kugel vorauszusagen, die aus einem Gefäß mit anfangs 5 weißen und 5 schwarzen Kugeln gezogen wird, wenn bekannt ist, dass eine vorher gezogene Kugel, die nicht wieder zurückgelegt wurde, eine 5 weiße Kugel war. Wir sehen, dass die Menge dieser Ungewissheit geringer ist als die Menge der Ungewissheit beim Vorliegen von 5 weißen und 5 schwarzen Kugeln.
Die Ausführung des vorhergehenden Versuches hat die Ungewissheit des nachfolgenden Versuchs vermindert.

Experiment	Wahrscheinlichkeiten	Ungewissheit
	$P(w) = \frac{5}{10}; P(s) = \frac{5}{10}$	$H = 1{,}0000$ bit
	$P_w(w) = \frac{4}{9}; P_w(s) = \frac{5}{9}$	$H = 0{,}9911$ bit; Information $I = 0{,}0089$ bit

5.5 Die Information wird messbar

Die Differenz zwischen der absoluten Entropie $H(\alpha) = 1{,}0000$ und der bedingten Entropie $H_w(\alpha) = 0{,}9911$ nennen wir nun die Information $I(w,\alpha)$

$$I(w,\alpha) = H(\alpha) - H_w(\alpha) = 1{,}0000 - 0{,}9911 = 0{,}0089$$

die das vorhergehende Ereignis der Ziehung einer weißen Kugel über den Ausgang einer erneuten Ziehung liefert. Um 0,0089 bit ist unser Wissen um das Ergebnis einer Ziehung nach dem Herausgreifen einer weißen Kugel größer geworden.

Jetzt können wir die Information genau definieren: Unter α und β verstehe man zwei Versuche, deren Versuchsbedingungen voneinander abhängig sind, so dass der Ausgang des Versuches β den Ausgang des Versuches α beeinflusst.
Berechnet man die absolute Entropie von α und dazu die bedingte Entropie von α unter der Bedingung, dass der Versuch β durchgeführt wurde, so ergibt die Differenz

$$I(\alpha,\beta) = H(\alpha) - H_\beta(\alpha)$$

diejenige Menge Information, die durch die Realisierung des Versuches β über den Ausgang des Experimentes α gewonnen wird.
Wie die Entropie, so wird auch die Information, die ja nichts anderes ist als eine Differenz von Entropien (Ungewissheiten) in der Einheit bit angegeben.

Die Grundlagen der Informationstheorie wurden im Jahre 1948 vom amerikanischen Nachrichteningenieur C. Shannon veröffentlicht. Shannon zeigte, dass die Information messbar ist, dass man also von einer beliebigen Information sagen kann, wieviel »Informationseinheiten« sie enthält.

Eine Abwandlung der bekannten Redewendung von dem Mann, der 10 Pfund Bücher kaufen wollte, ist jetzt denkbar geworden. Dieser Mann könnte nun 100000 Informationseinheiten verlangen.

Die Entdeckung der Messbarkeit der Information hat eine ähnliche Bedeutung für die Erkenntnis der objektiven Realität wie die Entdeckung der Messbarkeit der Energie. Aus beiden Entdeckungen sind umfangreiche Theorien entstanden. Es gibt Maschinen, die Energie verarbeiten. Wir schicken uns an, Maschinen kennenzulernen, die Informationen verarbeiten.
Viele Erscheinungen lassen sich durch die eine oder die andere Theorie erklären. In den höchsten Entwicklungsformen der Materie treten aber Energie und Information in so inniger Wechselbeziehung auf, dass man versucht ist, eine noch höhere, umfassendere Anschauungsart der Welt zu suchen, in der Energie und Information gleich wichtige Bedeutung haben.

Bis zur jüngsten Gegenwart war die energetische Seite der Lebensvorgänge sehr viel besser bekannt als die informationstheoretische.
Es wundert daher nicht, dass die letztere nun gleichsam aus einem Bedürfnis des Aufholens stärker in den Vordergrund getreten ist. Ob es eine einheitliche Betrachtungsweise der biologischen Prozesse im Sinne einer einheitlichen Theorie, die Information und Energie umfasst, jemals geben wird, ist gegenwärtig kaum erkennbar.

5.6 Rationelles Raten und Rechnen

Warum wurde der Begriff der Information so eingehend besprochen, während wir doch die Absicht hatten, moderne elektronische Rechenautomaten kennenzulernen?
Welcher Zusammenhang besteht zwischen der Information und den Dualzahlen?

Es sei die Aufgabe gestellt, eine ganze Zahl zwischen 0 und 15 mit Hilfe von Fragen zu erraten, die nur mit »ja« oder »nein« beantwortet werden dürfen. Jede der 16 Zahlen

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

kann mit der Wahrscheinlichkeit $p = \frac{1}{16}$ die zu erratende sein. Es liegt demnach eine Ungewissheit von

$$H_{16} = -\frac{1}{16}\,\mathrm{ld}\,\frac{1}{16} - \frac{1}{16}\,\mathrm{ld}\,\frac{1}{16} - ... - \frac{1}{16}\,\mathrm{ld}\,\frac{1}{16}$$

vor, wobei der Summand

$$-\frac{1}{16}\,\mathrm{ld}\,\frac{1}{16}$$

insgesamt 16 mal vorkommt. Damit wird die Entropie

$$H_{16} = -16 \cdot \frac{1}{16}\,\mathrm{ld}\,\frac{1}{16} = \mathrm{ld}\,16 = 4{,}0000\text{bit}$$

Wenn wir uns daran erinnern, dass die Entropie eines Versuches mit 2 gleichwahrscheinlichen Ausgängen

$$H_2 = -\frac{1}{2}\,\mathrm{ld}\,\frac{1}{2} - \frac{1}{2}\,\mathrm{ld}\,\frac{1}{2} = \mathrm{ld}\,12 = 1{,}0000\text{bit}$$

beträgt, und die Beantwortung einer Ja-Nein-Frage als solchen Versuch ansehen, so ist zu schlussfolgern, dass die zu erratende Zahl mit 4 Fragen bestimmt werden kann, wenn diese so gestellt werden, dass die Antworten mit gleicher Wahrscheinlichkeit »ja« oder »nein« lauten können.

Selbstverständlich ist es möglich, durch Ungeschicklichkeit die Anzahl der Fragen zu erhöhen. Wenn wir zum Beispiel gleich am Anfang wissen wollen, ob die gedachte Zahl größer als 12 ist, so ist dies eine solche ungeschickte Frage, denn die Wahrscheinlichkeit dafür, dass die Zahl zwischen 0 und 12 liegt, ist größer als die Wahrscheinlichkeit, dass sie zwischen 13 und 15 zu finden ist, da das erste Zahlenintervall größer als das zweite ist.
Das bedingt, dass die Entropie dieses Experimentes nicht den höchstmöglichen Wert 1 einer Ja-Nein-Entscheidung annehmen kann.

Diese Tatsache ist von Bedeutung für aktive Teilnehmer von Rätselveranstaltungen. Sie müssen sich bemühen, ihre Fragen so zu stellen, dass gleichwahrscheinliche Antworten »ja« oder »nein« über den zu erratenden Gegenstand zu erwarten sind. Nur so wird der Berg Ungewissheit, der vor ihnen liegt, schnell abgetragen.

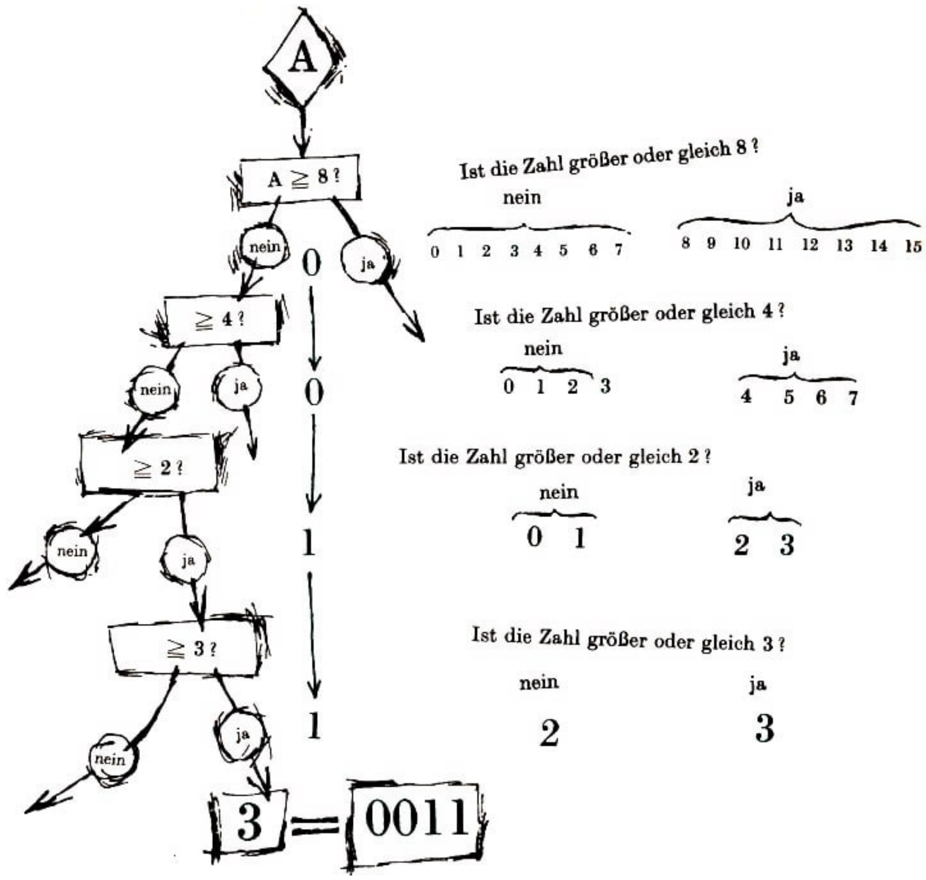
Wir kehren zu unserer Aufgabe zurück und nehmen an, es sei die Zahl 3 zu erraten. Jede gestellte Frage muss das abzusuchende Zahlenintervall in 2 gleich große Teile zerlegen. Als erstes bietet sich daher die Frage an: »Ist die Zahl größer oder gleich 8?«

Die Antwort auf diese Frage zerlegt unser Intervall in 2 gleichgroße Teile. Da die gesuchte Zahl 3 ist, muss die Antwort »nein« heißen.
Nun verbleibt ein Rest von 8 Zahlen, der wiederum in zwei gleichgroße Teile zerlegt werden muss: »Ist die Zahl größer oder gleich 4?«
Diese Frage muss wiederum mit nein beantwortet werden. Nun haben wir nur noch 4 Zahlen zur Auswahl: »Ist die Zahl größer oder gleich 2?«
Diese Frage wird erstmalig mit »ja« beantwortet. Zum Schluss verbleibt nur noch die Entscheidung, ob es sich um die Zahl 2 oder um die Zahl 3 handelt: »Ist die Zahl größer oder gleich 3?«
Die Antwort lautet »ja«. Damit haben wir die Zahl 3, wie verlangt, mit 4 Fragen erraten.

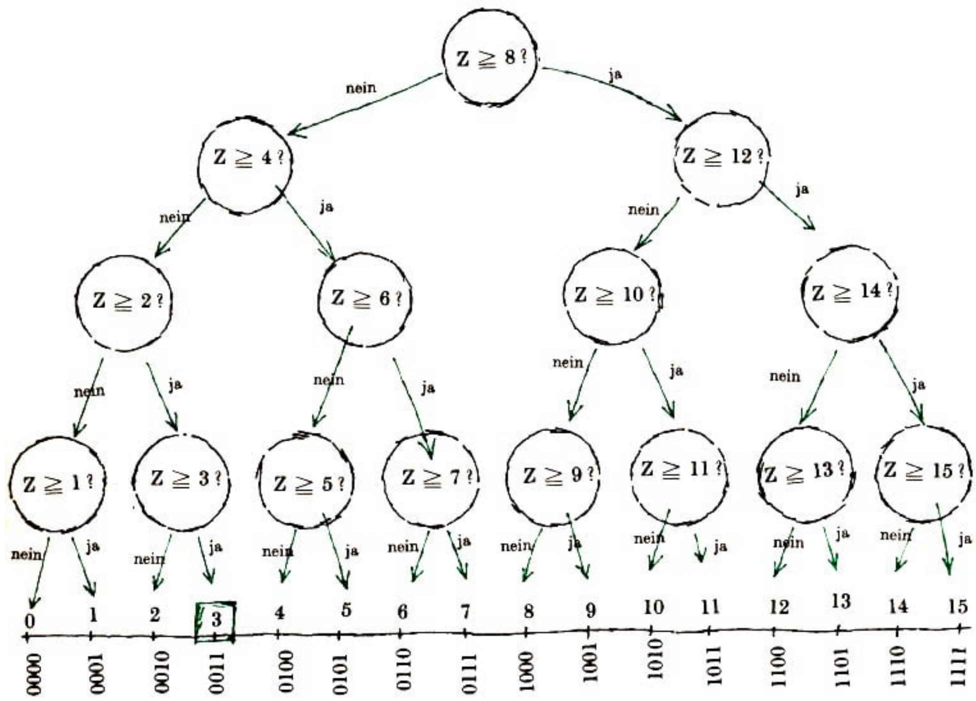
Dabei haben wir auch unbemerkt die Darstellung der Zahl 3 im Dualsystem erhalten. Um das einzusehen, notieren wir zunächst noch einmal alle Antworten auf unsere Fragen in der Reihenfolge, in der sie gegeben wurden: nein, nein, ja, ja.

Wir ordnen dem »nein« die Ziffer 0 zu und dem »ja« die Ziffer 1. Schreiben wir diese Ziffern entsprechend den gegebenen Antworten nieder, so erhalten wir die Darstellung der Zahl 3 im Dualsystem:

$$0011 = 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 3$$



Das ist kein Zufall und kein nur formales Ergebnis. Die Darstellung einer Zahl im Dualsystem entspricht ja einer Gruppenzerlegung nach Potenzen von 2. Jede Ziffer einer Dualzahl stellt also die Antwort auf eine Ja-Nein-Entscheidung dar, wobei die Null dem »Nein« und die Ziffer 1 dem »Ja« entspricht.



In jeder Ziffer einer Dualzahl ist demnach auch die Information einer Ja-Nein-Entscheidung im Werte von 1 bit enthalten. Das war der Anlass, um in den Rechenautomaten die

Anzahl der Dualstellen einer Zahl auch in bit auszudrücken.

Ein Speicher, der einer Dualzahl, die aus 24 Ziffern (0 oder 1) besteht, Platz bietet, besitzt demnach eine Informationskapazität von 24 bit. Wir erkennen jetzt, warum man Rechenautomaten auch als informationsverarbeitende Maschinen ansehen kann.

Mit diesen Bemerkungen zur Informationstheorie haben wir die wichtigsten Kapitel über die theoretischen Grundlagen der modernen elektronischen Ziffernrechner abgeschlossen.
Wir wissen, dass Rechenautomaten Informationen nach der Regie von Algorithmen verarbeiten, wobei die Durchführung der einzelnen Schritte nach den Gesetzen der Logik erfolgt. Bevor wir uns dann mit den mehr technischen Fragen befassen und die oft ans Wunderbare grenzenden Leistungen dieser Maschinen ausführlich betrachten, wollen wir zuvor eine völlig andere Klasse von Rechenautomaten untersuchen, deren Arbeitsweise auf einem Prinzip beruht, dessen Bedeutung weit über das ursprüngliche technische Anwendungsgebiet hinausgeht.

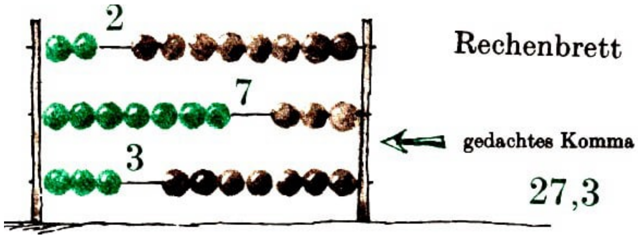
6 Das Rechnen ohne Zahlen

6.1 Ziffer und Vergleich

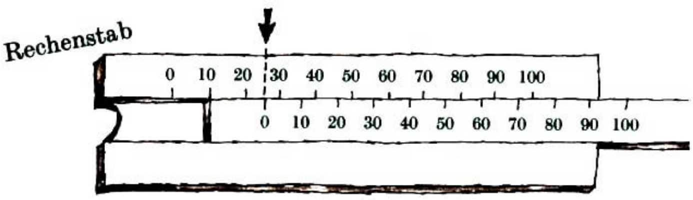
Wenn wir die Länge eines Gegenstandes angeben wollen, so können wir dazu eine Zahl benutzen und etwa »Der Gegenstand ist 45 cm lang« sagen oder auch einen passenden Vergleich suchen und beispielsweise sagen »Der Gegenstand ist so lang wie mein Unterarm«.
Im ersten Fall steht für die Länge eine Zahlenangabe oder - wie man auch sagt - eine digitale Größe, wobei das Wort digital auf die Finger hinweist und das Zählen veranschaulicht. Im zweiten Fall wird die nicht quantitativ gemessene Länge mit einer anderen verglichen. Hier gibt man also eine ähnliche oder analoge Größe an.

Diese beiden Möglichkeiten - Ziffer und Analogie - zur Angabe von Größen finden wir in den zwei wichtigen Arten von Rechenmaschinen wieder, die den obigen Begriffen entsprechend Digitalrechner und Analogrechner heißen.
Für beide Gruppen von Rechenmaschinen existieren einfache Modelle, die uns sehr gut bekannt sind. Das sind der Rechenstab und das Rechenbrett, das praktisch eine kleine Leiter darstellt, auf deren Sprossen Kugeln gleiten.

Wir wollen einmal die Darstellung der Zahl 27,3 auf beiden Geräten betrachten. Auf dem Rechenbrett sehen wir die Zahl in ihrer dezimalen Darstellung nach Zehnern, Einern und Zehnteln geordnet.



Auf dem Rechenstab erscheint die Zahl dagegen als Länge, deren Einheit vorher festgelegt wurde. Diese Länge ist das »Modell« der Größe unserer Zahl. Natürlich kann auch die Darstellung auf dem Rechenbrett als ein Modell der Zahl aufgefasst werden.



Der wesentliche Unterschied zwischen beiden liegt darin, dass das erste Modell ein »kontinuierliches« Abbild des Wertes der Zahl liefert, während das andere eine »diskontinuierliche« Abbildung in Ziffern darstellt.

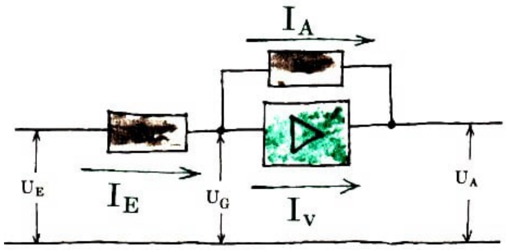
6.2 Das Rechnen mit elektrischen Spannungen

Die modernen Analogrechner bilden die Zahlengrößen auf elektrische Spannungen ab. Wie aber rechnet man mit Spannungen?

Dass sie in einfacher Weise addierbar sind, ist jedem bekannt.

Wir wissen auch, dass Spannungen mit konstanten Faktoren multipliziert werden können, wenn man veränderliche Widerstände vorschaltet. Allerdings erzielen wir auf diese Weise nur Spannungsverkleinerungen, das heißt, wir multiplizieren mit Faktoren, die kleiner als 1 sind.
Der wichtigste Baustein der Analogrechentechnik ist jedoch der elektronische Verstärker. Wir wollen nicht weiter auf seine technische Realisierung eingehen und uns statt dessen seiner Wirkungsweise zuwenden.

Wir betrachten eine Schaltung aus zwei Widerständen R_1 und R_2 und aus dem Verstärker V, der in der Skizze durch ein Dreieck kenntlich gemacht ist. Am Eingang dieser Schaltung herrsche die Spannung U_E. Wir wollen untersuchen, welche Spannung U_A am Ausgang der Schaltung auftritt.



Im Punkt G der Schaltung mündet der Strom I_E, der durch die Eingangsspannung $U - E$, die Spannung U_G im Punkt G und den Widerstand R_1 bestimmt wird:

$$I_E = \frac{U_E - U_G}{R_1}$$

Vom Punkt G gehen zwei Ströme aus. Der eine passiert die Leitung mit dem Widerstand R_2, während der andere über den Verstärker V fließt. Der erste dieser Ströme wird also durch die Spannungsdifferenz $U_G - U_A$ und den Widerstand R_2 bestimmt und hat daher die Größe

$$I_A = \frac{U_G - U_A}{R_2}$$

Der den Verstärker durchfließende Strom hat entsprechend die Größe

$$I_V = \frac{U_G - U_A}{R_V}$$

wobei R_V der Widerstand des Verstärkers ist.
Die bekannte Kirchhoffsche Regel, nach der in einem Verzweigungspunkt die Summe der ankommenden Ströme so groß wie die Summe der abfließenden Ströme sein muss, ergibt

$$I_E = I_A + I_V$$

Setzen wir für die einzelnen Ströme die sich aus den Spannungsdifferenzen und den Widerständen ergebenden Ausdrücke ein, so erhalten wir

$$\frac{U_E - U_G}{R_1} = \frac{U_G - U_A}{R_2} + \frac{U_G - U_A}{R_V}$$

Wir setzen nun voraus, dass der Widerstand des Verstärkers sehr groß ist. In diesem Fall wird der Ausdruck

$$\frac{U_G - U_A}{R_V}$$

sehr klein, falls die Spannungsdifferenz nicht übermäßig groß ist, und wir können ihn in der Formel streichen. Nutzen wir die Eigenschaft des Verstärkers aus, dass seine Ausgangsspannung ein Vielfaches seiner Eingangsspannung beträgt, so gilt

$$U_A = K \cdot U_G$$

Lösen wir diese einfache Beziehung nach U_G auf und setzen den sich für U_G ergebenden Ausdruck in die ursprüngliche Gleichung ein, so entsteht die Beziehung

$$\frac{U_E - \frac{U_A}{K}}{R_1} = \frac{\frac{U_A}{K} - U_A}{R_2}$$

In dieser Gleichung fehlt bereits das als vernachlässigbar klein erkannte Glied, das den Strom I_V darstellte. Nun fordern wir vom Verstärker einen sehr hohen Verstärkungsgrad K. Dann ergibt sich ein sehr kleiner Wert für den Quotienten

$$\frac{U_A}{K}$$

der jetzt gegenüber den Größen U_E und U_A nicht mehr ins Gewicht fällt. Aus diesem Grunde vereinfachen wir die Formel zu

$$\frac{U_E}{R_1} = -\frac{U_A}{R_2}$$

Schließlich lösen wir diese Gleichung nach U_A auf und erhalten für die gesuchte Abhängigkeit der Ausgangsspannung von der Eingangsspannung die Beziehung

$$U_A = -\frac{R_2}{R_1} \cdot U_E$$

Die Verstärkerschaltung gibt uns also die Möglichkeit, die Eingangsspannung mit beliebigen Faktoren $-\frac{R_2}{R_1}$ zu multiplizieren. Es kommt nur darauf an, die beiden Widerstände R_2 und R_1 geeignet zu wählen.

Dies wird im sogenannten Multiplikator des Analogrechners ausgenutzt, bei dem mit Hilfe der Verstärkerschaltung eine Eingangsspannung mit einem einstellbaren beliebigen Faktor multipliziert werden kann. Die Beschränkung, dass dieser Faktor kleiner als 1 sein müsse, konnten wir fallenlassen.

Das negative Vorzeichen des Faktors überrascht uns zunächst; es scheint beim Rechnen auf Analogrechenmaschinen sehr hinderlich zu sein. Diese kleine Ungereimtheit wird aber schnell beseitigt, wenn man sogenannte Inverter einschaltet, die aus einer gegebenen Eingangsspannung eine dem Absolutbetrag gleichgroße, aber mit dem entgegengesetzten Vorzeichen versehene Spannung bilden.

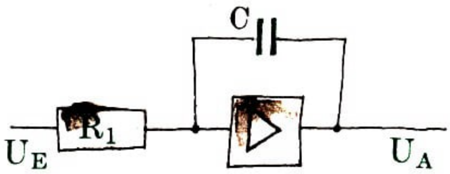
Eine Kombination einiger Multiplikationen mit einer einfachen Summenbildung ergibt den sogenannten Additor oder Summator. Dieses Rechenelement bildet aus zwei Eingangsspannungen U_{E_1} und U_{E_2} eine Ausgangsspannung

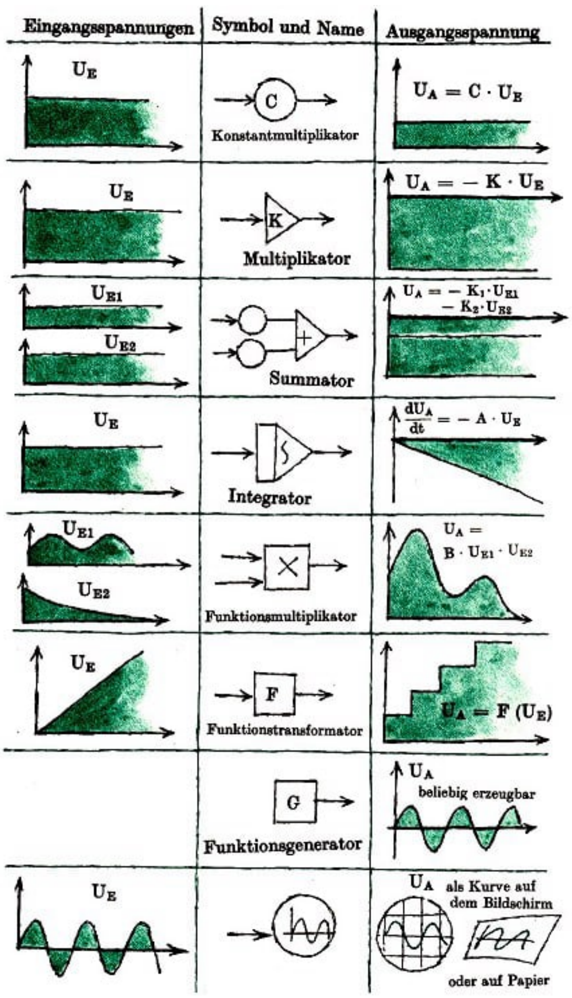
$$-U_A = K_1 \cdot U_{E_1} + K_2 \cdot U_{E_2}$$

also eine Summe, bei der jeder Summand mit beliebig einstellbaren Konstanten multipliziert wird.

6.3 Anstelle umfangreicher Programme

Hierin liegt ein wichtiger Unterschied des Analogrechners gegenüber dem noch zu besprechenden Ziffernrechner. Der Ziffernrechner kennt nur die »klassischen« Grundrechenarten Addition, Subtraktion, Multiplikation und Division. Solche kombinierten Additionen, wie sie beim Summator des Analogrechners als fertige Rechenbausteine vorliegen, erfordern beim Ziffernrechner ein Programm, das das gewünschte Verhalten aus den Grundrechenoperationen zusammensetzt.





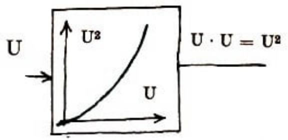
Das wird noch deutlicher, wenn wir den sogenannten Integrator des Analogrechners betrachten. Er entsteht, wenn wir in der Multiplikatorschaltung an die Stelle des Widerstandes R_2 einen Kondensator C setzen.

Geben wir auf den Eingang des Integrators eine konstante Spannung, so erhalten wir am Ausgang des Bausteines eine Spannung, die mit der Zeit linear ansteigt. Der Integrator summiert fortlaufend die Eingangsspannung, er bildet also mathematisch gesehen das Integral der Eingangsgröße.
Wollen wir die Integration einer Funktion auf einem Ziffernrechner durchführen, so ist dazu bereits ein recht umfangreiches Programm zu schreiben, während auf dem Analogrechner ein kompletter Rechenbaustein zur Verfügung steht.

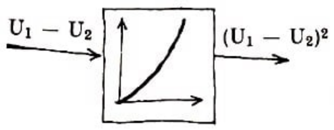
Während die Multiplikation einer Spannung mit einem konstanten Faktor bereits erklärt ist, müssen wir noch die Multiplikation einer Spannung U_1 mit einer anderen Spannung U_2 beschreiben, die Multiplikation zweier sich zeitlich ändernder Größen oder, mathematisch ausgedrückt, die Multiplikation zweier Funktionen $U_1(t)$ und $U_2(t)$ der Zeit t. Der Rechenbaustein, der diese Aufgabe löst, soll den Namen Funktionsmultiplikator erhalten.

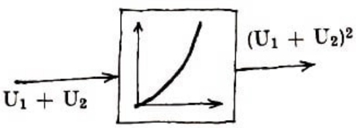
Es zeigt sich, dass die Multiplikation zweier Spannungen nur über einen kleinen Umweg möglich ist. Das Herzstück des Funktionsmultiplikators ist ein Netzwerk aus Dioden und Widerständen mit einer quadratischen Kennlinie, das sogenannte Quadrierelement.

Bilden wir durch Summatoren die beiden Ausdrücke $U_1 - U_2$ und $U_1 + U_2$ und geben sie auf den Eingang je eines Quadrierelementes, so erhalten wir durch Bildung des Quadrates der Differenz und des Quadrates der Summe der zwei Spannungen die beiden Ausgänge $(U_1 - U_2)^2$ und $(U_1 + U_2)^2$.

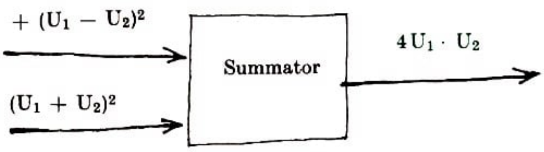


Das Quadrierelement

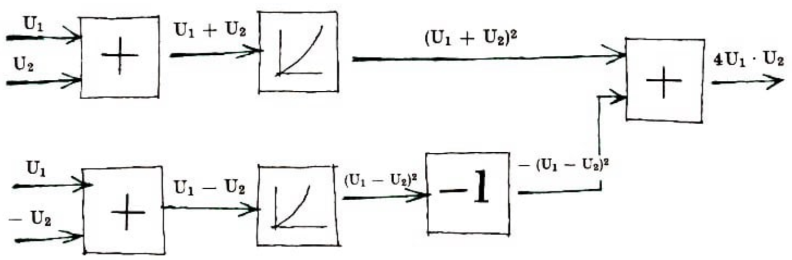




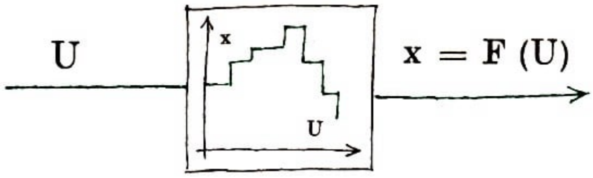
Bilden wir dann mit Hilfe eines Summators die Differenz dieser beiden quadratischen Ausdrücke, so entsteht an dessen Ausgang eine Spannung, die dem Produkt $U_1 \cdot U_2$ proportional ist. Es ist leicht zu überprüfen, dass die Ausgangsgröße das Vierfache des gesuchten Produktes ist.



Insgesamt ergibt sich für den Funktionsmultiplikator die nachstehende Prinzipschaltung, wobei die Vorzeichenänderung beim Summator außer acht gelassen wurde.

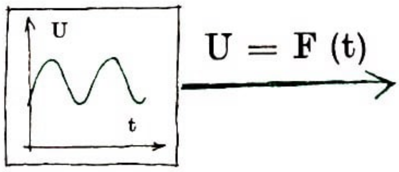


Ähnlich wie die quadratische Kennlinie kann man nun jede beliebige Funktion mit Hilfe von Dioden und Widerständen näherungsweise nachbilden. Der dazugehörige Rechenbaustein des Analogrechners heißt Funktionstransformator. Er besitzt eine Reihe von Schalt- knöpfen, von denen jeder die Lage einer kleinen geraden Strecke bestimmt, aus denen die gewünschte Kurve aufgebaut wird.



Zu beachten ist hierbei, dass das Eingangssignal des Funktionstransformators eine zeitlich veränderliche Spannung ist, die am Ausgang als eine in der gewünschten Weise umgewandelte oder transformierte zeitlich veränderliche Spannung erscheint. Es handelt sich damit um eine Transformation einer Funktion $U(t)$ in eine Funktion $F(U(t))$.

Der Analogrechner kennt aber auch Bausteine, die keine Funktionen umwandeln, sondern Funktionen erzeugen.
Solche Bausteine heißen Funktionsgeneratoren; sie haben keinen Eingang, es sei denn, dass man die unabhängige Veränderliche »Zeit t« als Eingangsgröße definiert.



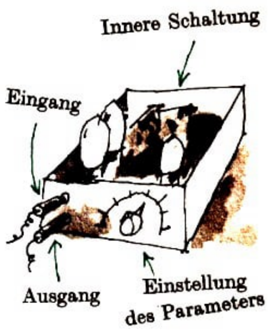
Ein Funktionsgenerator dient beispielsweise zur Erzeugung einer fortlaufend gleichmäßigen Schwingung, die irgendwo während der Rechnung als Eingangsgröße eines anderen Rechenbausteins benötigt wird.
Die Liste der Rechenbausteine des Analogrechners wird schließlich durch ein Ausgabegerät vervollständigt, das ein Oszillograf sein kann, auf dem die Lösungskurve als stehendes Bild erscheint, oder ein Kurvenschreiber, der die Lösung automatisch niederschreibt.

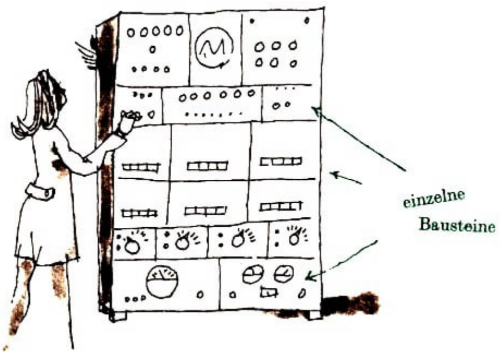
6.4 Die Addition im Schubfach

Auf Seite 56 sind die Symbole der Rechenbausteine des Analogrechners sowie ihre Beziehungen zwischen dem Ein- und Ausgang zur Übersicht in einer Tabelle dargestellt. Der Analogrechner besitzt für jedes Symbol dieser Liste eine Anzahl konkreter Ausführungen, die durch elektrische Kabel miteinander verbunden werden. Jeder Baustein ist in einem Einschub untergebracht, der an seiner Stirnseite eine Steckbuchse für das Kabel der Eingangsspannung und eine für das Kabel der Ausgangsspannung besitzt.
Benötigt ein Rechenelement mehrere Eingangsspannungen, so ist natürlich auch die entsprechende Anzahl von Steckbuchsen vorhanden. Des weiteren finden wir auf dem Einschub einen oder mehrere Drehknöpfe zur Einstellung der erforderlichen oder gewünschten Werte für die wählbaren Konstanten.

Einen Analogrechner stellen wir uns am besten als einen Schrank mit mehreren Schubfächern vor. Jedes Schubfach beinhaltet eine der genannten Grundoperationen des Rechners. Dabei muss jeder Rechenbaustein so oft vorhanden sein, wie er in der zu bearbeitenden Aufgabe benötigt wird. Sind also beispielsweise Multiplikationen von 4 verschiedenen Spannungen mit 4 Faktoren durchzuführen, so brauchen wir dazu vier

Multiplikatoren. Die Lösbarkeit einer Aufgabe auf dem Analogrechner hängt offensichtlich sehr stark von dessen Größe ab.





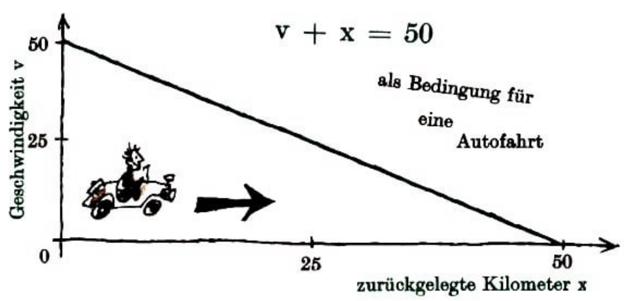
Nach der Anzahl der Bauelemente unterscheidet man verschiedene Größenklassen von Analogrechenmaschinen. Die Anzahl der Verstärker ist die wichtigste Kennzahl.
Kleine Tischrechner, die nur einige wenige Verstärker besitzen, sind auch nur zur Lösung wenig umfangreicher Probleme geeignet. Mittlere Analogrechner können 15 bis 30 Verstärker vorweisen und stellen die am weitesten verbreitete Größenklasse dar. Es sind aber auch Großrechenanlagen bekannt, die sich aus mehr als 200 Verstärkern aufbauen.

6.5 Viele kleine Schritte

Der erste Vorschlag, einen Analogrechner zu bauen, wurde von Lord Kelvin im Jahre 1876 gemacht. Er wollte eine Maschine konstruieren, die Differentialgleichungen lösen sollte. Bis heute ist die Lösung von Differentialgleichungen das wichtigste Anwendungsgebiet der Analogrechner geblieben.

Eine Differentialgleichung ist eine Gleichung, die neben der gesuchten Unbekannten auch die Änderungstendenzen dieser Unbekannten in Bezug auf die Größen enthält, von denen sie abhängig ist. Eine solche Gleichung enthält mathematisch gesprochen Differentialquotienten.
Treten in einer Differentialgleichung nur Differentialquotienten nach einer einzigen unabhängigen Variablen auf, so nennt man diese Gleichung eine gewöhnliche Differentialgleichung. Der Analogrechner ist für die Lösung gewöhnlicher Differentialgleichungen am besten geeignet. Besonders häufig werden Probleme behandelt, in denen die Zeit als unabhängige Variable auftritt. Das sind Schwingungsprobleme, Aufgaben aus der Regelungstechnik und vieles andere mehr.

Die Lösung einer solchen Differentialgleichung bestimmt die Unbekannte nicht als Zahl, sondern als eine sich zeitlich ändernde Größe, also als eine Funktion der Zeit.

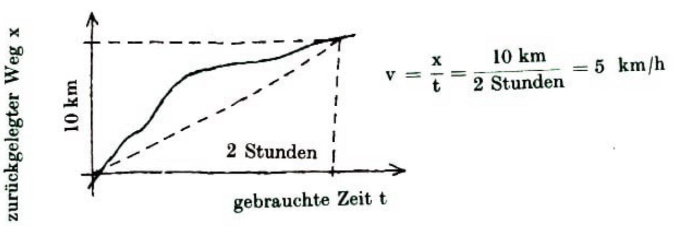


Wir betrachten einen Autofahrer, der die Aufgabe erhält, so zu fahren, dass die Summe aus der in Kilometern pro Stunde angegebenen Geschwindigkeit v und der zurückgelegten Kilometerzahl x immer den Wert 50 hat. Der Start soll beim Kilometer 0 erfolgen. Daraus folgt, dass die Geschwindigkeit in diesem Punkt den Wert 50 haben muss. Das lässt sich leicht realisieren, wenn vor der Startlinie eine genügend lange Anlaufstrecke vorhanden ist, die das Erreichen der Startlinie mit der geforderten Geschwindigkeit ermöglicht.

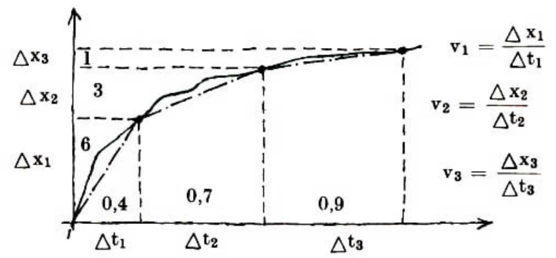
Das entspricht in der Ausdrucksweise der Differentialgleichung der Festlegung der Anfangsbedingungen. Von nun an fährt der Fahrer unter der Bedingung $v + x = 50$, d.h., er muss durch ständige Beobachtung des Kilometerzählers und des Geschwindigkeitsmessers die Geschwindigkeit so verändern, dass diese Bedingung stets erfüllt ist.
Wenn wir eine Strecke x in der Zeit t durchfahren, so sagen wir, dass wir diese Strecke mit der Geschwindigkeit

$$v = \frac{x}{t}$$

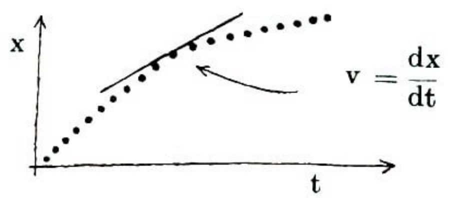
zurückgelegt haben. Diese Aufgabe ist ungenau.
Zwischen den Zeitpunkten 0 und t können durchaus Geschwindigkeitsschwankungen aufgetreten sein. Es ist sicher exakter, wenn wir den zurückgelegten Weg in Teilstrecken zerlegen und von jedem Wegabschnitt sagen, in welcher Zeit bzw. mit welcher Geschwindigkeit wir ihn zurückgelegt haben.



Aber auch das reicht nicht aus. Auch in diesen kleinen Intervallen ist die Geschwindigkeit im allgemeinen nicht immer dieselbe.



Wir müssen durch Verkleinerung der Intervalle schließlich zu einem Grenzwert übergehen, der die Geschwindigkeit in dem gerade betrachteten Augenblick angibt.



Das ist aber der Ausdruck, der in der Mathematik als Differentialquotient des Weges nach der Zeit bezeichnet und in der Form

$$v = \frac{dx}{dt}$$

geschrieben wird. Dieser Ausdruck ist dann in jedem Punkt t bekannt und stellt selbst wieder eine Funktion der Zeit dar.
Wir können jetzt die Bedingung unserer Autofahrt in Form einer Differentialgleichung niederschreiben:

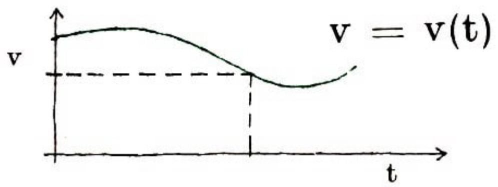
$$\frac{dx}{dt} + x = 50$$

Diese Differentialgleichung ist gelöst, wenn es uns gelingt, den zurückgelegten Weg x in Abhängigkeit von der Zeit t, also als Funktion

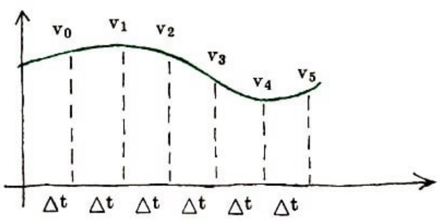
$$x = x(t)$$

darzustellen. Wir wollen zeigen, wie man die Differentialgleichung mit Hilfe eines Analogrechners lösen kann.

Nehmen wir einmal an, dass wir die Geschwindigkeit eines Fahrzeuges in jedem Zeitpunkt seiner Fahrt kennen, so ist es möglich, die Geschwindigkeit v als Funktion der Zeit t in ein Schaubild eint zutragen.



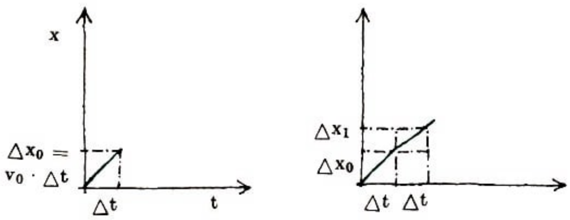
Aus dem Verlauf der Geschwindigkeit lässt sich dann der zurückgelegte Weg in Abhängigkeit von der Zeit berechnen. Dazu wird der gesamte betrachtete Zeitraum in kleine Zeitintervalle zerlegt.



Ist die Zerlegung nur genügend fein gewählt, so darf man annehmen, dass kein allzu großer Fehler entsteht, wenn jetzt wie folgt vorgegangen wird. Am Anfang hat das Fahrzeug die Geschwindigkeit v_0. Es legt demnach in der kleinen Zeit Δt die kleine Weglänge

$$\Delta x_0 = v_0 \cdot \Delta t$$

zurück. Wir zeichnen diesen Wegzuwachs in ein weiteres Schaubild, das den zurückgelegten Weg in Abhängigkeit von der Zeit darstellen soll.

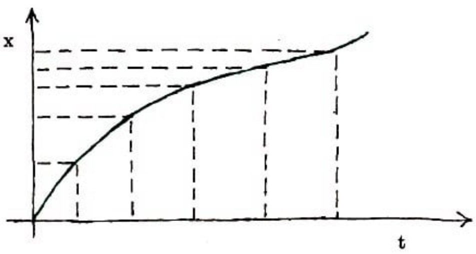


Nach dem Ablauf der Zeit Δt erreicht das Fahrzeug die neue Geschwindigkeit v_1 mit der es die kleine Zeit Δt lang weiterfährt. Dabei wird eine Strecke von

$$\Delta x_1 = v_1 \cdot \Delta t$$

zurückgelegt. Diesen neuen Streckenzuwachs zeichnen wir wieder in die graphische Darstellung und erhalten so nach vielen Wiederholungen des Vorganges die gewünschte Darstellung des zurückgelegten Weges als die Summe vieler kleiner Teilwege in Abhängigkeit von der Zeit.

Dabei ist zu beachten, dass wir willkürlich von dem Weg 0 ausgegangen sind. An unserer Darstellung hätte sich aber auch nichts geändert, wenn wir einen bestimmten, bereits zurückgelegten Weg zugrunde gelegt hätten.



Der Integrator als Baustein des Analogrechners vollzieht nun fast genau dieselbe Prozedur, die wir eben beschrieben haben. Der einzige Unterschied besteht darin, dass er es kontinuierlich tut und folglich sein Ausgangssignal nicht aus aneinandergefügten geradlinig verlaufenden Spannungen besteht, sondern einen »glatten« Charakter hat.

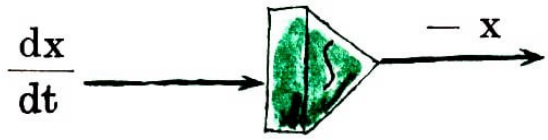
Der Integrator bildet aus dem Differentialquotienten $\frac{dx}{dt}$ die Größe x selbst wieder zurück. Da er aber mit einem elektronischen Verstärker arbeitet, liefert er die Größe mit einem entgegengesetzten Vorzeichen als $-x$.

6.6 Ein Kreis wird geschlossen

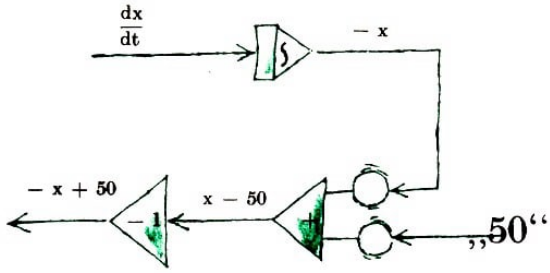
Nun können wir endlich daran gehen, einen Schaltplan für die zu lösende Aufgabe aufzustellen. Dazu schreiben wir die Differentialgleichung unserer seltsamen Autofahrt in der etwas abgeänderten Form

$$\frac{dx}{dt} = -x + 50$$

Wir wissen bereits, dass der Integrator aus dem Eingangssignal $\frac{dx}{dt}$ das Ausgangssignal $-x$ bildet:



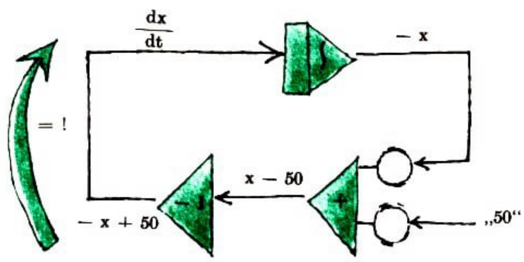
In die Differentialgleichung geht die konstante Größe 50 ein. Wir benötigen daher eine konstante Spannung, die der Zahl 50 entsprechen soll. An jedem Analogrechner sind die Entnahmestellen für konstante Spannungen vorhanden.
In unserem Fall muss die konstante Spannung »50« laut Differentialgleichung zu der negativen Kilometerzahl x addiert werden. Wir müssen demnach den Ausgang des Integrators und die konstante Spannung »50« mit einem Summator addieren.



Am Ausgang des Summators erscheint $x - 50$, weil auch der Summator die Vorzeichen vertauscht. Durch das Einschalten des Inverters gelangen wir aber zu der geforderten Größe $-x + 50$.
Nun fällt auf, dass wir den Inhalt der Differentialgleichung sofort realisieren können, wenn wir den Ausgang des Inverters mit dem Eingang des Integrators verbinden. Wir setzen, wie es die Gleichung verlangt, die beiden Werte

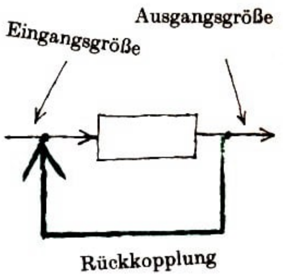
$$\frac{dx}{dt} \qquad \text{und} \qquad -x + 50$$

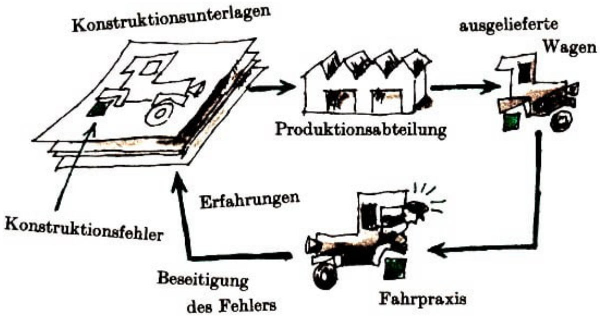
einander gleich. Jetzt entsteht ein geschlossener Kreis, der die Spannung zwingt, sich in der von der Differentialgleichung geforderten Weise zu verhalten.



Der Koppelplan stellt also das Programm dar, nach dem die Spannungen auf dem Analogrechner verlaufen müssen. Hierbei haben wir das wichtige Prinzip der Rückkopplung kennengelernt. Man versteht darunter die Beeinflussung der Eingangsgröße irgendeines Wirkungsgefüges durch dessen Ausgangsgröße.

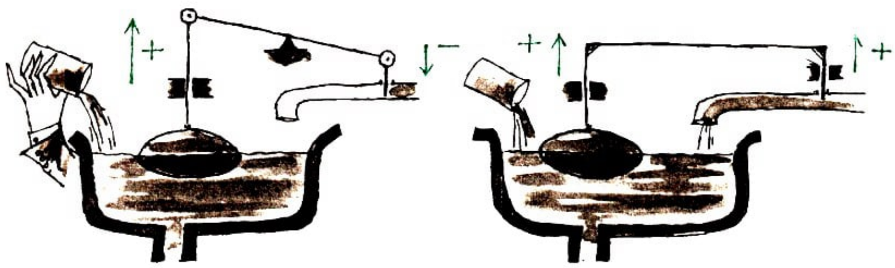
Betrachten wir beispielsweise die Produktionsabteilung eines Fahrzeugwerkes, das nach vorgegebenen Konstruktionsunterlagen Lastwagen produziert. Unterstellt man, wie das in unserem Bild geschehen ist, dass dem Konstrukteur ein grober Fehler unterlief, so wird sich dieser Fehler spätestens in der Fahrpraxis herausstellen und auf die Konstruktionsunterlagen rückwirken: Der Konstrukteur wird das viereckige Rad durch ein rundes ersetzen.





Der Konstruktionsfehler war eine Störung, die Fahrpraxis wirkte dieser Störung entgegen, sie veranlasste die Beseitigung des Fehlers.

Dieses Prinzip hat den Namen »negative Rückführung« oder »negative Rückkopplung« erhalten. Das Beiwort »negativ« wurde gewählt, weil die Rückwirkung eine aufgetretene Störung des normalen Verhaltens wieder rückgängig zu machen sucht.
Das Wörtchen »negativ« ist hier kein Werturteil, es ist eine rein technische Bezeichnung. Die negative Rückkopplung hat einen stabilisierenden Einfluss auf das Gesamtverhalten eines Systems.



Das soll uns ein weiteres Beispiel verdeutlichen. Auf der Oberfläche einer Flüssigkeit in einer Wanne, der durch ein Rohr weitere Flüssigkeit zugeführt wird, befindet sich ein Schwimmer, der über ein Hebelsystem das Ventil der Zuleitung so steuert, dass bei Erhöhung des Flüssigkeitsspiegels eine verminderte Zuführung von Flüssigkeit erfolgt. Gießen wir in diese Wanne ein Glas Wasser, so wird das die plötzlich eingetretene Volumenvergrößerung mit einer Drosselung der Flüssigkeitszufuhr beantworten und damit nach kurzer Zeit die Wiederherstellung des alten Gleichgewichtszustandes bewirken.

Das Gegenstück der negativen Rückführung, die positive Rückkopplung, hat dagegen keinen stabilisierenden Einfluss.
Ersetzt man in unserem Beispiel das Hebelsystem durch ein starres Gestänge, so erhält man ein System, das jede auftretende Störung verstärkt. Gießt man jetzt in die Wanne ein Glas Wasser, so erhöht sich der Flüssigkeitsspiegel, der Schwimmer steigt und gibt damit den Weg für zusätzlich einströmende Flüssigkeit frei. Das Gefäß wird infolgedessen überlaufen.

So schädlich der Effekt der positiven Rückführung im gewählten Beispiel ist, so nützlich erweist er sich in anderen Fällen, wo eine Verstärkung schwacher Eingangssignale benötigt wird, wie bei der Schaltung des elektronischen Verstärkers.

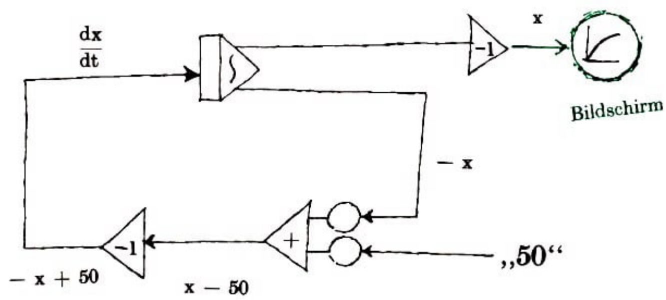
Mit den genannten Beispielen haben wir zwei einfache Aufgaben aus der Regelungstechnik kennengelernt. Solche Aufgaben sind sehr gut geeignet, auf Analogrechenmaschinen gelöst zu werden.
Bei der Regelung einer Dampfturbine kommt es z. B. darauf an, nach der Abschaltung des Generators den Anstieg der Drehzahl der entlasteten Turbine in gewissen Grenzen zu halten, um eine Havarie der Maschine zu vermeiden. Die Dampfventile müssen durch starke hydraulische Stellmotoren rechtzeitig geschlossen werden.
Dieser Schließvorgang und das zeitliche Verhalten des Turbinenläufers sowie die Energie des einströmenden Dampfes ergeben - mathematisch formuliert - ein System von Differentialgleichungen, das auf dem Analogrechner gelöst werden kann.

Daraus ergeben sich die benötigten Schlussfolgerungen für die Konstruktion des Regelungsmechanismus.
Nach diesen Betrachtungen wollen wir zu unserer seltsamen Autofahrt zurückkehren.

6.7 Wenn man keine Maßstäbe setzt

Der Koppelplan des Problems wird auf den Analogrechner übertragen.
Man verbindet die Ein- und Ausgänge der benötigten Rechenbausteine des Automaten durch Kabelschnüre und stellt so ein gerätetechnisches Abbild des Schaltplanes her. An jeder gewünschten Stelle kann auch das Ausgabegerät angeschlossen werden.

Da wir uns für den Verlauf des zurückgelegten Weges x interessieren, schalten wir das Sichtgerät hinter den Ausgang des Integrators und fügen einen Inverter ein, um das positive x zu erhalten. Gerätetechnisch ist das so verwirklicht, dass jeder Baustein des Rechners mehrere Ausgangsleitungen besitzt, an denen jeweils dieselbe Ausgangsgröße erscheint.

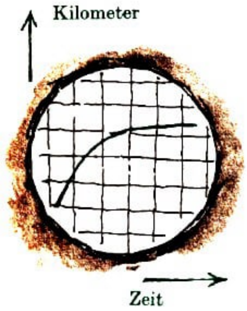


Dann ist es soweit. Wir betätigen den Startknopf und wenden uns hoffnungsvoll dem Bildschirm zu, auf dem die Lösung erscheinen soll. Unsere Enttäuschung ist groß. Auf dem Oszillografen sehen wir Vorgänge, die unmöglich die Lösung der gestellten Aufgabe sein können.
Haben wir eine gedämpfte Schwingung erwartet, so kann uns ein fast geradliniges Kurvenstück überraschen. Unter Umständen bleibt aber der Bildschirm sogar völlig leer.

In der Tat haben wir eine wichtige Seite des Rechnens auf Analogiegeräten völlig außer

acht gelassen. Wir haben auf den letzten Seiten so getan, als ob auf dem Analogrechner wirklich mit der Kilometerzahl x, mit der Geschwindigkeit v und mit der Zahl 50 gerechnet wird.
Das ist natürlich nicht der Fall.

Das Analogiegerät rechnet ja nicht mit Zahlen, sondern mit einer ganz bestimmten, dimensionsbehafteten Größe, mit der Spannung. Man muss also gewisse Maßstabsfaktoren einführen, deren Zusammenhänge in Gleichungen festhalten und diese Gleichungen auflösen.
Dann ergeben sich für alle Faktoren zahlenmäßig bestimmte Werte.
Diese berechneten Werte werden am Analogrechner mit Hilfe von Drehknöpfen und Schaltern an den entsprechenden Rechenbausteinen eingestellt. Wenn wir jetzt den Start der Rechnung befehlen, können wir sicher sein, auf dem Bildschirm oder am Kurvenschreiber das gewünschte Ergebnis zu erhalten.



Während die Aufstellung des Koppelplanes in den meisten Fällen recht einfach vonstatten geht und einiges Vergnügen bereitet, ist die Bestimmung der zu wählenden Maschinenkonstanten insbesondere bei größeren Aufgaben oft sehr mühevoll.

Neben den universellen Analogrechnern, wie sie bisher besprochen wurden, existieren speziellere Maschinen für die Modellierung des dynamischen Verhaltens komplizierter technischer und biologischer Systeme. Man kennt Simulatoren für Kernreaktoren und Flugtriebwerke. Das sind Analogrechner, bei deren Konstruktion bereits die typischen Funktionsweisen der nachzubildenden Geräte berücksichtigt werden, die demnach eine ganze Schar von Maschinen verkörpern.
Eine konkrete Ausführung eines Kernreaktors ist dann durch die Festlegung aller seiner Parameter bestimmt. An solchen Modellen lassen sich Experimente unter angenommenen extremen Bedingungen der Maschinen schnell, billig und gefahrlos durchführen. Auf diesen Simulatoren darf ein Flugtriebwerk in der Luft explodieren, während der Ingenieur dicht daneben interessiert zusieht.
Nach der Betrachtung der Analogrechner wollen wir uns jetzt dem programmgesteuerten Ziffernrechner zuwenden.

7 Die Anatomie eines Elektronenhirns

7.1 Rechnen allein genügt nicht

Zu den Ziffernrechnern gehören auch die bekannten Tischrechenmaschinen. Angefangen bei der einfachen Handkurbelmaschine bis hin zum Vollautomaten für die 4 Grundrechenarten haben diese Maschinen allerdings nur wenig mit dem Prinzip der Programmsteuerung gemeinsam.

In einem System von Zahnrädern und Hebeln werden sowohl die einzelnen Ziffern als auch die Algorithmen der Grundrechenarten fest verankert. Die gewünschten Zahlen werden mit Hilfe von Tasten eingedrückt; in der gleichen Weise werden die durchzuführenden Operationen ausgelöst. Nach jeder abgearbeiteten Rechenoperation hält die Maschine an.

Ein sinnvolles Arbeiten dieser Maschine ist nur in Verbindung mit dem sie bedienenden Menschen möglich, der die jeweils benötigten Zahlen eingeben, die Operation auslösen und das Ergebnis: notieren muss.

Worin besteht hierbei der Fortschritt gegenüber dem Rechnen mit Kopf, Papier und Bleistift? Überlegen wir uns, wie das Kopfrechnen eigentlich vonstatten geht, so erkennen wir, dass wir die Algorithmen der Grundrechenarten auswendig wissen, sie also irgendwo im Gehirn gespeichert haben. Dabei sind wir in der Lage, ohne wirkliche Gedankenarbeit zu rechnen.
Die Gesetze der Addition von Ziffern, des Übertrags auf die nächsthöhere Stelle und das kleine Einmaleins sind in unserem Gehirn durch jahrelange Übung bereits »fest verdrahtet«. Es ist dann nur ein kleiner Schritt, wenn wir diese Verdrahtung auf eine Maschine übertragen und so zur Tischrechenmaschine gelangen.

Bereits um 1637 versuchte Schickard eine erste Rechenmaschine für die vier Grundrechenarten zu bauen. 1641 konstruierte Blaise Pascal eine primitive Addier- und Subtrahiermaschine. Um 1670 bemühte sich auch Leibniz um eine Rechenmaschine für die vier Grundrechenarten. Grillet und Polenius entwickelten ähnliche Modelle, aber erst 200 Jahre später sollten diese Maschinen ihre Reife erlangen.
Genannt sei hier der Arithmometer des Petersburger Ingenieurs W. T. Odhner. Ein System von Hebeln, Kurbeln und Sprossenrädern, die die Bezeichnung Odhnersche Räder erhalten haben, ermöglichte die Durchführung der rechnerischen Grundoperationen.

Wesentlich mehr als die Tischrechenmaschinen hat ein ganz anderer, heute sehr unscheinbarer Automat mit der Programmsteuerung zu tun. Seine Geburt in Europa - in China war er viel früher bekannt - mag er im 11. Jahrhundert erfahren haben, als ein Glöckner, der nach einem Stundenglas die Tageszeiten auszuläuten hatte, auf den Gedanken kam, diese Tätigkeit von einem Schlagwerk ausführen zu lassen, das von einem Gewicht betrieben wurde.
Das regelmäßige Einschalten des Schlagwerkes mag denselben Glöckner veranlasst haben, ein zweites Räderwerk zu konstruieren, das die Aufgabe hatte, nach einer bestimmten Anzahl von Umdrehungen das Glockenspiel in Gang zu setzen. Zeiger und

Zifferblatt vollendeten die Erfindung der Räderuhr.

Worin besteht die Programmsteuerung bei der Uhr? Hier wird ein sehr einfaches Programm mit nur einer einzigen Anweisung immer wieder abgearbeitet. Die Anweisung lautet: »Rücke um einen Zahn weiter!« Dieser Befehl wird nun in endloser Monotonie befolgt.

7.2 Der Siegeszug der Lochkarte

Im System Mensch-Tischrechenmaschine treten ernsthafte Schwierigkeiten auf, wenn das Zahlenmaterial umfangreicher wird. Nehmen wir einmal an, es seien 10000 Zahlen mit dem Faktor 3,14 zu multiplizieren.
Der Mensch wird gezwungen, vor Beginn jeder einzelnen Operation die beiden benötigten Zahlen einzudrücken. Erst dann kann er die bereits mechanisierte Rechenoperation befehlen.

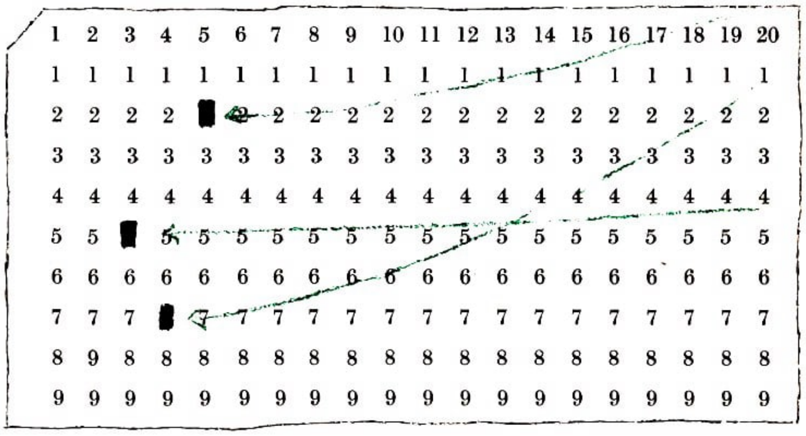
Schließlich muss er das Ergebnis auf einem Blatt Papier notieren. Der Gedanke ist daher naheliegend, auch die Ein- und Ausgabe von Zahlen zu mechanisieren. Möglich wurde das mit den Hollerith- oder Lochkartenmaschinen.

Die Lochkarte war bereits im 18. Jahrhundert bekannt. 1745 stellte der vielseitige Automatenbauer und Erfinder Vaucanson einen von Lochkarten gesteuerten Musterwebstuhl vor. Zu Anfang des 19. Jahrhunderts führte Jacquard die Lochkarte zur Steuerung von Webstühlen für feine Muster in der Lyoner Seidenindustrie ein.
Die Lochkarte versetzte den englischen Philosophen und Mathematiker Charles Babbage in so große Begeisterung, dass er sein Porträt auf einem Jacquard-Webstuhl herstellen ließ. Ausgehend von der Lochkarte skizzierte er den modernen Rechenautomaten:

Zahlen und Operationsbefehle sollten auf Lochkarten gestanzt und in ein Speicherwerk eingegeben werden. Zur Koordinierung des Ablaufs der Operationen hielt Babbage ein Steuerwerk für notwendig.
Die eigentlichen Rechenoperationen sollten in einem Rechenwerk ablaufen. Babbages Vorstellungen scheiterten damals an der noch wenig entwickelten Technik.

Im Jahre 1880 fand die 10. amerikanische Volkszählung statt. Herrmann Hollerith, der an der Auswertung mitarbeitete, wurde durch die gewaltigen Zahlenmengen zum Nachdenken angeregt.
Die Auswertung war in ihrer ersten Stufe durchaus primitiv, die Arbeiten bestanden im wesentlichen aus Sortierungen und Summenbildungen. Waren dazu Menschen erforderlich ?
Zwischen 1884 und 1889 reiften die Überlegungen Herrmann Holleriths zur großen Erfindung. In seiner Maschine. wurden Lochkarten durch metallische Fühlstifte abgetastet. 1890 wertete man die 11. amerikanische Volkszählung bereits mit Lochkartengeräten aus.

Im Jahre 1894 wurden bei einer Volkszählung im zaristischen Russland mehr als 100 Millionen Lochkarten verarbeitet. Zu Beginn des 20. Jahrhunderts waren die Hollerithanlagen über die ganze Erde verbreitet.

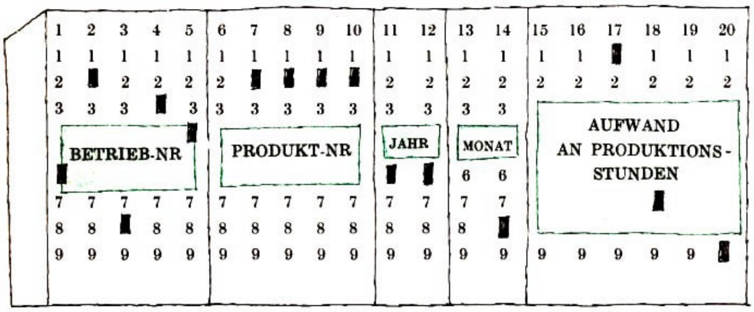


Die Lochkarte erwies sich als geradezu ideales Medium für die Eingabe von Zahlen in Rechenmaschinen. Wir wollen diese Karten ein wenig näher betrachten und wählen als Beispiel eine 20spaltige Karte mit 9 Zeilen. Um das Problem nicht unnötig zu komplizieren, verwenden wir den einfachsten Kode für die Darstellung von Ziffern:
Soll in irgendeiner Spalte eine bestimmte Ziffer gelocht werden, so muss die Stanzung in der gleichnamigen Zeile erfolgen. Die Null wird nicht gelocht.

Auf diese Weise ist in der Spalte 3 die Ziffer 5 gelocht, in Spalte 4 die Ziffer 7 und in Spalte 5 die Ziffer 2. Die Spalten 3 bis 5 nehmen daher die dreistellige Zahl 572 auf. Die dargestellte Lochkarte kann auf diese Weise eine insgesamt 20stellige Zahl aufnehmen. In der Praxis kennt man 80spaltige und 45spaltige Karten; unser Beispiel ist also eine grobe Vereinfachung.

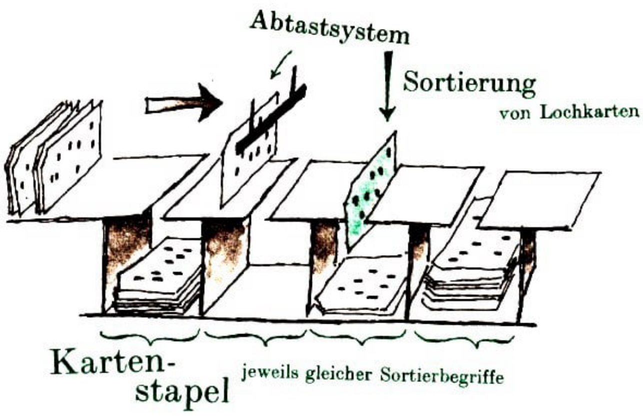
Die Lochkarte wird in einzelne Felder unterteilt, die jeweils Zahlen verschiedenen Charakters aufnehmen. In der Regel sind viele dieser Zahlen gewisse Ordnungsdaten oder Schlüsselnummern, die die Bedeutung der eigentlichen Zahlenangabe erläutern. Wir haben als Beispiel folgende Aufteilung der Karte gewählt.
In den Spalten 1 bis 5 soll die Schlüsselnummer des Betriebes stehen, der ein Produkt, dessen Kodezahl in den Spalten 6 bis 10 gelocht wird, erzeugt. In den Spalten 11 und 12 soll das Jahr vermerkt werden, in dem das Produkt hergestellt wurde, und in den Spalten 13 und 14 als genauere Zeitangabe der Monat. Die Spalten 15 bis 20 bleiben dann der eigentlichen Zahlenangabe vorbehalten, die in unserem Fall der Aufwand an Produktionsstunden für das jeweilige Produkt sein soll.
Auf der vorliegenden Lochkarte handelt es sich um den Betrieb mit der Kennzahl 62834, um das Produkt mit der Nummer 2222, das im Jahre 66 im Monat 8 hergestellt wurde und insgesamt 1709 Stunden Fertigungszeit beanspruchte.

Zu einer Lochkartenanlage gehört als wesentlicher Bestandteil ein Sortiergerät, mit dessen Hilfe die Karten nach bestimmten eingelochten Kennzeichen sortiert werden können. So ist es denkbar, dass im obigen Beispiel die Lochkarten nach Betriebsnummern geordnet werden; für jede Ziffer der Betriebsnummer ergibt sich dann in den Sammelkästen der Sortiermaschine ein gesondertes Kartenbündel.



Nach einer bestimmten Anzahl von Sortierdurchläufen, die der Länge des Sortierbegriffs entspricht, erhält man in unserem Beispiel ein Kartenpaket, das in der Reihenfolge aufsteigender Betriebsnummern geordnet ist. Die effektive Arbeitsgeschwindigkeit von Sortiermaschinen liegt bei 40000 Kartendurchläufen je Stunde.

Nach der Sortierung werden die Karten in die zentrale Verarbeitungseinrichtung einer Lochkartenanlage, in die Tabelliermaschine, eingegeben. Die Karten gelangen in den Kartenabtaster und werden spaltenweise abgefühlt. Bei jedem angetroffenen Loch schließt ein Fühlstift einen elektrischen Stromkreis, und die entsprechende Ziffer wird in ein inneres Register der Maschine eingelesen.



Es kommt der Wahrheit recht nahe, wenn man sich das Innere der Tabelliermaschine wie das einer elektrischen Tischrechenmaschine aufgebaut denkt.
Der Unterschied besteht darin, dass die Zahlen nicht eingetastet, sondern mit Hilfe von Lochkarten eingelesen werden und die Auslösung der Rechenoperationen mechanisiert ist.

Beim Durchlaufen der Karten werden in unserem Beispiel alle darauf gelochten Zahlen der Spalten 15 bis 20 in einem Register addiert. Jede Lochkarte ergibt auf der gleichzeitig gedruckten Tabelle eine Zeile. Sind alle Lochkarten durchgelaufen, wird die Summe aller addierten Werte gedruckt. Im vorliegenden Fall wäre das der Gesamtaufwand an Produktionsstunden für alle Betriebe und alle Erzeugnisse.

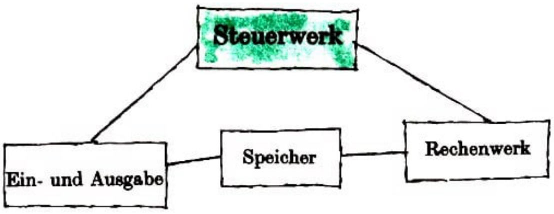
Lochkartenanlagen wurden bisher vor allem auf ökonomischem Gebiet bei Abrechnungsarbeiten angewendet. Die dort anfallenden Zahlenmengen sind sehr groß, die Rechnungen beschränken sich häufig auf die Durchführung elementarer Rechenoperationen, und die Zahlenangaben sind nach vielfältigen Merkmalen zu sortieren und auszuwerten.

Aber auch bei der Auswertung größerer Versuchsreihen, so z. B. in der Medizin, in der Metallurgie und in der Landwirtschaft, wurden Lochkartenanlagen mit Erfolg eingesetzt. Trotzdem ist die große Zeit dieser Maschinen bereits vergangen. Sie werden mehr und mehr durch moderne programmgesteuerte elektronische Datenverarbeitungsanlagen verdrängt.

7.3 Auf dem Weg zum Elektronenhirn

Wir haben erkannt, dass man zu Lochkartenmaschinen gelangt, wenn die Ein- und Ausgabe von Zahlen und die Auslösung der Rechenoperationen mechanisiert wird. Dabei ist zu berücksichtigen, dass Lochkartenanlagen bereits einfache Programme abarbeiten, die mit Hilfe von Stecktafeln gespeichert werden.
Die nächste zu erwartende Weiterentwicklung besteht nun darin, dass man die Möglichkeit für die Speicherung von Zahlen und Programmen im Inneren des Automaten in starkem Maße vergrößert.

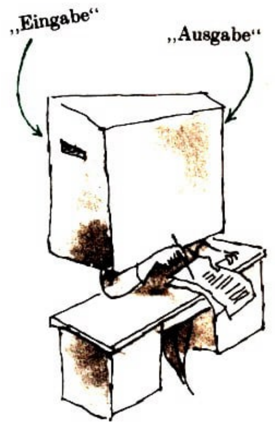
Programmgesteuerte elektronische Rechenautomaten müssen zunächst ein umfangreiches Speicherwerk besitzen, in das die Algorithmen der jeweiligen Probleme nebst dem erforderlichen Zahlenmaterial eingegeben werden. Weiter müssen sie, um den Namen Rechenmaschine mit Recht zu tragen, natürlich auch ein Rechenwerk vorweisen, in dem in Form elektronischer Schaltungen die Ausführungsbestimmungen der Grundrechenoperationen enthalten sind. Ein weiterer wichtiger Bestandteil jeder elektronischen Rechenmaschine sind die Ein- und Ausgabevorrichtungen für Zahlen und Befehle.



Als zentrales Organ ist schließlich diesen drei Teilen ein Steuerwerk oder Leitwerk übergeordnet, das die ablaufenden Operationen im gesamten Automaten auslöst und überwacht. Diesem Schema soll der Mensch, der mit Hilfe einer Tischrechenmaschine eine bestimmte Rechnung ausführt, gegenübergestellt werden. Dabei lässt sich der gleiche grundsätzliche Aufbau erkennen.

Die Tischrechenmaschine entspricht dem Rechenwerk des Automaten. Der Bogen Papier, auf dem die durchzuführenden Rechenoperationen verzeichnet sind, sowie benötigte Tabellenwerke mit Logarithmen, Potenzen und ähnlichem übernehmen die Rolle des Speicherwerkes.

Die Ein- und Ausgabe kann wieder durch ein Blatt Papier verwirklicht werden, auf das die angefallenen Ergebnisse sowie neue Aufträge geschrieben werden. Das überwachende Steuerwerk ist in diesem Bild der Mensch.



Dieser Vergleich zeigt, dass der Rechenautomat nicht nur die Tischrechenmaschine ersetzt, sondern das gesamte System umfasst, das aus dem rechnenden Menschen, der Tischrechenmaschine, dem Rechenblatt und den Tabellenwerken sowie aus dem Ein- und Ausgabeformular besteht.
Denken wir uns über den Schreibtisch und den Stuhl mit dem rechnenden Menschen einen Kasten gestülpt, der an der Seite Schlitze für die Eingabe von Zahlen und Befehlen und für die Ausgabe von Ergebnissen offenlässt, so haben wir das Modell eines »elektronischen programmgesteuerten Rechenautomaten « vor uns.
Heißt das nun, dass der wirkliche Rechenautomat ebenfalls ein Gehirn enthält?
Das wäre der Fall, wenn das Rechnen auf der Tischrechenmaschine das menschliche Gehirn in allen seinen Äußerungen und in seiner gesamten Leistungsfähigkeit in Anspruch nehmen würde. Das trifft aber nicht zu. Setzen wir ein genügend klar abgefasstes Rechenformular voraus, so genügen zur Bearbeitung desselben zwei Eigenschaften des rechnenden Menschen.

Das sind die Fähigkeiten, gewissenhaft und diszipliniert zu arbeiten. Die Disziplin kann man in einen völligen Gehorsam von elektronischen Bauteilen verwandeln und die Gewissenhaftigkeit in fehlerlose logische Schaltungen übertragen. So gelangt man zum programmgesteuerten Automaten.

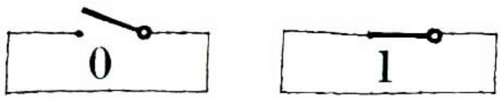
Wollen wir nun noch einmal über den Schreibtisch des rechnenden Menschen einen Kasten stülpen, um ein Modell des Rechenautomaten zu erhalten, so müssen wir dafür sorgen, dass der größte Teil des Kopfes des Menschen außerhalb des Kastens bleibt. Da es aber sehr schwierig sein dürfte, die genaue Proportion des außen verbleibenden Teils zum eingeschlossenen Teil des Kopfes anzugeben, wollen wir auf die weitere Untersuchung dieses Modells verzichten und uns statt dessen ein wenig ernsthafter mit dem Original selbst befassen.

7.4 Die Nervenzellen des Rechenautomaten

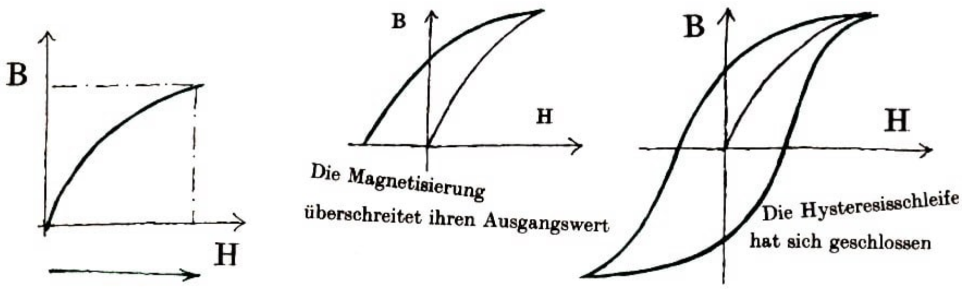
Wir benötigen als Grundbaustein für programmgesteuerte Automaten zum Aufbau logischer Schaltungen ein Element, das in der Lage ist, mehrere voneinander streng unterscheidbare Zustände anzunehmen, denen die Ziffern eines beliebigen Zahlensystems zugeordnet werden können.

Da das duale Zahlensystem mit der Grundzahl 2 besonders einfach ist und die gebräuchliche Logik nur die beiden Wahrheitswerte »wahr« und »falsch« besitzt, denen man ebenfalls die Ziffern 1 und 0 zuordnen kann, wird bei der überwiegenden Anzahl aller elektronischen Rechenautomaten das Dualsystem verwendet.
Der gesuchte Grundbaustein muss demzufolge zwei ausgewählte Zustände besitzen, die

mit den Namen »0« und »1« belegt wurden. Das einfachste Element dieser Art ist der elektrische Schalter, der einen Stromkreis bei »0« schließen und bei »1« öffnen kann.



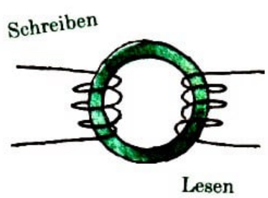
Allerdings ist ein solcher einfacher Schalter zu wenig mechanisiert, um als Bauelement in Schaltungen von Rechenautomaten Verwendung zu finden. Seine vervollkommneten Formen sind das elektronische oder das elektromagnetische Relais und schließlich die Elektronenröhre und der Transistor.
Neben dem Transistor als typisches Schaltelement besitzt ein moderner Rechenautomat sogenannte Ferritkerne als Grundbausteine zur Speicherung von Ja-Nein-Informationen. Die Informationsspeicherung ist die wichtigste materielle Voraussetzung der Programmsteuerung. Wir wollen daher auf den Ferritkern etwas näher eingehen.



Ferritkerne sind winzige Ringe aus einem Material, das die Eigenschaft hat, magnetische Zustände zu speichern. Setzt man ein solches Material in völlig unmagnetisiertem Zustand einer magnetischen Erregung H aus, so wächst seine Magnetisierung B bis zu einem gewissen Höchstwert.
Lässt man jetzt die magnetische Erregung H abklingen, so folgt die Magnetisierung mit großer Verzögerung. Selbst beim Fehlen einer magnetischen Erregung, also bei $H = 0$, bleibt das Material weiter im magnetisierten Zustand.

Kehrt man nun das magnetische Feld H um, so erreicht die Magnetisierung ihren Wert 0 erst bei einer ganz bestimmten Größe der entgegengesetzten Erregung. Von nun an folgt die Magnetisierung der Erregung auf der sogenannten Hysteresisschleife.

Es ist wichtig, dass die Ferritkerne beim Fehlen eines magnetischen Feldes in zwei magnetischen Zuständen vorgefunden werden können, die von der Vorgeschichte der Magnetisierung abhängen und denen man wieder die Ziffern 0 und 1 zuordnen kann.



Die Ferritkerne werden nun mit einer Schreibwicklung und einer Lesewicklung versehen. Während die Schreibwicklung die Überführung in einen bestimmten Zustand gestattet, liefert die Leseleitung eine Information über den vorgefundenen Zustand. Einen solchen Ferritkern kann man getrost als Nervenzelle des Rechenautomaten bezeichnen.

Mit Hilfe von Ferritkernen und Röhren oder Transistoren lassen sich nun all jene logischen Schaltungen verwirklichen, die wir bereits in den Ausführungen über die Schaltalgebra angedeutet haben. Aus den Grundschaltungen setzt man dann die Strukturen für die Rechenoperationen und alle anderen logischen Leistungen des Automaten zusammen.
Im Rechenwerk des Automaten werden vor allem Schaltungen für die Grundrechenarten benötigt. Wir wollen hier die einfache Schaltung für die Addition zweier einstelliger Zahlen im Dualsystem entwerfen. Wir nennen die beiden einstelligen Dualzahlen a und b und erinnern uns an die Additionstabelle:

a	b	$a+b$
0	1	1
1	0	1
0	0	0
1	1	10 Übertrag

a	b	$a+b$ ohne Übertrag
0	1	1
1	0	1
0	0	0
1	1	0

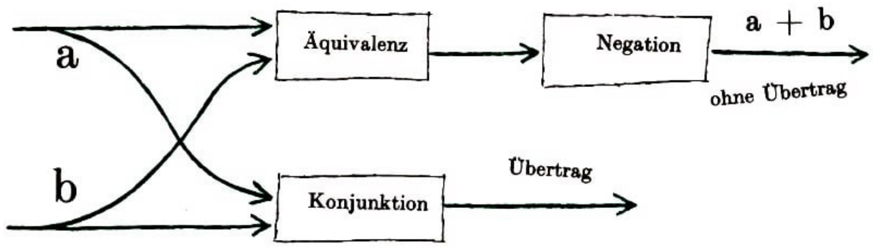
Bei der Addition $1+1$ tritt demnach ein Übertrag in die nächste Stelle auf. Wir wollen diesen Übertrag gesondert behandeln.
Er findet sich offensichtlich nur in dem Fall, wo sowohl a als auch b den Wert 1 haben. Das ist aber das Charakteristische der Konjunktion $a \wedge b$, was wir leicht aus der dazugehörigen Tabelle erkennen können, wenn wir für den Wahrheitswert W die Zahl 1 und für den Wahrheitswert F den Wert 0 einsetzen.
Dem Übertrag entspricht also die Schaltung der Konjunktion.

Besehen wir uns jetzt die Addition ohne Übertrag und suchen eine logische Schaltung, die diese Addition realisieren kann. Bei genauer Betrachtung der logischen Äquivalenz zeigt es sich, dass ihre Verneinung das geforderte Verhalten aufweist. Noch deutlicher tritt die Entsprechung zwischen der logischen Äquivalenz und der Addition ohne Übertrag für zwei Dualziffern hervor, wenn wir in der Tabelle an Stelle des Wahrheitswertes W die Zahl 1 und an Stelle des Wertes F die Zahl 0 setzen.
Die Addition ohne Übertrag kann daher durch die Verneinung der Äquivalenz realisiert werden.

a	b	$\sim (a \leftrightarrow b)$
F	W	W
W	F	W
F	F	F
W	W	F

Setzen wir das Vorhandensein einer solchen Äquivalenzschaltung sowie der Konjunktionsschaltung voraus, so lässt sich die folgende Gesamtschaltung für die Addition zweier Dualziffern angeben.

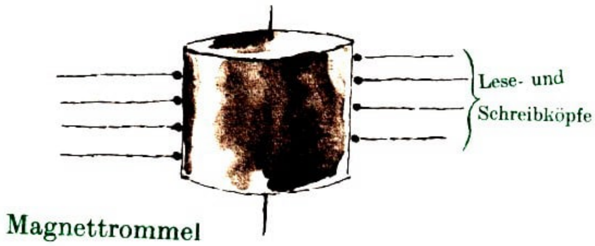
Wollen wir die vollständige Addition mehrstelliger Dualzahlen verwirklichen, so müssen mehrere solcher Schaltungen verwendet und des weiteren auch die Überträge addiert werden. In dieser Art können wir uns auch die Ausführung der anderen Grundrechenoperationen im Automaten vorstellen.



Moderne Großrechenanlagen arbeiten mit unvorstellbaren Geschwindigkeiten.
Es existieren Maschinen, die eine Addition zweier 10stelliger Dezimalzahlen in weniger als 10 Millionstel Sekunden bewerkstelligen. Solche Maschinen sind in der Lage, mehr als 100000 Additionen in einer Sekunde durchzuführen. In dieser Zeit hätte der Mensch gerade eine Addition erledigt.

7.5 Revue der künstlichen Gedächtnisse

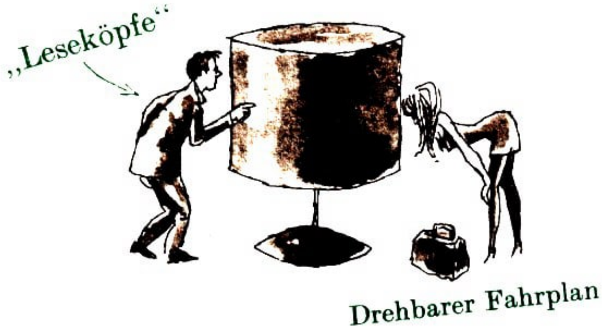
Nach dem Rechenwerk wollen wir uns dem Speicherwerk einer elektronischen Rechenmaschine zuwenden. Wir kennen die Speicherung von Informationen auf magnetischen Medien durch das Tonbandgerät. Denken wir uns ein übermäßig breites Tonband auf einen Zylinder gewickelt, so haben wir praktisch die Magnettrommel als Speicherwerk des Rechenautomaten vor uns.



Der Trommel sitzen winzige Lese- und Speicherköpfe gegenüber, die Zahlen und Befehle von der Trommel ablesen und auf die Trommel transportieren.
Ein mit regelungstechnischen Geräten ausgestatteter Elektromotor sorgt für eine konstante Umdrehungsgeschwindigkeit der Magnettrommel. Dadurch wird sie gleichsam zur inneren Uhr des Rechenautomaten, nach der sich das Steuerwerk beim Aufruf neuer Befehle und Zahlen richten muss.

Am besten vergleichen wir die Magnettrommel mit drehbaren Fahrplänen, wie wir sie auf manchen Bahnhöfen finden können. Die Leseköpfe werden in diesem Bild von den Köpfen der Reisenden dargestellt.

Weiter kann man die Analogie allerdings nicht treiben, da die Magnettrommel keineswegs angehalten werden muss, wenn eine Angabe benötigt wird. Es sind Geschwindigkeiten von 3000 bis 12000 Umdrehungen in der Minute üblich.
Beim Lesevorgang wird die Zahl, während sie am Lesekopf vorüberfliegt, Ziffer für Ziffer von ihm aufgenommen und in einem Register wieder zusammengesetzt.



Jeder Speicher ist in einzelne Zellen aufgeteilt, denen man jeweils eine feste Zahl, die Adresse der Zelle, zuordnet. Wickelt man die Magnetschicht der Speichertrommel auf eine ebene Unterlage ab, so erhält man ein Viereck, das in eine Vielzahl von Teilvierecken zerlegt ist.

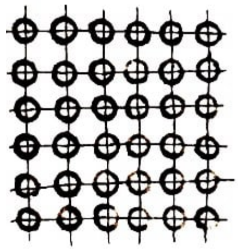
0	6		...			994
1			...			995
2			...			996
3			...			997
4			...			998
5			...			999

Oberfläche der Magnettrommel mit den Zellen 0 bis 999

Jedes dieser Teilvierecke stellt eine einzeln ansprechbare Speicherzelle dar, in der in den meisten Fällen eine Zahl oder ein Befehl Platz findet.

Die Adressierung der Zellen erfolgt sinnvollerweise so, dass durch eine Mindestanzahl von Ja-Nein-Entscheidungen aus der Angabe der Adresse die konkrete Lage der Zelle gefunden werden kann.
Aus diesem Grunde besitzt die Mehrzahl der Rechenautomaten Speicherwerke mit einer merkwürdig »unrunden« Anzahl von Zellen wie 1024, 4096 oder 65536. Das sind aber gerade die Zweierpotenzen 2^{10}, 2^{12} und 2^{16}.
Aus dem Rateexperiment im Kapitel über die Grundlagen der Informationstheorie wissen wir, dass in solchen Fällen zur eindeutigen Ermittlung einer bestimmten Zelle so viele Ja-Nein-Entscheidungen benötigt werden, wie der Exponent der aufgeschriebenen Potenzen angibt.

Es hat sich eingebürgert, die Zahl 2^{10} als Maß für die Kapazität eines Speichers zu benutzen. Man spricht in Anlehnung an das Kilogramm von Kilo-Worten, Kilo-Zellen oder Kilo-Zeichen.
Die Angabe 4 Kilo-Zellen ist gleichbedeutend mit $4 \cdot 2^{10} = 4 \cdot 1024 = 4096$ Zellen.
Das innere Speicherwerk eines elektronischen Rechenautomaten ist demnach ein großes Hotel für Zahlen und Befehle, die in eindeutig nummerierten Einzelzimmern wohnen. Die Lage der Zelle auf der Trommel kann man sich durch den Lesekopf, der die dazugehörige Reihe abtastet, und durch die »Uhrzeit« oder den Zeittakt festgelegt denken, zu dem die Zelle am Lesekopf eintrifft.

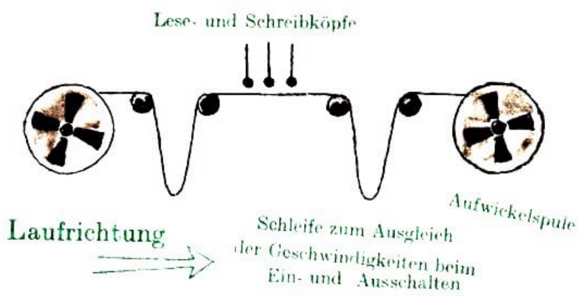


Neben den Magnettrommeln werden hauptsächlich die Kernspeicher als Hauptspeicher von Rechenautomaten verwendet. Kernspeicher sind gitterförmige Strukturen von Drähten, in deren Kreuzungspunkten Ferritkerne angebracht sind, die jeweils einen von zwei möglichen ausgeprägten magnetischen Zuständen annehmen, wodurch jeder Ferritkern zum Speicher einer Information von 1 bit wird.
Die Adressierung erfolgt hier über die waagerecht und senkrecht verlaufenden Stromleiter. Der Vorteil der Kernspeicher gegenüber den Trommelspeichern besteht darin, dass sie ohne mechanisch bewegte Teile auskommen und damit wesentlich kürzere Zeiten für das Lesen und Speichern benötigen.

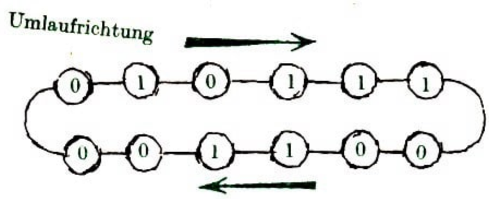
Moderne Rechenmaschinen besitzen daher Ferritkernspeicher als Hauptspeicher oder, wie man auch sagt, als Arbeits- oder Operativspeicher, während die Trommeln als zusätzliche, periphere Speicher verwendet werden.
Die in den ersten Rechenmaschinen benutzten Speicher bestanden aus einer Anordnung von Relais, also von Schaltern, die sich in den Stellungen »Ein« und »Aus« befinden konnten. Solche Speicherwerke verfügten über eine nur geringe Kapazität und verursachten überdies noch hohe Kosten.

Fast alle modernen Rechenautomaten besitzen Magnetbänder als Zusatzspeicher. Die magnetisierbare Schicht des Bandes wird ähnlich wie beim gewöhnlichen Tonband durch elektromagnetische Impulse beschrieben. Dabei unterscheidet man wieder zwei magnetische Zustände, denen man die Ziffern 0 und 1 zuordnet.

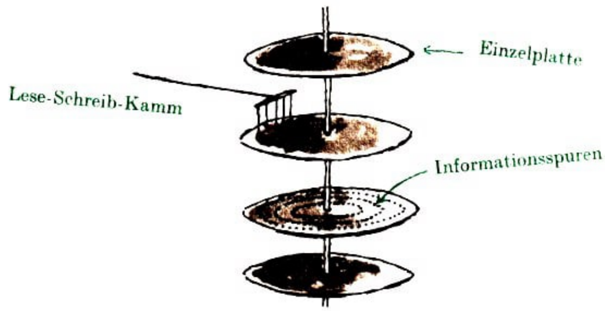
Das Spuren Band wird in einzelnen Spuren.oder Kanälen beschrieben, die in Richtung der Längsausdehnung des Magnetbandes verlaufen und denen die Schreib- und Leseköpfe gegenüberliegen. Die gleichmäßig voneinander entfernten Zeilen des Bandes, die senkrecht zu den Spuren verlaufen, nehmen dann je nach der Anzahl der vorhandenen Kanäle beispielsweise 8 oder 14 bit auf.



Die folgende Abbildung erklärt das Prinzip einer Magnetbandeinheit. Beim Lesen oder Schreiben läuft das Band von der eingelegten vollen Spule auf eine anfangs leere Aufwickelspule. Die Informationen werden von den Lese-Schreibköpfen gelesen oder auf das Band geschrieben.
Verbindet man eine größere Anzahl von Ferritkernen über ihre Lese- und Schreibleitungen zu einem Ring, so entsteht das sogenannte Register mit umlaufendem Inhalt, wobei man mit Hilfe von Transistoren oder Elektronenröhren als Verstärker veranlasst, dass die in diesem Register aufbewahrte Information ständig kreist.



Die Information wird in einem Zustand der Bereitschaft gehalten; sie steht bei Bedarf schnell zur Verfügung. Solche Register finden als Speicher häufig gebrauchter Informationen Verwendung. Ein Sonderfall dieses Registers ist das sogenannte Flip-Flop, das nur ein einziges bit aufbewahrt.
Mit der zunehmenden Entwicklung der Rechentechnik wuchs der Bedarf an Speichern, die große Datenmengen aufnehmen und obendrein einen schnellen Zugriff zu diesen Daten gestatten sollten. Die bekannteste technische Realisierung dieser Großraumspeicher sind die Magnetplattenspeicher. Bereits heute existieren solche »Datenbanken«, die mehrere 10 Millionen Zahlen speichern können.



Die Abbildung veranschaulicht den Aufbau des Plattenspeichers. Jede Einzelplatte des Stapels ist mit einer gewöhnlichen Schallplatte vergleichbar, allerdings mit dem Unterschied, dass die Speicherung auf den Oberflächen magnetisch erfolgt.
Ein Lese-Schreib-Kamm besorgt das Lesen und Beschreiben der Informationsspuren.

Gegenwärtig werden neben einigen anderen Speicherarten auch sogenannte Tiefsttemperaturspeicher im Hinblick auf ihre Einsatzmöglichkeit in der modernen Rechentechnik erforscht und geprüft.
Bei diesem Typ des Speichers wird ein interessanter Effekt der Elektrodynamik ausgenutzt. Bei tiefen Temperaturen, in der Nähe des absoluten Nullpunktes von -273°C, verliert ein gewöhnliches Stück Draht seinen elektrischen Widerstand. Seine Leitfähigkeit wird praktisch unbegrenzt.

Ein einmal angeregter Stromfluss bleibt daher in einem geeignet ausgebildeten Leiter bei solch tiefen Temperaturen lange Zeit erhalten. Auf diese Weise will man in den Tiefsttemperaturspeichern Informationen aufbewahren. Vorerst sind jedoch die Herstellungs- und Wartungskosten noch außerordentlich hoch.

Allgemein ergehen an neuzeitliche Speicher folgende Forderungen:
Sie müssen zunächst eine hohe zeitliche Beständigkeit besitzen. Es wird gewünscht, dass eine einmal gespeicherte Information recht lange unverfälscht im Speicher verbleibt. Für eine Magnettrommel liegt diese Zeit in der Größenordnung von Wochen, für Magnetbänder ist sie um ein Vielfaches größer.
Neben angestrebten großen Fassungsvermögen verlangt man weiterhin kleine Zugriffszeiten.
Eine gespeicherte Information soll schnell erreichbar sein. Schließlich erhebt man auch Forderungen nach niedrigen Herstellungs- und Unterhaltungskosten sowie nach großer Betriebssicherheit und wenig Raumbedarf.

In der Tabelle sind die charakteristischen Daten, wie Zugriffszeiten, Kapazität und Speicherdichte der bekanntesten Speichertypen, einander gegenübergestellt.

Speicherart	Zugriffszeit	Maximale Kapazität bit	Speicherdichte bit/cm^3
Relais	5 ms	10^2	0,01
Flip-Flop	0,5 μs	10^3	0,01
Magnettrommel	10 ms	10^6	50
Kernspeicher	2 μs	10^5	130
Tiefsttemperaturspeicher	5 ns	10^7	10^3
Magnetband	bis 60 s	10^8	10^5
Magnetplattenspeicher	5 μs	10^9	10^6

Bei der Betrachtung der Zugriffszeit sollte man sich vergegenwärtigen, dass eine Millisekunde (1 ms) eine tausendstel Sekunde ist, eine Mikrosekunde (1 μs) den millionsten Teil einer Sekunde bildet und eine Nanosekunde (1 ns) ein Milliardstel einer Sekunde bedeutet.

Die maximale Kapazität ist in Zehnerpotenzen dargestellt. Bei der Trommel finden wir beispielsweise eine Kapazität von $10^6 = 1$ Million bit.
Die Speicherdichte verbindet die maximale Kapazität und den Raumbedarf zu einer aussagekräftigen Kennzahl. Bei der Magnettrommel liegt diese bei 50 bit/cm^3, was besagt, dass 50 bit in einem Kubikzentimeter untergebracht werden können.

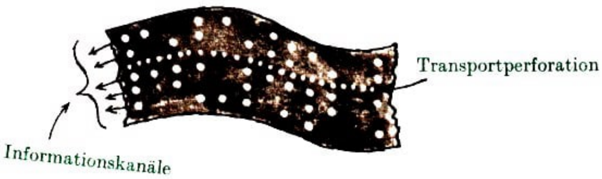
7.6 Die Sinnesorgane des Rechners

Bevor das Rechnen auf einem elektronischen programmgesteuerten Automaten beginnen kann, muss dafür gesorgt werden, dass sich alle dazu notwendigen Informationen im Hauptspeicher befinden. Die Zahlen und die Programme, nach dem die Zahlen verarbeitet werden sollen, sind auf irgendeinem Wege dem Automaten mitzuteilen; sie müssen ihm eingegeben werden.

Eingabegeschwindigkeiten	
Lochkarten	700 Karten/min
Lochstreifen	1000 Zeichen/s

Das bekannteste Eingabemedium ist die Lochkarte. Zahlen und Befehle werden nach einer speziellen Kodierungsvorschrift in die Karte gelocht. Im Eingabegerät werden die Lochkarten von Bürsten abgetastet, die jedes gestanzte Loch in einen Spannungsimpuls umwandeln. Gleichzeitig tritt die logische Schalttechnik des Automaten in Aktion.
Die in einem Register auflaufenden Zahlen müssen in die rein duale Form übergeführt werden, in der sie der Automat verarbeiten kann. Man nennt diesen Vorgang die Konvertierung.
Nach der Konvertierung der Zahlen in die duale Darstellung von Spannungsimpulsen werden sie dann auf den Hauptspeicher des Automaten geleitet.

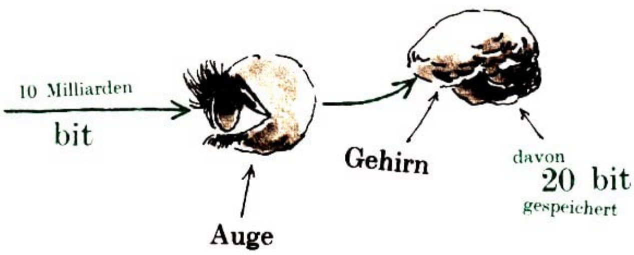
Neben der Lochkarte kennen wir den Lochstreifen, in den wie in die Karte Löcher gestanzt werden. Je nachdem, wieviel Lochungen in einer Lochstreifenzeile möglich sind, unterscheidet man 5-, 6-, 7- und 8-Kanalstreifen.



Das Magnetband, auf dem die Informationen bereits in elektronischer Schrift festgehalten sind, kann in jüngster Zeit auch als Eingabemedium angesprochen werden. Ursprünglich war es kein eigentlich primäres Eingabemittel. Wollte man Magnetbänder beschreiben, so setzte das stets den Eingabevorgang über die Lochkarte oder den Lochstreifen voraus. Neuerdings sind nun Geräte entwickelt worden, die das Übertragen von Informationen von einer Tastatur direkt auf das Magnetband erlauben.

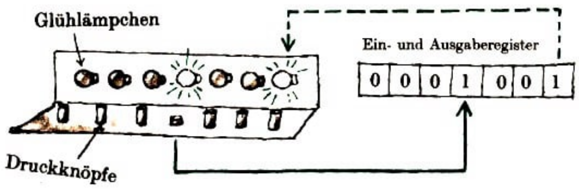
In der Zukunft wird auch das direkte Lesen der Belege große Bedeutung erlangen. Bereits heute existieren einige Belegleser, die nicht nur spezielle Magnetschriften, sondern auch handgeschriebene Ziffern und Buchstaben entschlüsseln können.
Das in Vilnjus entwickelte sowjetische Lesegerät »Ruta 701« entziffert z. B. 200 gedruckte oder handgeschriebene Zeichen in der Sekunde.

Interessant ist ein Vergleich der Eingabevorrichtung eines Automaten mit den menschlichen Sinnesorganen.



Der Automat muss sich im Vergleich zum Gehirn mit einer äußerst geringen Menge von Informationen begnügen. Allein das menschliche Auge ist in der Lage, 10 Milliarden bit in der Sekunde aufzunehmen, während die Aufnahmegeschwindigkeit der Lochkarteneingabe eines schnellen Rechenautomaten ungefähr bei tausend bit in der Sekunde liegt.
Diese auffallende Überlegenheit der menschlichen Sinnesorgane verschwindet bei näherer Betrachtung vollständig: Trotz der riesigen Aufnahmekapazität seiner Sinnesorgane ist der Mensch nicht in der Lage, mehr als 20 bit in der Sekunde dauerhaft zu speichern, d. h. im Gedächtnis zu behalten.

Die Rechenmaschine kann dagegen die Informationen mit derselben Geschwindigkeit, mit der sie sie über ihre Eingabekanäle erhält, auch dauerhaft abspeichern.
Eine wichtige Form der Eingabe von Informationen im Rechenautomaten darf hier nicht unerwähnt bleiben. Die meisten Automaten besitzen ein Ein- und Ausgaberegister, dessen Inhalt durch Glühlämpchen auf dem Steuerpult nach außen sichtbar gemacht und von außen durch Drücken von Knöpfen beliebig gefüllt werden kann.

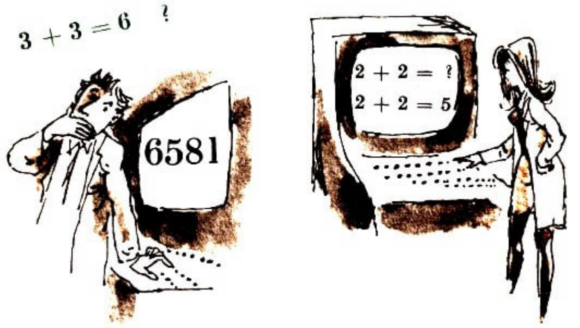


Jeder niedergedrückte Knopf, jedes leuchtende Glühlämpchen entspricht dabei in diesem Register einer dualen 1. Den Inhalt des Ein- und Ausgaberegisters kann man durch Schalter und Drehknöpfe in jedes beliebige innere Register des Automaten und in jede beliebige Speicherzelle seines Hauptspeichers bringen. Das ist eine bequeme Eingriffsmöglichkeit in den Automaten, um Befehle oder Zahlen zu ersetzen oder abzuändern.

Mit Hilfe dieses Ein- und Ausgaberegisters können solche Rechnungen wie $2 \cdot 2 = 4$ oder $1+1 = 2$ auf dem programmgesteuerten Automaten ähnlich wie auf einer Tischrechenmaschine durchgeführt werden. Man bringt den Automaten mit Hilfe eines Schalters in einen Zustand, in dem er nach jedem Befehlsschritt anhält, gibt die beiden Operanden und die dazugehörigen Rechenoperation ein und befiehlt die Ausführung dieser Operation.

Die Durchführung einer solchen Rechnung hat vor Jahren einmal ein Journalist etwa folgendermaßen erlebt: Er bat, ihm doch einmal vorzuführen, wie der Automat die Summe $3 + 3 = 6$ berechnet.
Zu seinem Erstaunen fasste sich der Mathematiker in den Bart und murmelte mit einiger Verlegenheit: »Das ist gar nicht so einfach«. Dann drückte er eine Anzahl von Knöpfen, ließ Glühbirnen aufleuchten, schaltete an verschiedenen Hebeln und sagte nach einiger Zeit: »Bitte«.

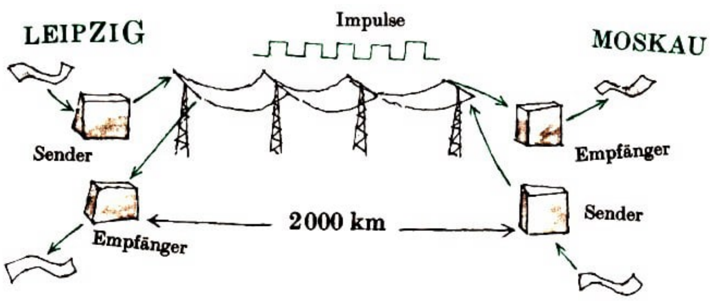
Der Automat hatte das Ergebnis gedruckt. Der Journalist hielt einen Zettel in der Hand und las 6581. Der Mathematiker bemerkte die Verwunderung seines Gastes und war nach einem verlegenen Blick auf das Papier bereit, ihn vom richtigen Ergebnis vermittels einer neuen Rechnung in Kenntnis zu setzen. Der Besucher aber dankte höflich.



Das »Gespräch« mit dem Rechenautomaten ist inzwischen durch den Einsatz von sogenannten Bildschirmanzeigegeräten wesentlich einfacher geworden. Solche Geräte haben eine Tastatur, die alle dem entsprechenden Automaten »verständlichen« Zeichen aufweist.
Jedes angeschlagene Zeichen erscheint auf dem Bildschirm, der eine gewisse Anzahl von Zeilen aufnimmt und damit die Funktion des Papierbogens bei der normalen Schreibmaschine erfüllt.

Ist die Nachricht an die Rechenmaschine fertiggestellt und auf dem Bildschirm nach Fehlern durchmustert, so wird sie mit einem Knopfdruck in den Hauptspeicher des Automaten gesandt. Umgekehrt können alle Antworten der Rechenmaschine auf diesem Bildschirm sichtbar gemacht werden.
Der Begriff des Eingabegerätes wird hier durch den Begriff des Kommunikationsgerätes ersetzt. Die Informationen werden dem Automaten direkt, d. h. ohne Bindung an ein Zwischenmedium im sogenannten on-line-Betrieb, eingegeben, während bei der Eingabe mit Lochkarten oder Lochstreifen die Daten an diese materiellen Überträger gebunden waren und - wie man sagt - im off-line- Betrieb in den Automaten gelangten.

Von größter Bedeutung für die Gestaltung der künftigen Arbeit von Konstrukteuren und Technologen sind solche Bildschirmgeräte, die eine direkte Eingabe der Koordinaten von Kurven und Linienzügen in den Automaten gestatten.
Der Konstrukteur zeichnet mit einem »Lichtstift« die Grobansicht des Werkstückes auf den Bildschirm, gibt die zu verwendenden Materialarten an und erhält wenig später genaue Konstruktionsangaben und Berechnungsergebnisse zur Festigkeit und Funktionsweise des Bauteils.



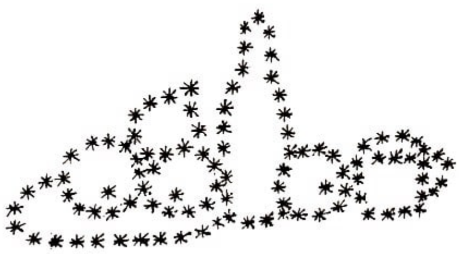
Eine andere Gruppe von Kommunikationsgeräten ermöglicht uns, das »Gespräch« mit dem Automaten über große Entfernungen zu führen. Mit dieser Gruppe sind die Datenfernübertragungsgeräte gemeint.

Anlässlich der Frühjahrsmesse 1969 wurde eine 2000 km lange »Datenbrücke« Leipzig-Moskau der Öffentlichkeit vorgestellt, die auf der in der DDR entwickelten Datenfernübertragungseinrichtung DFE 550 basierte. Bei dieser Vorführung war die Sendeeinrichtung an einen Lochstreifenleser und die Empfangseinrichtung an einen Lochstreifenstanzer gekoppelt. Die Übertragung vollzieht sich in Form von Ja-Nein-Impulsen über das öffentliche Fernsprechnetz mit einer Geschwindigkeit von 1200 bit/s.

Die Bedeutung der Möglichkeit, Daten über Telefonleitungen zu übertragen, lässt sich heute wahrscheinlich kaum in ihrem vollen Ausmaß erkennen. Vielleicht werden wir eines Tages über das Telefon Zugang zu den verschiedensten zentral angelegten Informationsspeichern in großen Automaten besitzen und diese beauftragen können, Berechnungen und Untersuchungen für uns durchzuführen. Die Ergebnisse würde uns überdies eine freundliche Mädchenstimme, keineswegs eine knarrende, kaum verstehbare Roboterkehle übermitteln.

Bis dahin müssen wir uns mit dem heute bekanntesten Ausgabevorgang, dem Drucken, begnügen. Eine Papierrolle gleitet hinter einem Farbband und einem Typensatz von Zahlen, Buchstaben und sonstigen Zeichen vorüber. Die in das Druckwerk gelangenden Impulse werden verstärkt und regen die entsprechende Type zum Druck an.
Zuvor müssen die Zahlen aus der dualen Darstellung im Automaten in die dezimale Schreibweise umgewandelt werden. Hier wird also der Weg der Eingabe in umgekehrter Richtung durchlaufen, weshalb man den Vorgang auch Rückkonvertierung nennt.

In den meisten Fällen ist die Ausgabevorrichtung der Automaten als Tabellendruck ausgebildet, mit dem man neben Tabellen und erläuternden Bemerkungen auch lange Berichte sowie grobe graphische Darstellungen und einfache Zeichnungen ausdrucken kann. Ein Stillleben frei nach Cezanne hätte dabei ungefähr das im Bild gezeigte Aussehen.



Wer die Rechenautomaten wegen dieser Fähigkeit als Künstler bezeichnet, hat natürlich unrecht. Das Original wurde vorher mit einem Raster überzogen, den Eckpunkten ordnete man die Kodezahl des Zeichens - zu, und das so entstandene Tableau von Kodezahlen wurde schließlich über Lochkarten auf den Hauptspeicher gebracht und von dort aus in kurzer Zeit ausgedruckt.

Ausgabegeschwindigkeiten	
Drucken	500 Zeilen/min
Lochkarten	300 Karten/min
Lochstreifen	1000 Zeichen/s

Nach dem Tabellendruck gibt es noch weitere Möglichkeiten der Ausgabe. So können die Zahlen auf Lochkarten oder Lochstreifen gestanzt werden. Sie verwandeln sich

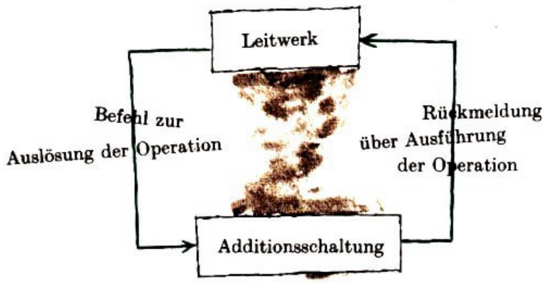
damit in Eingabewerte für weitere Rechnungen. Ergebnisse können auch auf Magnetbänder geschrieben oder mit direkt angeschlossenen Datenübertragungsgeräten über große Entfernungen transportiert werden.
Um wertvolle Zeit auf der Rechenmaschine zu sparen, überführt man die Ergebnisse mitunter auf einen außerhalb der Maschine gelegenen Speicher, der mit einer Druckvorrichtung gekoppelt ist. Sind alle Daten in das Gerät eingelaufen, so kann der Druckvorgang ausgelöst werden. Der eigentliche Rechenautomat arbeitet indessen bereits an anderen Aufgaben.

7.7 Das Gehirn des Elektronenhirns

Mit der Erklärung der Ein- und Ausgabevorrichtungen haben wir das letzte untergeordnete Organ der elektronischen Rechenmaschine kennengelernt. Nun soll die Beschreibung des übergeordneten Steuerwerks folgen, das gewissermaßen als »Gehirn des Elektronenhirns « angesehen werden kann.

Das Nervensystem und insbesondere das Gehirn des tierischen Organismus kontrolliert und koordiniert das Zusammenspiel der untergeordneten Teile des Körpers. Eine ähnliche Aufgabe hat das Leit- und Steuerwerk in Bezug auf die übrigen Teile des programmgesteuerten Automaten zu erfüllen.
Es besitzt wechselseitige Informationsverbindungen zu allen ihm unterstellten Geräten und vergibt »Aufträge« an sie.

Ordnet das Leitwerk beispielsweise die Durchführung einer Addition im Rechenwerk an, so muss es auf die Beendigung dieser Operation warten, ehe es neue Aufgaben an die Recheneinheit vergibt.



Auf der anderen Seite melden sich einige Geräte »unaufgefordert« bei der zentralen Steuereinheit. Der Lochstreifenleser kann seine Einschaltung durch den Bediener mitteilen: Nun muss für die bevorstehende Eingabe von Informationen Platz im Kernspeicher geschaffen werden.

Jedem Rechenautomaten stehen eine innere Uhr oder ein Taktgeber zur Verfügung. Viele Operationen, die vom Leitwerk ausgelöst werden, haben eine ganz bestimmte feststehende Zeitdauer. Das gestattet eine exakte Planung und Kontrolle dieser Vorgänge. Während des Ablaufs einer solchen Operation, deren Zeitdauer bekannt ist, kann das Leitwerk die nötigen Signale zur Auslösung einer anderen Operation bereitstellen oder diese zweite Operation bereits einleiten und zum gemeinsamen Zeitpunkt der Beendigung beider Operationen beispielsweise eine dritte Handlung anordnen.

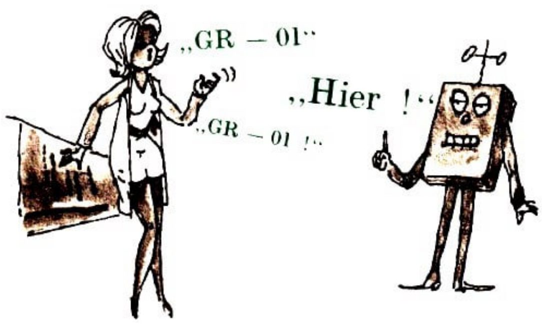
Alle gespeicherten Befehle werden nach einem unveränderlichen, dem Automaten zugrunde liegenden Schlüssel vom Leitwerk entziffert und danach im einzelnen abgearbeitet. Die Rechenmaschine wird dadurch, dass sie tausend Mal die Rechenoperation $3 + 2 = 5$ ausführt, nicht klüger. Treten im tausendundersten Fall dieselben Zahlen 3 und 2 auf, die durch dieselbe Rechenoperation $+$ verknüpft werden sollen, so befiehlt das Leitwerk die Auslösung der Operation, und das Rechenwerk führt nach wie vor alle dazu nötigen Einzelschritte genau wie in den vorangegangenen Fällen aus.

Die Starrheit einer solchen Konstruktion und die damit verbundene Unfähigkeit zum Lernen berauben uns jeder Hoffnung, in programmgesteuerten Automaten gehirnähnliche Mechanismen zu finden. Wenn wir später von Programmen hören werden, die bei ihrer Abarbeitung auf elektronischen Rechenmaschinen Lernfähigkeit zeigen, so sind diese Lernprogramme nicht mit dem Automaten selbst zu verwechseln, dessen innere Einrichtung nur dazu in der Lage ist, den Vorgang der Befehlsabarbeitung in ständigem Gleichmaß nach starren Gesetzen zu wiederholen.

Die Anweisungen eines Algorithmus und die während seiner Abarbeitung benötigten Zahlen werden in verschlüsselter Form über ein geeignetes Eingabemedium auf den Hauptspeicher des Automaten gebracht. Nun beginnt die Abarbeitung der Befehle, der Automat tritt in den Zustand der Programmsteuerung ein.
Jeder Befehl wird bis auf später zu besprechende Ausnahmen in derselben Reihenfolge aufgerufen, in der er gespeichert wurde, das Leitwerk veranlasst die Durchführung aller logischen Operationen, auf denen der Befehl aufgebaut ist.
Nach der Abarbeitung des Befehls geht eine Rückmeldung an das Leitwerk, das jetzt den nächsten Befehl aufruft. Dieser Vorgang wiederholt sich bis zu dem Zeitpunkt, an dem der Befehl »Halt« entschlüsselt wird. Mit der Abarbeitung dieses Befehls beendet die Maschine ihre Tätigkeit an der ihr vorgegebenen Aufgabe und kann wenig später mit der Lösung eines weiteren Problems beauftragt werden.

7.8 Das Wort einer Maschine

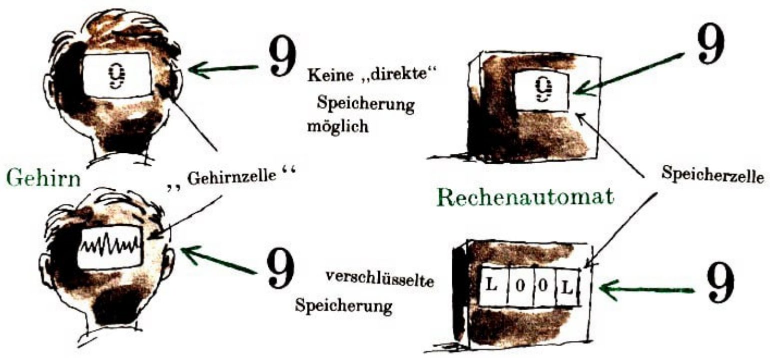
Obwohl wir den Begriff »Befehl« schon oft verwendet haben, fehlt uns noch dessen genaue Erklärung. Hier beginnt das Problem, die Sprache der Rechenmaschine zu lernen.



Es gibt viele verschiedene Automatentypen und damit viele verschiedene Maschinensprachen. Wir wollen uns auf eine einzige Maschine festlegen und obendrein diesen Automaten völlig frei erfinden. Es wäre hoffnungslos, die vollständige Befehlsliste eines konkreten Automaten im Rahmen dieses Kapitels ausführlich und verständlich beschreiben zu wollen; andererseits lässt sich das Grundsätzliche einer Automatensprache auch an einer vereinfachten, gedachten Rechenmaschine sehr deutlich zeigen.

Unser gedachter Rechner soll auf den Namen »Gedachter Rechenautomat 01«, kurz GR 01, hören und eine Sprache sprechen, deren Alphabet aus dem lateinischen Alphabet, den Ziffern 0 bis 9 und einigen Sonderzeichen besteht.

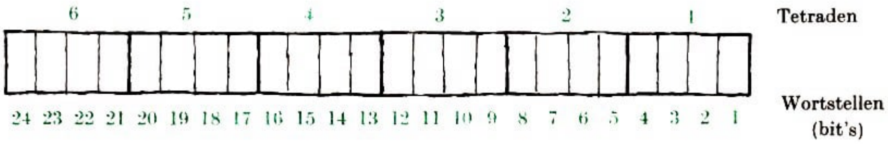
Verschlüsselung	Bedeutung	Verschlüsselung	Bedeutung	Verschlüsselung	Bedeutung
0000 0000	0	0000 000L	1	0000 00L0	2
0000 00LL	3	0000 0L00	4	0000 0L0L	5
0000 0LL0	6	0000 0LLL	7	0000 L000	8
0000 L00L	9	0000 L0L0	A	0000 L0LL	B
0000 LL00	C	0000 LL0L	D	0000 LLL0	E
0000 LLLL	F	000L 0000	G	000L 000L	H
000L 00L0	I	000L 00LL	J	000L 0L00	K
000L 0L0L	L	000L 0LL0	M	000L 0LLL	N
000L L000	O	000L L00L	P	000L L0L0	Q
000L L0LL	R	000L LL00	S	000L LL0L	T
000L LLL0	U	000L LLLL	V	00LL 0000	W
00LL 000L	X	00LL 00L0	Y	00LL 00LL	Z
00LL 0L00	.	00LL 0L0L	,	00LL 0LL0	!
00LL 0LLL	?	00LL L000	+	00LL L00L	-
00LL L0L0	·	00LL L0LL	:		



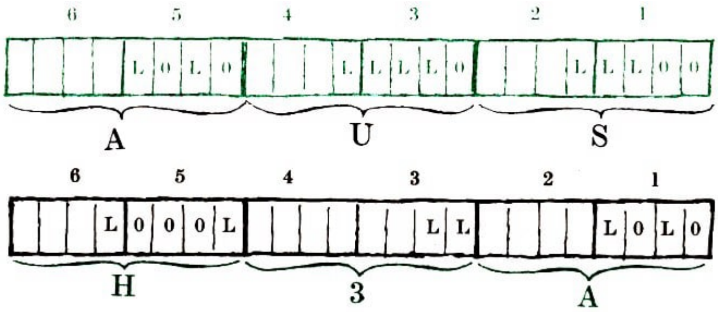
Ähnlich, wie das menschliche Gehirn das Zeichen 9 nicht als einfache Kopie in seinen Gehirnzellen enthält, speichert auch der Automat die 9 nicht als »9«, sondern als verschlüsselte Folge von Impulsen oder bestimmten magnetischen Zuständen ab, denen man - wie wir wissen - die Bedeutung der dualen 0 und 1 gibt. Die Verschlüsselung der Ziffern 0 bis 9 haben wir bereits kennengelernt.
Wenn wir an Stelle einer Tetrade zwei solcher Tetraden benutzen, haben wir mehr als genügend Platz zur Darstellung des gesamten lateinischen Alphabetes und der Sonderzeichen, wie es unsere Tabelle zeigt. Sie veranschaulicht gleichzeitig das Grundsätzliche der Zeichenkodierung. eines Rechenautomaten.

In allen Rechenmaschinen reichen 2 Tetraden für die Verschlüsselung eines Zeichens aus. Deswegen werden die 8 bit oder 2 Tetraden oft auch zusammengefasst als 1 Byte bezeichnet.

Die einzelnen Zeichen, die eine Rechenmaschine beherrscht, können nun zu Maschinenwörtern zusammengesetzt werden. Unsere gedachte Rechenmaschine soll eine feste Wortlänge von 6 Tetraden haben. In jedem Wort können dann 3 Zeichen untergebracht werden, was eine Gesamtlänge von $2 \cdot 3 \cdot 4 = 24$ Dualziffern ausmacht.



Das obere, nachfolgende Bild zeigt das vollständige leere Wort unserer gedachten Rechenmaschine. Die Zeichenfolge »Aus« wird wie im unteren Bild kodiert. Diese Zeichenfolgen heißen alphabetische Wörter.



Kommen zu den Buchstaben noch Ziffern dazu, spricht man von alpha-numerischen Wörtern, wie es im dritten Beispiel mit »H3A« gezeigt ist.
Mit derartigen Wörtern kann ein Automat natürlich nicht rechnen. Sie dienen vielmehr als Marken, als Beschriftungen beim Ausdruck von Tabellen und als Textbestandteil von Berichten und Mitteilungen.

7.9 Von bewegten und festen Kommas

Als nächstes soll die Zahlendarstellung des GR-01 erläutert werden. Wir unterscheiden zwei Arten von Darstellungen von Zahlen.
Zunächst wollen wir Zahlen von der Form

$$Z = M \cdot 10^E$$

behandeln. Dabei ist M die sogenannte Mantisse und E der Exponent. Die Mantisse setzt sich aus den Ziffern der Zahl zusammen, während der Exponent die Lage des Kommas bestimmt.
Die Mantisse soll so normiert sein, dass sie grundsätzlich in der Form »$0,...$« geschrieben wird, wobei die erste Stelle hinter dem Komma keine Null sein darf. Die Zahl 10 hätte danach die Darstellung

$$10 = 0{,}1 \cdot 10^2$$

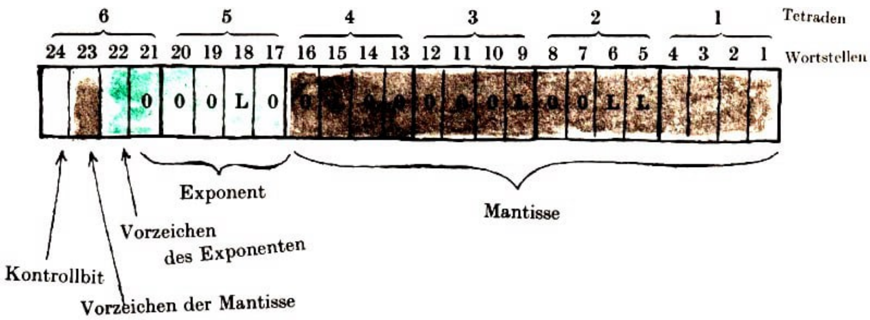
und die Zahl 412,5 müsste in der Form

$$412{,}5 = 0{,}4125 \cdot 10^3$$

geschrieben werden.
Im Maschinenwort unserer Anlage sollen dem Exponenten die Wortstellen der Tetrade 5 und die Wortstelle 21 der Tetrade 6 eingeräumt werden. Die Wortstelle 22 soll für das Vorzeichen des Exponenten vorgesehen sein, da wir ja auch mit Zahlen rechnen müssen, deren Wert kleiner als 1 sein kann, wie bei

$$0{,}8 = 0{,}8 \cdot 10^0, \qquad 0{,}05 = 0{,}5 \cdot 10^{-1}, \qquad 0{,}0002 = 0{,}2 \cdot 10^{-3}$$

Die Wortstelle 23 verwenden wir für das Vorzeichen der Gesamtzahl.
Die Tetraden 4, 3, 2 und 1 bleiben der Mantisse vorbehalten. Dabei soll jeder Tetrade eine Dezimalziffer in der bereits bekannten dezimal-dualen Verschlüsselung entsprechen. Mit diesen Festlegungen können wir die Zahl 41,3 in der folgenden Darstellung wiedererkennen.



Eine solche Zahlendarstellung nennt man auch Zahlendarstellung im beweglichen Komma oder im Gleitkomma, weil das Komma eine vom Exponenten bestimmte Lage hat, die von Zahl zu Zahl verschieden sein kann. Im gewählten Beispiel steht das Komma zwischen der ersten und der zweiten Tetrade.

Aus der Aufteilung des Maschinenwortes ist erkennbar, dass der GR-01 nur Zahlen darzustellen erlaubt, die aus 4 Ziffern bestehen. Längere Zahlen müssen ab- bzw. aufgerundet werden.

$$583679 \approx 0{,}5837 \cdot 10^6$$

Die größte auf dieser Rechenmaschine Platz findende Zahl ist

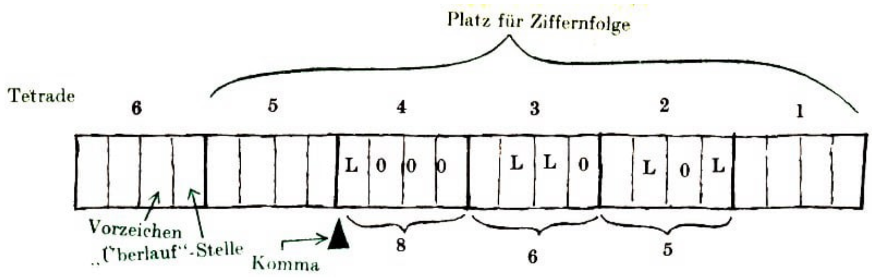
$$0{,}9999 \cdot 10^{19}$$

und die kleinste von Null verschiedene positive Zahl lautet

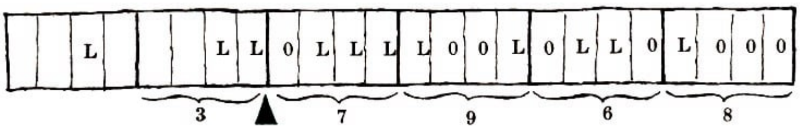
$$0{,}1 \cdot 10^{-19}$$

Für die Null im beweglichen Komma legen wir fest, dass sie aus der Mantisse 0 und dem Exponenten $E = 19$ bestehen soll.

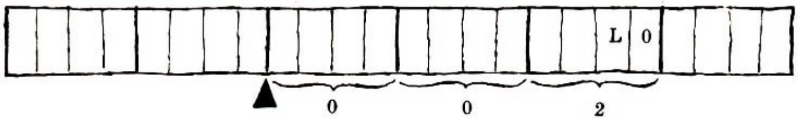
Die Wortstelle 24 wird schließlich für eine interne Kontrolle der Zahl verwendet. Man kann z. B. die Anzahl der besetzten Wortstellen immer auf eine ungerade Zahl ergänzen. Wenn dann im Verlauf des Transportes der Zahl die Anzahl der besetzten Wortstellen gerade wird, so ist die Zahl fehlerhaft übertragen worden. Der Automat kann dann selbständig geeignete Korrekturmaßnahmen ergreifen.



Neben der Darstellungsform des beweglichen Kommas kennen wir noch die des Festkommas. In dieser Darstellung wird die Zahl so in den Automaten eingegeben, wie sie geschrieben wird.



Man muss lediglich eine Stelle definieren, an der das Komma zu stehen hat. Setzen wir das Komma zwischen die vierte und fünfte Tetrade, so müssen die Zahlen dementsprechend eingegeben werden,



wie es unsere Beispiele für 0,865 oder -3,7968 und schließlich 0,002 zeigen. Dabei sind die einzelnen Dezimalziffern wieder in der dezimal-dualen Verschlüsselung geschrieben. Die sogenannte Überlaufstelle zeigt das Verlassen des Zahlenbereichs nach links an.

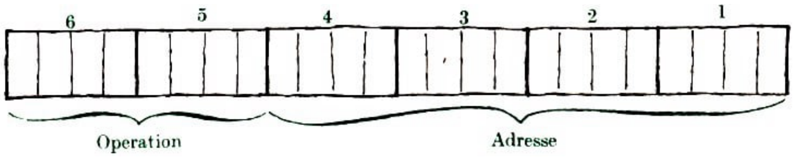
Mit diesen Zahlendarstellungen kann der Automat aber sowohl bei feststehendem als auch bei beweglichem Komma noch nicht rechnen. Die Zahlen müssen in einem Konvertierungsvorgang in die rein duale Schreibweise überführt werden. Erst in dieser Darstellung können alle befohlenen Operationen mit den eingegebenen Zahlen ausgeführt werden.

Vielfach sagt man, dass die Zahlen dann als »innere«, konvertierte oder auch kompakte Zahlen vorliegen. Die Bezeichnung »kompakt« rührt daher, dass eine ganze, rein dual geschriebene Zahl weniger Raum beansprucht als ein dezimaldual gespeicherter Wert.

Alle vorgestellten Wörter der Rechenmaschine waren daher sogenannte äußere Wörter. In diese Formen werden sie nach der Abtastung von der Lochkarte überführt. Auf der Lochkarte kann durch eine geeignete Kennzeichnung vermerkt sein, in welche der gezeigten Formen diese Umwandlung stattfinden soll.

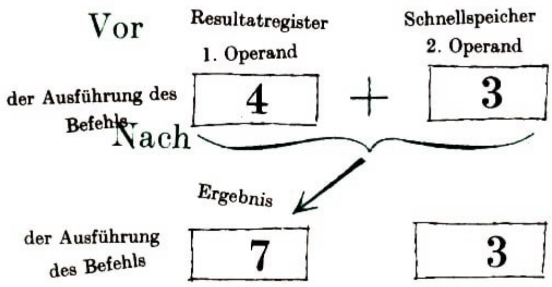
7.10 Im Befehlston

Der GR-01 sei eine Maschine, die einen Ferritkernspeicher mit einer Kapazität von 10000 Worten als Hauptspeicher besitzen soll. Die Zellen des Speichers seien von 0 bis 9999 nummeriert. Demzufolge werden 4 Tetraden für die Angabe der Adressen benötigt, es seien dies die Tetraden 1, 2, 3 und 4. Die Tetraden 5 und 6 bleiben dann zur Charakterisierung der eigentlichen Operation. Die Adresse bezieht sich auf den Hauptspeicher und benennt eine konkrete Speicherzelle, in der immer ein Maschinenwort Platz findet.



Neben dem Hauptspeicher soll der GR-01 ein Magnetbandgerät als Zusatzspeicher besitzen.
Bei gewissen Befehlen des GR-01 ist die Speicheradresse nicht erforderlich. Das bedeutet, dass diese Befehle sich auf bestimmte, fest vorgegebene Register beziehen oder gänzlich ohne Angabe einer Adresse auskommen.

Nun ist es aber an der Zeit, den Rechenautomaten auch wirklich rechnen zu lassen: Wir verlangen also, dass er die vier Grundrechenarten Addition, Subtraktion, Multiplikation und Division beherrscht. Diese 4 Operationen verknüpfen jeweils 2 Zahlen miteinander und bilden aus ihnen eine neue Zahl, die als Ergebnis der Operation bezeichnet wird.
In der gedachten Rechenmaschine soll der erste Operand, beispielsweise die Zahl 4 der Operation $4 + 3 = 7$, immer in einem besonderen Register, dem Resultatregister RR, aufbewahrt werden. Ein zweites Register, das wir Schnellspeicher SS nennen, nehme den zweiten Operanden auf.



Das Ergebnis der Rechenoperation soll wieder ins Resultatregister gelangen. Während nach der Ausführung der Operation die Zahl 4 im Resultatregister vom Ergebnis 7 überschrieben wird, bleibt der zweite Operand 3 im Schnellspeicher erhalten.

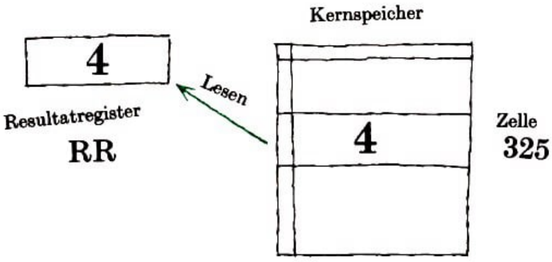
Dasselbe gilt für die anderen Grundrechenarten. Wir bezeichnen die entsprechenden Automatenbefehle mit den bekannten Symbolen $+$, $-$, $\times$ und :.

Wenn wir also der Maschine den Befehl $\times$ erteilen, so »weiß« sie, dass sie zwei Zahlen miteinander zu multiplizieren hat, von denen die erste im Resultatregister und die zweite im Schnellspeicher zu finden ist.

Wie wir uns erinnern, stehen vor Beginn einer Rechnung grundsätzlich alle Zahlen und Befehle zur Lösung der gegenwärtigen Aufgabe auf dem Hauptspeicher. Von dort werden die Zahlen in die beiden Register des Rechenwerks unseres Automaten überführt. Dazu bedarf es eines neuen Befehls, der den Namen »Lesen« tragen soll. Dieser Befehl ist nur in Verbindung mit einer Speicheradresse sinnvoll. Es soll also beispielsweise

"Lese 325" oder kürzer "L 325"

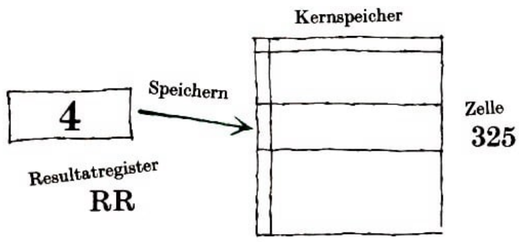
bedeuten, dass der Inhalt der Speicherzelle 325 in das Resultatregister geholt wird. Man beachte, dass es nicht heißt »Lese die Zahl 4«, sondern »Lese die Zahl, die unter der Adresse 325 zu finden ist«.



Das Aufsuchen von Zahlen durch Angabe ihrer Adresse ist ein für programmgesteuerte Automaten typischer Vorgang.
Die Umkehrung des Befehls »Lesen« stellt der Befehl »Speichern« dar. Auch er muss mit einer Adresse verknüpft werden. Der Befehl bewirkt die Überführung einer im Resultatregister vorliegenden Zahl in diejenige Zelle des Hauptspeichers, deren Adresse im Speicherbefehl angegeben wird.
Lautet die Anweisung

"Speichere nach 325" oder kürzer "S 325"

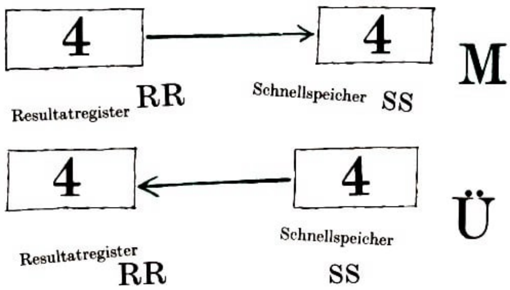
so gelangt nach der Abarbeitung dieses Befehls die sich im Resultatregister befindende Zahl auf den Speicherplatz mit der Adresse 325 des Hauptspeichers. Dabei wird grundsätzlich der alte Inhalt der Zelle 325 gelöscht und mit dem neuen Inhalt überschrieben.



Neben den Befehlen für das Lesen und Speichern vom bzw. auf den Hauptspeicher muss die Rechenmaschine auch Befehle für den Informationstransport zwischen Magnetband und Arbeitsspeicher besitzen.
Auf diese Befehle soll später eingegangen werden, weil sie den Informationstransport zwischen den peripheren Geräten und der Zentraleinheit besorgen und nicht zu den hier besprochenen Befehlen der Zentraleinheit gehören.

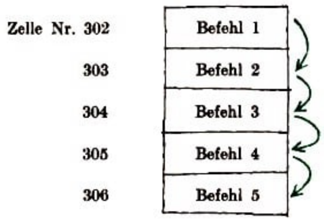
Zur Ausführung einer Rechenoperation benötigen wir den zweiten Operanden im Schnellspeicher. Das Füllen des Schnellspeichers soll durch den Befehl »Merke« oder kurz »M« erfolgen. Dabei gelangt eine im Resultatregister stehende Zahl in den Schnellspeicher.

Der in der entgegengesetzten Richtung erfolgende Transport einer Zahl soll durch den Befehl »Übertrage aus dem Schnellspeicher« - kurz »Ü« - bewirkt werden.



Die soeben beschriebene Gruppe von Befehlen veranlasste den Transport von Zahlen von einem Speicherplatz zu einem anderen.

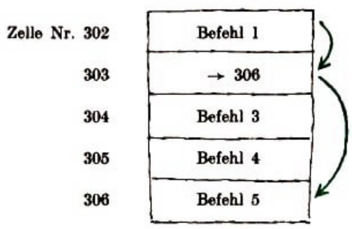
Eine weitere wichtige Gruppe von Befehlen besteht aus den sogenannten Sprungbefehlen.
Jeder auf den Ferritkernspeicher notierte Befehl befindet sich in einer bestimmten Zelle, die durch ihre Adresse festgelegt ist. Die Gesamtheit aller gespeicherten Befehle bildet das Programm. Im Normalfall erfolgt die Abarbeitung der Befehle in derselben Reihenfolge, in der sie gespeichert sind.



Besteht die Notwendigkeit, diese Reihenfolge zu durchbrechen, um z. B. einige Befehle zu überspringen, so verwendet man den Befehl »Springe in die Zeile, deren Adresse angegeben wird«. Dieser Befehl soll in der Kurzfassung mit einem Pfeil »$\rightarrow$« gekennzeichnet sein. Demnach bedeutet

$$\rightarrow 306$$

eine Anweisung für den Automaten, die Befehlsabarbeitung mit der Zelle 306 fortzuführen.
Das setzt voraus, dass auf der Speicherzelle 306 ein sinnvoller Befehl zu finden ist. Keineswegs darf dort eine Zahl stehen, da sie, vom Automaten als Befehl aufgefasst, ungewollte und höchst merkwürdige Operationen auslösen würde. In unserem Beispiel

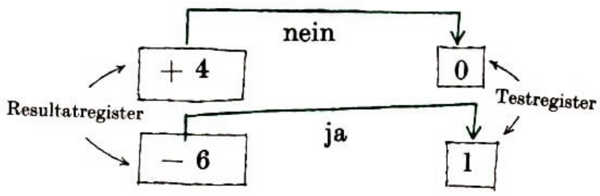


liegt in der Zelle 303 ein Sprungbefehl vor. Er bewirkt, dass der nächste abzuarbeitende Befehl aus der Zelle 306 geholt wird.

Erweist sich der dort angetroffene Befehl nicht seinerseits als Sprungbefehl, so erfolgt

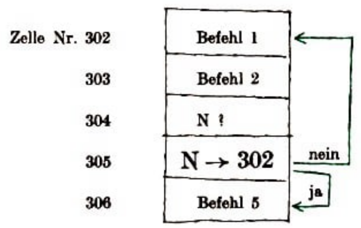
die weitere Verarbeitung der Befehle in der Reihenfolge der belegten Speicherplätze. Diese Reihenfolge wird aber wieder unterbrochen, wenn ein Sprungbefehl entschlüsselt wird.
Neben diesem Sprungbefehl, der ohne jede vorher gestellte Bedingung ausgeführt wird, kennen wir noch den sogenannten bedingten Sprung, der nur dann erfolgt, wenn eine zuvor gestellte Frage in einem festgelegten Sinn beantwortet wird. Für das Verständnis des bedingten Sprungbefehls ist es besser, wenn wir zunächst die zu stellende Frage oder, wie man sagt, den dazugehörigen Testbefehl kennenlernen.

Der Rechenautomat GR-01 soll feststellen können, ob eine im Resultatregister stehende Zahl positiv oder negativ ist. Nach dem Erlass dieses Testbefehls »Ist die Zahl negativ?« speichert der Automat die Antwort »Ja« oder »Nein« in ein besonderes Register, das Testregister, das mit einer Kapazität von nur 1 bit auskommt. Im Fall der Antwort »Ja« hat es eine 1 und im Fall der Antwort »Nein« eine 0 aufzunehmen.



So beantwortet der Automat die Frage für die Zahl +4 mit »Nein« und füllt das Testregister mit der Dualzahl 0. Bei der Zahl -6 erkennt er das negative Vorzeichen und bringt infolgedessen die Dualzahl 1 in das Testregister. Der Befehl soll die Abkürzung »N ?« erhalten.

Im Anschluss an den Testbefehl kann nun ein bedingter Sprung befohlen werden, der nur dann auszuführen ist, wenn die Frage mit »Nein« beantwortet wird. Deswegen soll der Befehl auch »Nein- Sprung« heißen und mit dem Symbol »N→« gekennzeichnet sein. In unserem Beispiel wird bei »Nein« der Befehl 1 angesprungen.



Liegt die Antwort »Ja« vor, ist also die Zahl negativ, so soll der Sprung nicht ausgeführt werden. In diesem Fall ist der nächste abzuarbeitende Befehl derjenige, der unmittelbar auf den bedingten Sprungbefehl folgt.
Liegt also die Befehlskombination

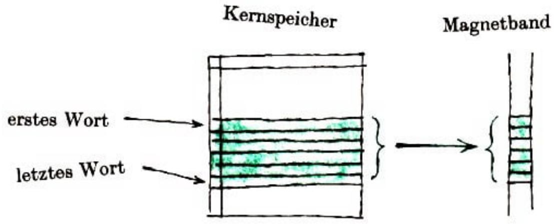
$$\begin{gathered}\text{N?}\\ \text{N} \rightarrow 302\end{gathered}$$

vor, so sind in Abhängigkeit von der Antwort auf die erste Frage zwei Fortsetzungen des Programms möglich. Wird der Test mit »Nein« beantwortet, »springt« der Automat mit der Befehlsabarbeitung in die Zeile 302.

Ergibt sich die Antwort »Ja«, folgt in der regulären Reihenfolge die Durchführung des in der Zelle 306 stehenden Befehls.

Der Testbefehl und der bedingte Sprungbefehl sind die eigentlichen Ursachen für die vielen Anwendungsmöglichkeiten des Rechenautomaten. Mit Hilfe dieser beiden Befehle lassen sich im Programm viele Verzweigungen und Schleifen realisieren, ohne die eine Lösung komplizierter Aufgaben undenkbar wäre.
Wir wollen die Beschreibung der Befehle des Automaten GR-01 fortsetzen, indem wir den Druckbefehl und die Magnetbandbefehle erwähnen. Der Befehl »Drucke« oder kurz »D« bewirkt die Ausgabe einer im Resultatregister stehenden Zahl auf die Papierrolle des Druckers.

Für das Magnetband benötigen wir zwei Befehle: einen zum Schreiben und einen zum Lesen. Der Schreibbefehl bewirke, dass eine Reihe von Worten der Maschine, beginnend mit dem Wort, dessen Adresse im Resultatregister RR steht, auf das Magnetband übertragen wird. Die Adresse des letzten zu übertragenden Wortes stehe im Schnellspeicher SS. Der so begrenzte Informationsblock wird in einem Transportvorgang auf das Magnetband geschrieben.



Umgekehrt bewirke der Befehl Lesen, dass ein vorgegebener Speicherbereich im Hauptspeicher mit Anfangsadresse im RR und Endadresse im SS durch Angaben des Magnetbandes gefüllt wird.
Die betreffenden Stellen auf dem Magnetband sollen sowohl für das Schreiben als auch für das Lesen gerade diejenigen sein, die in der Arbeitsrichtung des Bandes als nächste an den Lese-Schreibköpfen vorübergleiten. Aus diesem Grunde ist bei beiden Befehlen keine Adressenangabe für das Magnetband erforderlich.

Mit dem Befehl »Halt« oder kurz »H« soll die Rechenmaschine GR-01 schließlich die Arbeit an einer Aufgabe beenden.

7.11 Der Computer aus Radeberg

Um zu prüfen, wie weit wir uns mit unserer gedachten Rechenmaschine GR-01 von der Realität entfernt haben, wollen wir zum Abschluss dieses Kapitels einen kurzen Blick auf eine wirklich existierende Maschine werfen. Es soll sich dabei um die in Karl-Marx-Stadt entwickelte und in Radeberg produzierte Anlage Robotron 300 handeln.

Diese Maschine besitzt als Haupt- oder Operativspeicher einen Ferritkernspeicher mit einer Kapazität von 40000 alpha-numerischen Zeichen. Zur Verschlüsselung eines Befehls werden 6 Zeichen benötigt, so dass der Kernspeicher über 6600 Befehle fassen kann.

Das Rechenwerk gestattet die Ausführung aller arithmetischen Operationen im Festkomma; durch eine Rechenwerkergänzung werden auch verdrahtete Operationen im Gleitkomma möglich, ohne dass diese mit Hilfe von Unterprogrammen realisiert werden müssen.

Die Wortlänge des R 300 ist - abweichend vom GR-01 - variabel. Spezielle Wortmarken begrenzen das Wort des R 300, so dass es den Bedürfnissen des abzuarbeitenden Problems angepasst werden kann und damit auch eine sehr rationelle Speicherausnutzung ermöglicht.
Die Geschwindigkeit der Befehlsabarbeitung liegt zwischen 0,2 und 7 ms für die Operationen der Zentraleinheit.

Die peripheren Geräte, die mit Lochkarten oder Lochstreifen arbeiten, sind natürlich bedeutend langsamer. Aus diesem Grunde befindet sich in der Maschine ein sogenanntes Unterbrechungssystem, das die zeitlich ineinander verschachtelte Arbeit von Ein- und Ausgabeprogrammen mit rechenintensiven Auswertungsprogrammen ermöglicht.

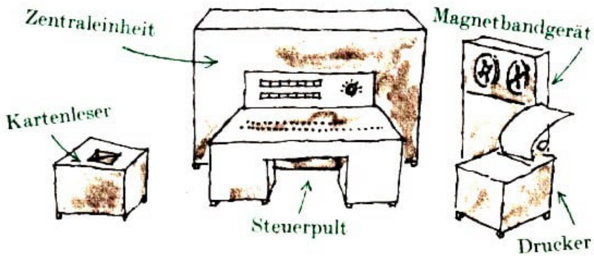
Der R 300 kann in der Stunde 18000 80spaltige Lochkarten lesen und ebensoviel stanzen, wobei die beiden Arbeitsgänge auf dem gleichen Gerät, der Lese-Stanzeinheit, durchgeführt werden. Für die Ein- und Ausgabe mit Hilfe des Lochstreifens liegen die Geschwindigkeiten bei 300 Zeichen in der Sekunde.
Der Drucker des R 300 ermöglicht eine Druckbreite von 156 Zeichen, was etwa 2 Zeilen des A 4-Formates entspricht. Mit einer Geschwindigkeit von 300 Zeilen je Minute kann das Gerät damit etwa 20 Seiten im Format A 4 in jeder Minute ausgeben. In jeder Stunde könnte so ein dickleibiges Buch fabriziert werden.

Auch die Möglichkeiten zur Informationsspeicherung sind beeindruckend. An den R 300 sind bis zu 8 Magnetbandgeräte anschließbar, wovon 6 unmittelbar mit der Anlage verbunden werden können, während 2 in Warteposition stehen müssen. Der Informationsaustausch mit der Zentraleinheit erfolgt mit einer Geschwindigkeit von 20000 Zeichen/s. Die Länge der verwendeten Bänder beträgt 750 m. Mit einer Zeichendichte von 12 Zeichen/mm könnte man damit auf einem Band theoretisch 9 Millionen Zeichen speichern.
Selbst die praktisch erzielbare Kapazität von 6 Millionen Zeichen ist immer noch riesig groß, sie entspricht der Kapazität von 75000 80spaltigen Lochkarten. Die Lochkarten werden in Kästen zu je 2000 Stück geliefert; das bedeutet, dass 37,5 solcher Kästen benötigt werden, um ein Magnetband zu beschreiben.

Als weitere Zusatzspeicher sind an den R 300 auch Magnettrommelspeicher anschließbar. Unser Bild zeigt das Äußere des Automaten.
Wir erkennen im Vordergrund links die Lese-Stanzeinheit, in der Mitte das Bedienungspult mit Lampenregistern und Schaltern und rechts den Drucker, dem die Ergebnisse entnommen werden. Im Hintergrund stehen als Schrank ausgebildet der ei-

gentliche Rechenautomat oder, wie man auch sagt, die Zentraleinheit und daneben die Geräte für die Magnetbandspeicherung. Wenn wir die Türen des zentralen Schrankes öffnen, so erblicken wir eine Unmenge elektrischer Schaltungen des Rechenwerkes und die einzelnen Blöcke des inneren Speichers des Automaten.
Hier liegt das Reich der Wartungsingenieure und Maschinenmathematiker, die sich in den »Eingeweiden« der Rechenmaschine zurechtfinden müssen, um bei Fehlern schnell eingreifen zu können.

Wenn man erfährt, dass der R 300 trotz der genannten großen Leistungszahlen nur als mittlere elektronische Datenverarbeitungsanlage bezeichnet werden kann, erhält man eine ungefähre Vorstellung von der Größe und Schnelligkeit heute verbreiteter Rechenautomaten. Operativspeicher mit einer Kapazität von 100000 bis 500000 Zeichen sind keine Seltenheit mehr.
Äußere Zusatzspeicher fassen längst eine halbe Milliarde Zeichen, und die Rechenoperationen benötigen nur noch wenige Mikrosekunden zu ihrer Ausführung.



Wie nutzt man die gewaltigen Möglichkeiten solcher Maschinen zur Informationsspeicherung und Informationsverarbeitung? Zur Beantwortung dieser Frage müssen wir zunächst die Technik des Programmierens kennenlernen.

8 Wie unterhält man sich mit Maschinen?

8.1 Der unfähige Programmierer

In diesem Kapitel geht es um das Problem, wie man sich programmgesteuerten Automaten verständlich macht, wie also ein Programm im einzelnen auszusehen hat.
Der einfachste Fall liegt dann vor, wenn wir das Programm im direkten Kode des Automaten schreiben. Dabei bezieht sich das Wort »einfach« nicht so sehr auf den Menschen als vielmehr auf den Automaten. Der geschickteste Programmierer ist seltsamerweise nicht in der Lage, ein längeres Programm in der unmittelbaren Sprache des Automaten ohne weiteres niederzuschreiben.

Er muss zunächst das zu programmierende Problem in einer ihm geläufigen Form sichtbar vor sich haben. Das eigentliche Programmieren vollzieht sich dann als formale Übersetzung aus dieser vom Menschen leicht beherrschten Sprache in die kleinliche, schwerfällige Ausdrucksweise des Automaten, was dann bereits als bloßes Kodieren bezeichnet werden muss.

Natürlich beherrscht der Programmierer auch die Ausdrucksweise des Automaten, sie ist ihm aber unbequem. Tatbestände, die er in seiner ihm geläufigen Sprache mit einem einzigen Wort charakterisiert, bedürfen in der Automatensprache einer aus vielen Wörtern bestehenden umständlichen Beschreibung.
Aus diesem Grunde denkt und schreibt der Programmierer nicht in der Automatensprache, sondern übersetzt rein formal eine ihm vertraute Sprache in die des Automaten, wobei er sich bemüht, keine falschen »grammatischen« Konstruktionen zu verwenden und die orthographischen Flüchtigkeitsfehler auf ein Mindestmaß zu beschränken, ein Vorhaben, das ihm auf Anhieb selten vollständig gelingt.

Damit haben wir das Programmieren zunächst ausreichend charakterisiert.
Wir verstehen darunter einen formalen Übersetzungsvorgang eines Algorithmus aus einer dem Menschen geläufigen Schreibweise in die Ausdrucksform des Rechenautomaten.

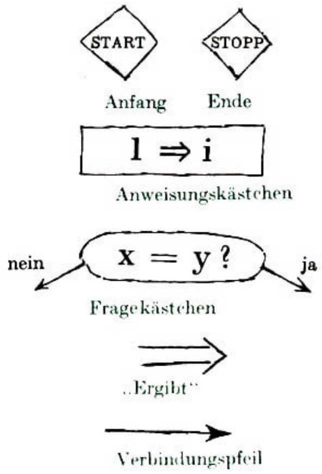
8.2 Aus einem Programmablaufplan...

Zunächst kommt es darauf an, eine vom Menschen leicht beherrschbare Form der Beschreibung von Algorithmen zu finden. Die Umgangssprache ist dazu wegen ihrer häufigen Mehrdeutigkeit und wegen des uns aufgelegten Zwanges, in zusammenhängenden, grammatisch vollständigen, d. h. in mitunter recht langen Sätzen zu sprechen, wenig geeignet.

Man verfiel daher auf den Gedanken, eine einfache Zeichensprache zu benutzen, die aus nur wenigen Grundzeichen besteht und deren Übersichtlichkeit deshalb sehr eindrucksvoll ist. In der folgenden Abbildung sind die wichtigsten Zeichen dieser Sprache zusammengestellt.

Für die Symbole dieser Zeichensprache existieren inzwischen in der DDR TGL-Vorschriften.
Das erleichtert ungemein die Austauschbarkeit der aus den Einzelzeichen entstehenden,

oft sehr komplizierten Diagramme. Mit diesen wenigen Zeichen lässt sich nun jeder beliebige Algorithmus formulieren.
Die Zeichen für Anfang und Ende bedürfen keiner näheren Erläuterung. Sie stehen am Anfang bzw. am Ende der Beschreibung eines Algorithmus.



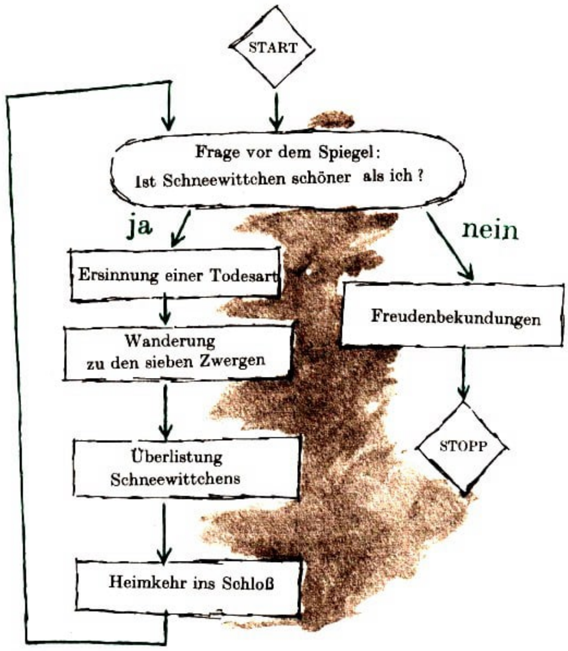
Das Anweisungskästchen lässt sich nur im Zusammenhang mit dem »Ergibt«-Zeichen erklären. Das »Ergibt«-Zeichen ist aus dem Gleichheitszeichen hervorgegangen und hat eine ähnliche Bedeutung wie jenes. Wenn wir sagen wollen, dass die Unbekannte x den Wert 3 besitzt, so schreiben wir $x = 3$.
Wenn man dagegen die Gleichheit als Gleichsetzung auffasst, schreibt man $3 \Rightarrow x$ und liest »3 wird x« oder »3 ergibt x«. Noch deutlicher tritt der Grund für die letzte Ausdrucksweise hervor, wenn ein aus mehreren mathematischen Einzelzeichen bestehender Ausdruck der kürzeren Schreibweise wegen mit einem neuen Symbol bezeichnet werden soll. Man schreibt

$$(3 \cdot x) : 5 \Rightarrow y$$

und liest: »Das Dreifache der Zahl x dividiert durch 5 wird der Wert y.«
Die eigentliche Ursache für die von der üblichen Norm abweichende Schreibweise des Gleichheitszeichen liegt aber darin, dass allen Symbolen für Zahlen wie den Zahlen selbst im Rechenautomaten konkrete Speicherzellen zugeordnet werden. Der Ausdruck »$3 \Rightarrow x$« besagt, dass die Zahl 3 auf die Stelle des Speichers gesetzt werden soll, die der Unbekannten x vorbehalten war.
Ähnlich bedeutet $(3 \cdot x) : 5 \Rightarrow y$, dass das Ergebnis der links stehenden Rechenoperation auf den Speicherplatz der Zahl y geschafft werden soll.
Solche Aussagen oder Anweisungen werden im Anweisungskästchen untergebracht.

Das Alternativ- oder Fragekästchen hat zwei Ausgänge, die den beiden möglichen Antworten auf eine Ja-Nein-Frage entsprechen.
In ihm finden etwa folgende Fragen Platz: »Ist $x = 3$?« oder »Ist der Wert y kleiner als eine gewisse vorgegebene Zahl a?«
Die Pfeile sorgen schließlich für eine richtige Verbindung der einzelnen Kästchen. Die dabei entstehenden Bilder nennt man Flussdiagramme oder Programmablaufpläne.

Als Beispiel soll ein Flussdiagramm der Handlungen der bösen Königin aus dem Märchen Schneewittchen angegeben werden.



Wir erkennen deutlich den teuflischen Zyklus der Mordversuche an der schönen Königstochter, der erst dann ein Ende hat, wenn die am Anfang gestellte Frage mit »nein« beantwortet wird.

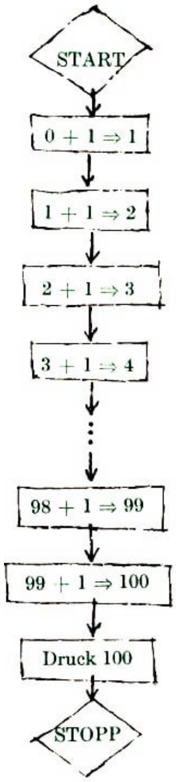
So verwerflich dieser Zyklus im Beispiel auch sein mag, für das elektronische Rechnen hat er eine entscheidende Bedeutung. Ein Problem, dessen Flussdiagramm keine solchen Zyklen oder Schleifen aufweist, die mehrmals durchlaufen werden müssen, ist für die Lösung auf einem Automaten ungeeignet; es sei denn, dass sehr viel Eingabedaten anfallen, die alle fortlaufend nach demselben Schema verarbeitet werden müssen.

Im allgemeinen ist ein gestrecktes Programm, wie wir ein Programm ohne Schleifen nennen, infolge der ungeheuren Rechengeschwindigkeiten unrentabel.

Während die Programmierung und das Lochen der Eingabewerte in so einem Fall höchstwahrscheinlich viele Stunden beanspruchen werden, kann die eigentliche Durchrechnung bereits in wenigen Sekunden erfolgen. Oft ist ein solches Problem mit gewöhnlichen Tischrechenmaschinen in einer Zeit zu bewältigen, die nicht wesentlich größer ist als die Vorbereitungszeit für das elektronische Rechnen.

Daher weisen die Flussdiagramme sämtlicher von Rechenautomaten durchgeführten Rechnungen viele Schleifen auf, die so lange durchlaufen werden müssen, bis ein an den Anfang oder an das Ende gestelltes Kriterium erfüllt ist.

Es sei uns das »Problem« gestellt, die Zahl 1 hundertmal zu addieren und das Ergebnis zu drucken. Wir wollen diese Aufgabe ernst nehmen, weil wir von ihrer Lösung einen Einblick in die Programmiertechnik erhoffen. Zunächst muss ein Flussdiagramm eines Algorithmus gesucht werden, der die Durchführung aller Additionen anordnet.



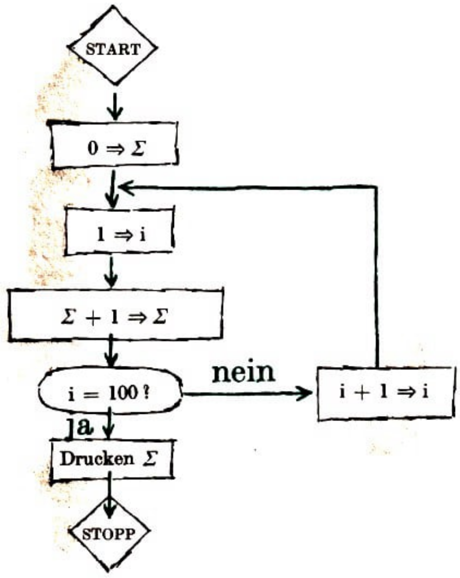
Die einfachste Form eines solchen Flussdiagramms wäre offenbar die nebenstehende.

Da uns die Nachteile eines gestreckten Programms bekannt sind, versuchen wir eine Schleife zu konstruieren, die uns von der hintereinander erfolgenden Aufzählung aller einzelnen Schritte entbindet. Eine solche Schleife müsste dann genau 100mal durchlaufen werden. In unserem Flussdiagramm muss also eine Zählung erfolgen.

Das kann realisiert werden, indem wir der laufenden Nummer der Zählschritte das allgemeine Symbol i verleihen und in jedem Schritt die Frage »$i = 100?$« stellen.
Weiterhin benötigen wir ein Symbol für die ständig wachsende Summe, die am Ende die Summe von hundert Einsen, also 100 sein wird. Dieses Symbol soll das Zeichen $\sum$ sein.

Am Anfang der Rechnung soll die Summe $\sum$ Null sein. Die Zählung beginne mit der Nummer $i = 1$. In jedem Schritt wird die auflaufende Summe um 1 erhöht. Das ist in der Form

$$\sum +1 = \sum$$



geschrieben. Der bisher vorliegende alte Wert von $\sum$ ergibt also, um 1 erhöht, den jeweils neuen Wert von $\sum$.

Im nächsten Kästchen steht die Frage »$i = 100?$«. Solange sie mit »nein« beantwortet wird, muss der Zählindex um 1 erhöht werden. Hat die Weiterzählung um 1 stattgefunden, so kann wieder die Summe vergrößert werden. Der Pfeil, der vom Kästchen $i + 1 \Rightarrow i$ ausgeht, weist infolgedessen auf den Anfang des Kästchens der Summenerhöhung.

Auf diese Weise erhalten wir einen Zyklus, der 100mal durchlaufen wird. Ist schließlich $i = 100$ erreicht, haben wir die gestellte Aufgabe gelöst, lassen das Ergebnis ausdrucken und setzen das Symbol »Ende der Rechnung« an den Schluss des Flussdiagramms.

8.3 … wird ein Programm für unsere Rechenmaschine

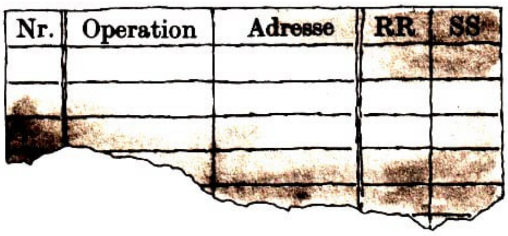
Dieses Flussdiagramm soll nun in ein Programm für die im vorigen Kapitel beschriebene Rechenmaschine übersetzt werden. Zur Erleichterung der Aufgabe fassen wir alle Befehle dieser Maschine in einer Liste zusammen, der wir noch die Kodierung der Befehle für die Übertragung auf Lochkarten beifügen.

Befehlsliste des GR-01

Befehl	Bedeutung	Kodierung Operationsteil
+ Addition	Die im Schnellspeicher stehende Zahl wird zu der im Resultatregister stehenden Zahl addiert. Das Ergebnis gelangt ins Resultatregister.	1
− Subtraktion	Von der im Resultatregister stehenden Zahl wird die im Schnellspeicher stehende Zahl subtrahiert. Das Ergebnis gelangt ins Resultatregister.	2
× Multiplikation	Die im Resultatregister stehende Zahl ist mit der im Schnellspeicher stehenden Zahl zu multiplizieren. Das Ergebnis gelangt ins Resultatregister.	3
: Division	Die im Resultatregister stehende Zahl Division wird durch die im Schnellspeicher stehende Zahl dividiert. Das Ergebnis gelangt ins Resultatregister.	4
L Lesen	Die Zahl, die unter der angegebenen Lesen Adresse auf dem Hauptspeicher zu finden ist, wird ins Resultatregister gebracht.	5
S Speichern	Die im Resultatregister stehende Zahl wird in der durch die angegebene Adresse bestimmten Zelle des Hauptspeichers gespeichert.	6
Ü Übertragen	Die Zahl, die sich im Schnellspeicher befindet, wird ins Resultatregister gebracht.	7
M Merken	Die im Resultatregister stehende Zahl wird in den Schnellspeicher gebracht.	8
N? Ergebnis negativ?	Test: Ist die im Resultatregister stehende Zahl negativ? Die Antwort ja (1) oder nein (0) gelangt ins Testregister.	9
N→ Nein-Sprung	Wird der Test mit »nein« beantwortet so wird als nächster Befehl derjenige ausgeführt, der unter der angegebenen Adresse zu finden ist. Bei der Beantwortung mit »ja« folgt der in der regulären Reihenfolge nächste Befehl.	10
→ Sprung	Der nächste abzuarbeitende Befehl findet sich unter der angegebenen Adresse.	11
D Drucken	Die im Resultatregister stehende Zahl wird ausgedruckt.	12

Befehl	Bedeutung	Kodierung Operationsteil
H Halt	Nach Entschlüsselung dieses Befehls hält der Automat an.	13
SM Schreibe auf auf Magnetband	Ein Speicherbereich, dessen Anfangsadresse im RR und dessen Endadresse im SS steht, wird auf das Band geschrieben.	14
LM Lies vom Magnetband	Ein Speicherbereich auf dem Magnetband wird in den Hauptspeicher gebracht. Die Anfangsadresse des Hauptspeicherbereiches steht im RR, die Endadresse im SS.	15

Weiter benötigen wir ein Programmierungsformular, dem wir die folgende Gestalt geben wollen.



Nr.	Operation	Adresse	RR	SS

Operation und Adresse im mittleren Teil des Formulars gelangen dann später auf die Lochkarte.
Bevor wir mit der eigentlichen Programmierung beginnen, müssen einige wichtige Bemerkungen gemacht werden. Wir wissen, dass das später vorliegende Programm in den Hauptspeicher des Automaten gelangt. Daneben müssen auch alle Zahlen, die wir zur Lösung des Problems benötigen, auf den Speicher gebracht werden.

In unserem Beispiel sind die Zahlen 0, 1 und 100 erforderlich. Wir legen willkürlich fest, dass sich diese Zahlen in den Zellen 300, 301 und 302 befinden sollen.

Zahl	Zelle Nr.
0	300
1	301
100	302

Da wir weiter wissen, dass die Symbole i und $\sum$ jeweils ganz bestimmte Zahlen verkörpern, müssen wir auch für diese Symbole Speicherzellen reservieren. Wir wählen beispielsweise die Zellen 303 und 304:

Zahl	Zelle Nr.
i	303
$\sum$	304

Die eben beschriebene Speicherplatzzuweisung für alle im Flussdiagramm vorkommenden Zahlen und Symbole, die Zahlen verkörpern, kann man als das Grundgesetz der Programmierung betrachten.

Jede Zahl befindet sich stets entweder in einem Register oder in einer Zelle des Hauptspeichers und hat dort eine bestimmte Adresse, unter der sie jederzeit zu erreichen ist.

Nun endlich kann das Programm geschrieben werden. Wir beginnen mit der ersten Anweisung des Flussdiagramms, die verlangt, dass wir der Summe $\sum$ den Wert 0 erteilen. Das entspricht aber einer Überführung der Zahl 0 aus der Zelle 300, in der wir sie vorfinden, in die Zelle 304, in der ja laut Speicherbelegung die Summe $\sum$ gebildet werden soll. Die Überführung mit einem einzigen Befehl ist auf unserem Automaten nicht möglich.
Wir müssen die Zahl 0 zunächst ins Resultatregister holen, um sie von dort aus in die Zelle 304 zu bringen. Das Programm beginnt demnach mit folgenden Befehlen:

Nr.	Operation	Adresse	RR	SS
0	L	300	0	
1	S	304	0	

Nach der Ausführung des ersten Befehls gelangt die Zahl 0 ins Resultatregister, der zweite Befehl bewirkt die Überführung der Zahl 0 aus dem Resultatregister in die Speicherzelle 304. (Wir haben dem ersten Befehl die Nummer 0 zugewiesen. Das wird sich später als recht sinnvoll erweisen.)

Die zweite Anweisung des Flussdiagramms verlangt, dass der Zählindex i den Wert 1 annehmen soll. Wir müssen also dafür sorgen, dass die Zahl 1 aus der Zelle 301 in die Zelle 303 gelangt, da die Zelle 303 für die Aufnahme der Zählgröße i vorgesehen ist. Auf diese Weise entsteht die folgende Fortsetzung des Programms:

Nr.	Operation	Adresse	RR	SS
...	...	...	...	...
2	L	301	1	
3	S	303	1	

An dieser Stelle sei noch einmal daran erinnert, dass die von uns benutzte Rechenmaschine so konstruiert ist, dass eine Zahl nach dem Lesen unzerstört auf ihrem Speicherplatz erhalten bleibt. Es wird gewissermaßen eine Kopie dieser Zahl angefertigt und in das Resultatregister gebracht.
Grundsätzlich bleibt der Inhalt eines Registers oder einer Zelle des Hauptspeichers bis zu dem Zeitpunkt erhalten, an dem eine Neufüllung des Registers oder der Zelle mit einer anderen Zahl erfolgt.

In der nächsten Anweisung des Flussdiagramms wird die Erhöhung der Summe 2 um 1 verlangt. Die Zahl 1 steht bereits im Resultatregister, wohin sie im Ergebnis der vorhergehenden Leseoperation gelangt ist. Wir überführen sie durch den Befehl »Merke« in den Schnellspeicher und holen uns die Summe $\sum$ von der Trommel ins Resultatregister. Das ergibt zwei weitere Befehlszeilen:

Nr.	Operation	Adresse	RR	SS
...	...	...	...	...
4	M		1	1
5	L	304	$\sum$alt	1

Dabei handelt es sich bei der Summe $\sum$ noch um die alte, nicht um 1 erhöhte Summe, was durch die Schreibweise $\sum$alt zum Ausdruck kommen soll. Nun kann die Addition durchgeführt werden.
Als Ergebnis entsteht im Resultatregister der neue Wert der Summe. Diese neue Summe wird anschließend wieder auf den ihr zugewiesenen Speicherplatz 304 zurückgebracht:

Nr.	Operation	Adresse	RR	SS
...	...	...	...	...
6	$+$		$\sum$neu	1
7	S	304	$\sum$alt	1

Im Flussdiagramm steht jetzt die Frage »Ist $i = 100$?«. Ihre programmtechnische Realisierung muss auf dem Testbefehl N? basieren.
Bilden wir ständig die Differenz $i - 100$, so haben wir eine geeignete Testgröße zur Verfügung, die für alle ganzen Zahlen von 1 bis 99 negativ ist und erst für $i = 100$ den als positiv geltenden Wert 0 erreicht. Wir bilden also im Programm diese Differenz $i - 100$ und fragen, ob sie negativ ist:

Nr.	Operation	Adresse	RR	SS
...	...	...	...	...
8	L	302	100	1
9	M		100	100
10	L	303	i	100
11	-		$i - 100$	100
12	N?		$i - 100$	100

An dieser Stelle sieht das Flussdiagramm zwei Möglichkeiten zur Fortsetzung der Rechnung vor. Wenn i den Wert 100 erreicht, d. h., sobald die Differenz $i - 100$ nicht mehr negativ ist, soll die Summe gedruckt werden und der Automat anhalten.
Im anderen Fall soll i um 1 erhöht und die Rechnung an der im Flussdiagramm bezeichneten Stelle weitergeführt werden.

Dabei ergibt sich eine Schwierigkeit. Der Programmteil, der das Drucken und das Anhalten des Automaten bedingt, ist noch nicht geschrieben. Wir wissen demzufolge nicht, mit welcher Programmzeile er beginnt. Da die Programmzeilennummern nach der Eingabe des Programms auf den Ferritkernspeicher zu Adressen werden, können wir nicht sagen, wie diese Adresse lautet, unter der der Schlussteil unseres Programms zu finden ist.

Das ist aber gerade die Adresse, die der Nein-Sprung ansprechen soll. Wir sind daher gezwungen, diese Adresse im Befehl zunächst freizulassen:

Nr.	Operation	Adresse	RR	SS
...	...	...	...	...
13	N$\rightarrow$	?	$i - 100$	100

Wird der Test mit »ja« beantwortet, so führt der Automat den Nein- Sprung nicht aus, sondern arbeitet den darauffolgenden Befehl ab. Wir können also den Programmteil folgen lassen, der die Erhöhung von i um 1 bewirkt:

Nr.	Operation	Adresse	RR	SS
...	...	...	...	...
14	L	303	i	100
15	M			i
16	L	301	1	i
17	$+$		$i+1$	i
18	S	303	$i+1$	i

Nun fehlt noch der Sprung an die Stelle im Flussdiagramm, an der die Erhöhung der Summe $\sum$ um 1 stattfindet. Wir müssen demnach in unserem bisher geschriebenen Programm nach dieser Stelle suchen. Nach einiger Mühe entdecken wir sie in der Programmzeile 5, in der die alte Summe gelesen wird, um danach um 1 erhöht zu werden.

Dort aber wird verlangt, dass die 1 bereits im Schnellspeicher steht. Wollten wir in der nun folgenden Programmzeile 19 einfach den Sprung $\rightarrow$ 5 befehlen, so wäre das sicherlich falsch, weil der Schnellspeicherinhalt seit Programmzeile 15 aus dem Index i besteht und nicht, wie in Programmzeile 5 gefordert, gleich 1 ist.
Wir müssen vor dem Erreichen der Zeile 5 dafür sorgen, dass der Schnellspeicherinhalt wieder aus der Zahl 1 besteht. Das erreichen wir durch das Lesen der 1 und durch Sprung in die Zeile 4. In dieser Zeile wird dann die eben gelesene Zahl 1 in den Schnellspeicher überführt.

Nr.	Operation	Adresse	RR	SS
...	...	...	...	...
19	L	301	1	i
20	$\rightarrow$	4	1	i

Gelangt der Automat im Verlauf der Abarbeitung des Programms an die Zeile 20, so »springt« er in jedem Falle, wie es der Befehl $\rightarrow$ 4 vorschreibt, in die Programmzeile 4. Von Zeile 21 an ist also Raum für den unkomplizierten Schlussteil des Programms.

Nr.	Operation	Adresse	RR	SS
...	...	...	...	...
21	L	304	$\sum$	i
22	D		$\sum$	i
23	H		$\sum$	i

In der großen Freude, das Programm mit viel Geduld und Mühe beendet zu haben, vergessen wir leicht, dass eine Adresse offengeblieben ist. Wir erinnern uns, dass der bedingte Sprung der Zeile 13 in den Schlussteil des Programms führen sollte.

Daher muss in der Zeile die Adresse 21 im Befehl N$\rightarrow$ nachgetragen werden. Dann ist aber das Programm endgültig fertiggestellt.
Während die Locherinnen das Produkt unserer Arbeit nach dem Maschinenkode in Lochkarten oder Lochstreifen umsetzen, haben wir einige Zeit, die soeben beendete Tätigkeit noch einmal zu überdenken.

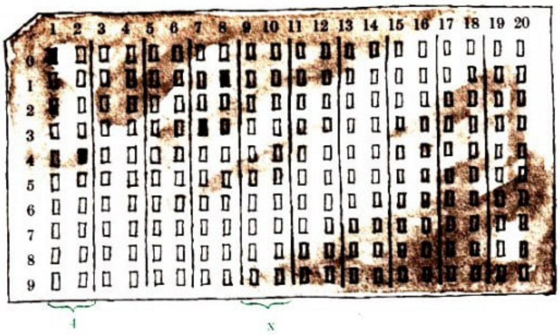
Das Programmieren in der Maschinensprache ist eine formale Übersetzung eines Flussdiagramms in die Befehlssprache des Automaten; es ist oft langwierig und mühselig, erfordert viel Geduld, und jede Ablenkung hat fast immer Fehler zur Folge.

Diese Tatsachen erwecken zwangsläufig den Gedanken, eine solche Tätigkeit dem Automaten selbst zu überlassen. So nimmt es nicht wunder, dass die Bemühungen, den Menschen vom Zwang zu befreien, in Maschinensprachen zu sprechen, letzten Endes genau so alt sind wie die Rechenautomaten selbst.
Vorerst müssen wir aber diesen Gedankengang unterbrechen und uns den Lochkarten zuwenden, auf die das Programm inzwischen übertragen worden sei.

8.4 Das Kartenlesen

Die Lochkarte des GR-01 soll dieselbe Gestalt haben, wie wir sie von der Betrachtung der Lochkartenmaschinen kennen. Für die Darstellung eines alpha-numerischen Zeichens wollen wir je 2 Spalten der Karte vorsehen, wobei wir nach der Verschlüsselung von Seite 110 jeder Spalte der Lochkarte den Inhalt einer Tetrade zuordnen.

Beispielsweise war die Ziffer 4 als 0000 0100 verschlüsselt. Wollen wir nun das Zeichen 4 in die Spalte 1 und 2 der Lochkarte bringen, so müssen wir in der Spalte 1 die 0 und in der Spalte 2 die 4 lochen.



Der Schlüssel des Zeichens x war 00LL 000L. Demzufolge ist in der Lochkarte in der Spalte 7 die 3 und in der Spalte 8 die 1 gelocht.

Für das Lochen des Programms, d. h. für die Übertragung der einzelnen Befehle, wählen wir eine andere Aufteilung der Lochkarte.
Wir wollen auf einer Karte 2 Befehle unterbringen. Der erste Befehl soll in den Spalten 1 bis 10 und der zweite Befehl in den Spalten 11 bis 20 gelocht werden. Für die Angabe der Adresse sollen 4 Spalten und für das Lochen der Operation 2 Spalten vorgesehen werden. Die restlichen Spalten reservieren wir für gewisse Kennzeichen, mit denen wir die Befehle zusätzlich markieren können. Auf der abgebildeten Lochkarte sind die ersten beiden Befehle unseres Programms dargestellt.

Alle Lochkarten müssen, bevor sie in den Automaten gelangen, geprüft werden. Das kann erfolgen, indem man Karte und Programmformular mit dem Auge vergleicht, wobei man den Maschinenkode weiß oder sichtbar vor sich liegen hat.
Sinnvoller ist eine Prüfung durch ein spezielles Prüfgerät, das in der Regel so arbeitet, dass nach der Eingabe der bereits gelochten Karte eine andere Locherin das entsprechende Programmstück noch einmal vom Programmierungsblatt ablocht.

Dabei werden aber keine Löcher gestanzt. Es findet lediglich eine Kontrolle der Übereinstimmung zwischen dem Anschlag der gedrückten Taste und der bereits vorliegenden

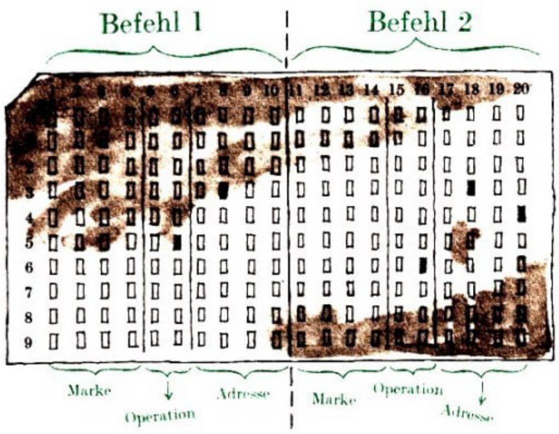
Lochung statt. Stellt das Kartenprüfgerät Nichtübereinstimmung fest, so wird der Fehler sofort angezeigt.

Die Bedeutung der Kontrolle ist außerordentlich groß, da der Automat kritiklos alle Anweisungen ausführt, die er entziffert. Er kann nicht wissen, dass die Adresse 312 gemeint ist, wenn irrtümlich 311 gelocht ist. Ebensowenig kann er ausrichten, wenn sich der Programmierer geirrt hat. Man ist daher gezwungen, bei neu angefertigten Programmen Proberechnungen durchzuführen oder zumindest die ersten Ergebnisse sehr kritisch zu betrachten.

An dieser Stelle könnte jemand die provozierende Frage stellen:
»Kann etwa nur das berechnet werden, was bekannt ist?« Jeder Bearbeiter einer Aufgabe hat ein gewisses Gefühl für die zahlenmäßige Größe der erwarteten Ergebnisse. Das beruht auf Erfahrung oder auf der Kenntnis gewisser leichter übersehbarer Spezialfälle der Aufgabe. In diesem Sinne ist ihm wirklich das Ergebnis vorher bekannt. Das bedeutet natürlich nicht, dass er die Lösung so genau angeben kann, wie sie der Automat ausdruckt.

Wenn andererseits Kontrollrechnungen verlangt werden, so heißt das keineswegs, dass jedes vom Automaten ausgegebene Ergebnis vom Menschen mühsam nachgerechnet werden muss. In der Regel genügt die Durchführung eines einzigen Falles, der alle Besonderheiten der Aufgabe aufweist und alle ihre Bedingungen derart erfüllt, dass er bei der Rechnung auf dem Automaten alle Teile des gespeicherten Programms in Anspruch nimmt.
Prüft man nur gewisse Sonderfälle, so besteht die Gefahr, dass man Fehler in nicht durchlaufenen Programmteilen übersieht. Gelingt keine allgemeine Proberechnung, so werden mehrere Spezialfälle berechnet.

Auf irgendeine Weise lässt sich aber fast immer eine Probe beibringen, die nach erfolgreicher Durchrechnung auf dem Automaten die Gewähr bietet, dass das geschriebene Programm keine Fehler enthält.



Unser Programm für die hundertmalige Addition der 1 ist so kurz und einfach, dass wir annehmen können, bei seiner Aufstellung keine Fehler gemacht zu haben. Während die Karten ins Eingabegerät laufen, werden sie Zeile für Zeile abgetastet und nacheinander auf den Hauptspeicher gebracht.

An dieser Stelle macht sich eine Bemerkung notwendig, die sich noch einmal mit den Adressen des Programms befasst. In dem von uns betrachteten Beispiel wurde stillschweigend die Verabredung getroffen, dass das Programm mit der Speicherzelle 0 des Hauptspeichers beginnt. Das ist natürlich nicht der allgemeine Fall.

Ähnlich, wie eine Bekanntmachung an verschiedenen Stellen einer Litfaßsäule angebracht werden kann, ohne dass sich der Sinn der Bekanntmachung ändert, kann auch das Programm auf verschiedene Stellen des Speichers geschrieben werden. Während bei der Bekanntmachung der Text unverändert bleibt, kann sich im Programm bei einem solchen »Umzug« sehr wohl einiges ändern.
Wir nehmen als Beispiel ein nicht im einzelnen ausgeführtes Programm, das in der Zeile 3 einen Sprung in die Zeile 1 besitzen soll.

Programmzeile	Befehl
0	$***$
1	$***$
2	$***$
3	$\longrightarrow 1$
4	$***$

Die Zeichen *** stellen irgendwelche Befehle dar, die für diese Untersuchung nicht von Belang sind. Steht dieses Programm, mit der Zeile 0 beginnend, auf dem Hauptspeicher, so liegt Übereinstimmung zwischen den Zeilennummern und den Speicherzellnummern vor, und der Befehlsablauf vollzieht sich in der geplanten Weise, so dass die von 0 an nummerierten Programmzeilen ab Speicherplatz 0 gespeichert sind.

Speicherzelle	Programmzeile	Befehl
0	0	$***$
1	1	$***$
2	2	$***$
3	3	$\longrightarrow 1$
4	4	$***$

Speichern wir dieses Programm an eine andere Stelle des Kernspeichers, etwa ab Zelle 300, so ist eine Abarbeitung im gewünschten Sinne nicht mehr möglich, weil der auf der Zelle 303 stehende Sprung nach 1 nicht, wie beabsichtigt, in die Zeile 1 des Programms führt, sondern den absoluten Speicherplatz mit der Nummer 1 berührt.

Speicherzelle	Programmzeile	Befehl	
300	0	$***$	
301	1	$***$	
302	2	$***$	
303	3	$\longrightarrow 1$	Sprung nach 1
304	4	$***$	

In der Zelle 1 befindet sich aber ein Inhalt, der nichts mit dem betrachteten Programm zu tun hat. Offensichtlich muss der Sprung $\rightarrow$ 1 in den Befehl $\rightarrow$ 301 umgeändert werden.

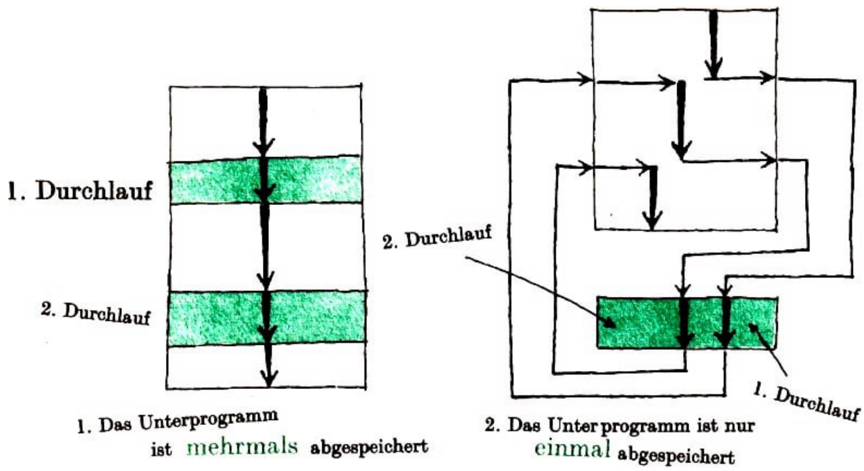
Wie man sieht, setzt sich die Adresse 301 aus der »Leitadresse« 300 und der Nummer der Programmzeile 1 zusammen. Die letzte Adresse heißt auch relative Adresse. Die Leitadresse ist die Nummer der Speicherzelle, in der der erste Befehl eines Programms Platz findet.

In der überwiegenden Mehrzahl der Fälle werden die Programme mit relativen Adressen geschrieben, d. h., es wird mit der Programmzeilennummer 0 begonnen.
Die benötigen absoluten Adressen bildet dann die Rechenmaschine selbst, indem sie zu allen vom Programmierer hervorgehoben relativen Adressen die jeweilige Leitadresse dazuzählt und die Befehle in dieser abgeänderten Form endgültig auf den Hauptspeicher bringt.
Für unsere gedachte Rechenmaschine GR-01 besteht beispielsweise die Möglichkeit, die auf der Lochkarte bisher leergebliebenen Spalten zur Kennzeichnung der relativen Adressen zu verwenden.

8.5 Vereinfacht und verbessert

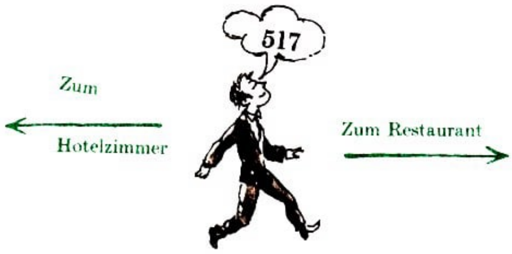
Nach der Erklärung der wichtigsten Tatsachen, die für das Programmieren notwendig sind, soll der Gedanke weiter verfolgt werden, wie diese Tätigkeit vereinfacht und verbessert werden kann.

Besteht die Notwendigkeit, im Verlauf einer Rechnung auf Werte gewisser häufig vorkommender elementarer Funktionen, wie etwa der Funktion $\sin x$, zurückzugreifen, so müssen diese Funktionen an der betreffenden Stelle berechnet werden. Das setzt natürlich ein Programm voraus, das diese Berechnung ermöglicht.
Wenn man nun an mehreren Stellen eines gewissen Programms verschiedene Werte von $\sin x$ benötigt, wäre es unklug, mehrere Exemplare des gleichlautenden Programms für die Berechnung von $\sin x$ zu verwenden und sie an den entsprechenden Stellen des ursprünglichen Programms unterzubringen, wie das im linken Teil der folgenden Abbildung gezeigt ist.



Es reicht natürlich aus, wenn das Programm für $\sin x$ nur in einem Exemplar auf dem Hauptspeicher vorhanden ist. Wir müssen nur Sorge tragen, dass dieses »Unter«-Programm von verschiedenen Stellen des »Haupt«-Programms angesprungen werden kann und dass die jeweilige Absprungadresse festgehalten wird, damit nach dem Durchlauf des Unterprogramms bekannt ist, wohin der Rücksprung zu erfolgen hat. Dies ist im rechten Teil der Abbildung gezeigt.

Man kann zum Vergleich mit diesem Vorgang einen Hotelgast nehmen, der beabsichtigt, ins Restaurant zu gehen und nach dem Essen wieder in sein Zimmer zurückzukehren. Dieser Hotelgast muss dabei seine Zimmernummer im Gedächtnis behalten.



Dieser Vergleich ist aber ungenau: Nach dem Durchlauf des Unterprogramms kann der Rücksprung nicht in dieselbe Speicherzelle erfolgen, von der aus die Ansteuerung des Unterprogramms erfolgt ist, da sich dann ein Zyklus ohne Ende ergäbe. Der Rücksprung findet stets in eine auf die Absprungstelle folgende Zeile statt. Das kann in den meisten Fällen die unmittelbar nächste Zeile sein.

Die Unterprogrammtechnik ist ein erstes wirksames Hilfsmittel des Programmierers. Jeder Rechenautomat besitzt eine Bibliothek von Unterprogrammen, auf die im Bedarfsfall zurückgegriffen werden kann. Dennoch bleibt genügend zu tun, um die Rahmenprogramme für die jeweils interessierenden Probleme aufzustellen.
Mit der Einführung der Unterprogrammtechnik haben wir erst einen kleinen Schritt auf dem Wege zur Vereinfachung der Programmierungsarbeiten zurückgelegt.

Ein weiterer Schritt wird mit der Anwendung sogenannter Kompilierprogramme getan. Das sind Programme, die das Programmieren mit symbolischen Adressen erlauben. Wenn wir uns an das Programm für die hundertfache Addition der Zahl 1 erinnern, so folgten wir dort der Vorschrift, alle Zeilen zu nummerieren und zu allen Befehlen, die sich auf den Hauptspeicher bezogen, konkrete Adressen anzugeben.
Beim Schreiben von Programmen, die anschließend kompiliert werden sollen, entfällt dieser Zwang. Beim Sprungbefehl beispielsweise muss dann nicht mehr die konkrete Adresse der Zelle angegeben werden, in die der Sprung führen soll. Es genügt, die anzuspringende Zelle mit einer Markierung zu versehen und diese Markierung auch als symbolische Adresse in den Sprungbefehl einzufügen.

Ein kleines Beispiel soll den Vorgang veranschaulichen. In einem Programm trete in der Zeile 203 der Sprung in die Zeile 517 auf.

Programmzeile	Befehl
0	* * *
1	* * *
⋮	⋮
203	$\to$ 517
⋮	⋮
517	H

Schreiben wir dieses Programm unter der Voraussetzung, dass es kompiliert werden kann, so entfällt die Notwendigkeit, die Programmzeilen durchgehend zu nummerieren. Lediglich in der früher mit 517 bezeichneten Zeile setzen wir eine gewisse Kennmarke, die aus einer willkürlich gewählten Zahl M bestehen soll. Diese Zahl kann bei der Rechenmaschine GR-01 in die links von der eigentlichen Operationsbezeichnung stehenden Spalten gelocht werden. Wir müssen natürlich die Marke M auch als Adresse im Sprungbefehl angeben:

Marke	Befehl
	$* * *$
	$* * *$
	$\vdots$
	$\to$ M
	$\vdots$
M	H

Nach der Eingabe eines derart geschriebenen Programms setzt die eigentliche Tätigkeit des Kompilierens ein: Die Markierung M im Sprungbefehl wird durch eine konkrete Adresse ersetzt. Der Vorgang selbst ist leichtverständlich.
Ausgehend von der Markierung im Sprungbefehl, wird Zeile für Zeile des Programms auf das Vorliegen derselben Markierung durchmustert.

In einem Register laufen die konkreten Adressen der Zeilen auf. Ist die gesuchte Markierung gefunden, so steht im Register die dazugehörige Speicheradresse bereit; diese wird anschließend in den Sprungbefehl eingesetzt, so dass man am Ende wieder ein Programm im direkten Maschinenkode auf dem Hauptspeicher erhält. Die eigentliche Rechnung erfolgt schließlich nach diesem Programm.

Das Kompilieren ähnelt dem Bezeichnen von Häusern durch Wappen oder Tierköpfe und der nachträglich erfolgten Vergabe von Straßennamen und Hausnummern oder, was dasselbe ist, der genauen Adresse.

8.6 Eine neue Sprache

Auch beim Kompilieren sind wir noch an die spezielle Rechenmaschine gebunden, für die wir das Problem aufbereiten, wenn das Programm auch nicht mehr in allen seinen Teilen im direkten Maschinenkode geschrieben wird.

Die höchste Stufe des Programmierens erreichen wir in dem Augenblick, in dem wir uns völlig von der Rechenmaschine lösen und die Aufgabe in einer Sprache formulieren, die hauptsächlich auf eine bequeme Darstellung des Problems orientiert ist, wobei die Herstellung des Maschinenprogramms in einem Übersetzungsvorgang vom Rechenautomaten selbst vorgenommen wird.
Wir sprechen im folgenden also von problemorientierten Sprachen im Gegensatz zu maschinenorientierten Sprachen.

Da das auf einer programmgesteuerten Rechenmaschine zu lösende Problem in algorithmischer Form vorliegen muss, handelt es sich bei diesen allgemeinen Sprachen um

Darstellungsarten von Algorithmen oder, wie man sagt, um algorithmische Sprachen. Die bekannteste algorithmische Sprache ist Algol.
Der an einen Stern erinnernde Name ist lediglich die Abkürzung des englischen Wortes »Algorithmic Language«, was nichts weiter als »algorithmische Sprache« bedeutet.

Die Sprache Algol entstand im Jahre 1960 nach mehrjähriger Arbeit von Fachleuten. Sie besitzt wie jede Sprache ein System von Regeln, nach denen die Einzelwörter zu Sätzen zusammengefügt werden. Die Bedeutung der Wörter erlernt man so, wie man die Bedeutung der Wörter einer gewöhnlichen Fremdsprache erlernt, durch Erklärung in der bekannten Sprache:

Das Alphabet von Algol hat eine stattliche Länge. Es besteht aus allen lateinischen Buchstaben, den Dezimalziffern, den Operationszeichen für die Grundrechenarten, den Symbolen für die logischen Operationen und arithmetischen Vergleiche und aus einigen Sonderzeichen.
Das wichtigste Sonderzeichen setzt sich aus einem Doppelpunkt und dem Gleichheitszeichen zusammen und soll inhaltlich dasselbe darstellen wie das »Ergibtzeichen« der Flussbildtechnik. Liegt z. B. der Ausdruck

$$a := 1 + b + c$$

vor, so ist er am besten folgendermaßen zu lesen: Der Wert für a ergibt sich gemäß der rechts vom Gleichheitszeichen stehenden Vorschrift, d. h. als Summe der Zahlen 1, b und c.
Man beachte aber, dass sich hier die Leserichtung gegenüber dem Ergibtzeichen der Flussbildsprache geändert hat.

Die Sprache Algol enthält weiter eine nicht sehr große Anzahl von Wörtern aus der englischen Sprache, denen eine ganz bestimmte neue Bedeutung beigelegt wird. Wir wollen einige wenige Wörter, die in der Schreibweise von Algol durch Unterstreichen oder durch besonderen Druck kenntlich gemacht werden müssen, etwas näher ansehen.

In vielen Algorithmen tritt die Situation auf, dass eine Größe x, beginnend mit einem Wert $x = a$, schrittweise solange um einen gewissen Betrag z zu erhöhen ist, bis der Wert $x = b$ erreicht ist.
Zur Charakterisierung dieses Sachverhalts werden in der Sprache Algol die englischen Wörter for, step und until verwendet, denen im Deutschen die Wörter »für«, »Schritt« und »bis« entsprechen. Die folgende Anordnung

$$\text{for } x := a \text{ step } z \text{ until } b$$

fordert also die schrittweise Erhöhung einer Größe x um den Wert z, beginnend bei $x = a$ und endend bei $x = b$.

In der Niederschrift eines Algorithmus treffen wir oft auf Anweisungen, die vorschreiben, welche Anweisung als nächste abzuarbeiten ist. Solche Sprunganweisungen werden in der Sprache Algol mit dem Wörtchen goto gekennzeichnet, dem im Deutschen der Ausdruck »gehe zu« entspricht.

Hinter goto wird dann noch die Markierung der betreffenden Anweisung gesetzt, zu der laut Vorschrift übergegangen werden soll. Zur Markierung kann man beliebige Zahlen oder Buchstaben verwenden.

Im Flussdiagramm lernten wir das Fragekästchen kennen, das in Beantwortung einer Ja-Nein-Frage zwei mögliche Ausgänge besitzt.
In der Sprache Algol werden zur Formulierung solcher Entscheidungen die Wörter if, zu deutsch »wenn«, then (dann) und else (sonst) verwendet. So lesen wir z. B. aus der Zeile

$$\text{if } i = 100 \text{ then goto } B \text{ else } i := i + 1$$

die Tatsache, dass ein Algorithmus in dem Fall, dass ein Zählindex i den Wert 100 annimmt, mit der markierten Zeile B seine Fortsetzung findet, während dann, wenn i nicht gleich 100 ist, dieser Index i um 1 erhöht werden muss.

Des weiteren wollen wir noch das Wörtchen stop kennenlernen, das dieselbe Bedeutung wie der Programmbefehl »Halt« besitzt, und kurz anmerken, dass in der Sprache Algol zwei Sätze stets durch ein Semikolon getrennt werden.
Auf eine Erklärung weiterer Algolwörter sowie auf die Beschreibung der Regeln, die dem Satzbau zugrunde liegen, müssen wir hier verzichten; statt dessen wollen wir den Algorithmus für das Problem, die Zahl 1 hundertmal zu addieren, in der Sprache Algol formulieren:

```
begin
Summe := 0;
i := 1;
A: Summe := Summe +1;
if i = 100 then goto B else i:=i+ 1;
    goto A;
B: stop
    end
```

Das am Anfang stehende Wort begin und das letzte Wort end sind die Zeichen für den Beginn und das Ende der Formulierung eines Algorithmus. Sie können natürlich auch inmitten eines Programms stehen und dort eine Reihe hintereinander auszuführender Anweisungen einschließen.

Für fast alle elektronischen Rechenmaschinen existieren Übersetzungsprogramme, die ein in Algol geschriebenes Problem in die konkrete Maschinensprache übertragen.
Darin liegt die große Bedeutung dieser neuen Sprache. Der Programmierer ist nicht mehr gezwungen, die verschiedenen speziellen Maschinensprachen zu erlernen. Er formuliert die Aufgaben in einer ihm bequemen Form, wobei die Zeit, die er hierbei braucht, wesentlich kleiner ist als diejenige, die er für das Programmieren in der Maschinensprache benötigen würde. Programme, die in Algol geschrieben sind, werden nun vollkommen austauschbar.
Sie können vom Menschen und vom Automaten gleichgut gelesen werden. Die Zahl der Vorteile, die bei der Verwendung von Algol entstehen, ist so groß, dass man unwillkürlich

fragt, ob es denn gar keinen Nachteil gibt.

Jeder kennt die goldene Regel der Mechanik: Was man an Kraft spart, muss man an Weg zusetzen.
Für die Programmierung scheint eine ähnliche Regel zu gelten: Was man an Programmierzeit spart, muss man an Rechenzeit zusetzen.

Es hat sich herausgestellt, dass maschinell aus der Sprache Algol übersetzte Programme wesentlich mehr Speicherplatz beanspruchen als Programme, die unmittelbar in der Maschinensprache geschrieben werden und auch längere Rechenzeiten zur Folge haben. Bedenkt man aber, dass die Entwicklung zu immer schnelleren und größeren Automaten führt, so ist dieser Schönheitsfehler zu verschmerzen.

Eine andere Gesetzmäßigkeit, die man beim Vergleich verschiedener Programmiersprachen erkannt hat, besitzt dagegen mehr Gewicht. Je »niveauvoller« eine Sprache ist, desto »niveauvoller«, d. h. ernsthafter werden die Fehler, die man in ihr begehen kann. Beim Programmieren in der Maschinensprache unterlaufen dem Bearbeiter viele Flüchtigkeitsfehler, die sich z. B. auf das Vergessen oder Vertauschen von Adressen oder Befehlen beziehen. Ernsthafte Fehler, die an den Grundaufbau des Algorithmus rühren, sind dagegen selten.

In Algol geschriebene Programme zeigen nun die entgegengesetzten Tendenzen. Es treten in verstärktem Maße Fehler auf, die darin bestehen, dass die Aufgabe in ihrem Sinngehalt nicht richtig geschrieben ist.
Hier aber hilft die Tatsache, dass zum Schreiben von Algolprogrammen wenig Zeit benötigt wird. Erweist sich ein solches Programm als falsch, so wird es mit größerer Aufmerksamkeit neu geschrieben und nach kurzer Zeit wieder auf der Maschine erprobt.

8.7 Eine ganze Sprachfamilie

Beim Erlernen einer künstlichen, von Fachleuten geschaffenen Sprache fragt man sich, ob diese Sprache wirklich notwendig ist. Liegt hier vielleicht eine ähnliche Situation vor, wie wir sie von der Weltsprache Esperanto kennen?
Dieser Vergleich ist durchaus berechtigt. Es gibt aber einen anderen, der ebenso erwähnenswert ist.

Noch in unserem Jahrhundert traten wiederholt Fälle auf, in denen von Sprachwissenschaftlern Schriftsprachen für Völker geschaffen wurden, die bis dahin keine solche

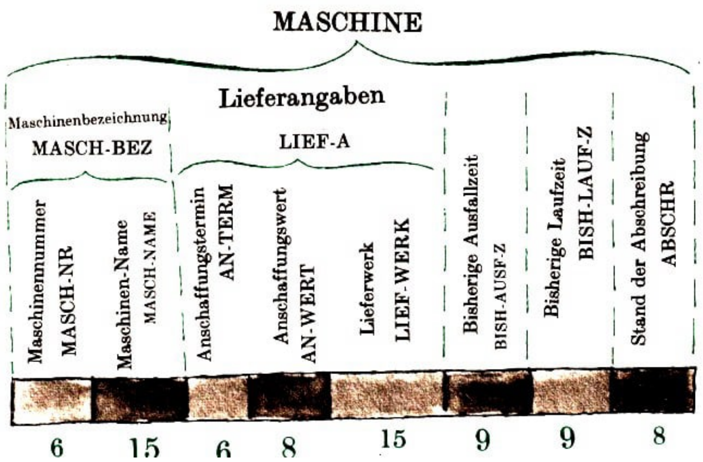
Sprachen kannten. Das bedeutet natürlich nicht, dass diese Völker vordem nicht gesprochen und nicht geschrieben hätten. Es fehlte lediglich eine einheitliche, auf festen Normen basierende Schriftsprache.
Man könnte sagen, dass mit Algol eine einheitliche Sprache für das Volk der Rechenmaschinen und Programmierer geschaffen worden ist. Es ist nur schade, dass es nicht bei dieser einen Sprache bleiben konnte.

Sehr bald erkannte man, dass die Sprache Algol zur Formulierung ökonomischer Probleme, insbesondere bei der Verarbeitung großer Datenmengen, die in unterschiedlicher Form auf verschiedenen Medien eingegeben und gespeichert werden, nicht besonders geeignet war. Aus diesem Grunde entwickelte man eine spezielle Sprache für die Behandlung solcher Probleme und nannte sie COBOL.
Dieser Name ist wieder eine Abkürzung und bedeutet sehr frei übersetzt so viel wie algorithmische Sprache für Aufgaben aus Wirtschaft und Ökonomie.

Die Sprache COBOL unterscheidet sich von der Sprache Algol vor allem in der Beschreibung der Ein- und Ausgabevorgänge bei elektronischen Datenverarbeitungsanlagen. Sie misst diesen Vorgängen größere Bedeutung zu und benutzt vor allem den für die Behandlung ökonomischer Probleme wichtigen Begriff des Datenbestandes.
Unter einem Datenbestand wollen wir eine nach bestimmten fachlichen Gesichtspunkten angelegte Sammlung von Informationen verstehen, die im allgemeinen einen großen Umfang hat.

Vorgegeben sei beispielsweise ein Datenbestand mit dem Namen MASCHINENVERZEICHNIS, der aus einer großen Anzahl von Datensätzen besteht. Den Aufbau eines solchen Datensatzes zeigt unser Bild auf der nächsten Seite.

Dieser Datensatz heiße MASCHINE. Es soll dabei nicht weiter interessieren, auf welchem konkreten Speichermedium der Bestand organisiert ist.
Die Angaben MASCH-NR (Maschinennummer) und MASCH-NAME (Maschinenname) sollen unter dem übergeordneten Namen MASCH-BEZ (Maschinenbezeichnung) zusammengefasst werden.
Dasselbe gelte für die Angaben ANTERM (Anschaffungstermin), AN-WERT (Anschaffungswert) und LIEF-WERK (Lieferwerk), die unter der gemeinsamen Bezeichnung LIEF-A (Lieferangaben) geführt werden sollen.



Die mit Großbuchstaben geschriebenen Bezeichnungen sind die für die Sprache COBOL typischen sogenannten Datennamen. Alle konkret gespeicherten Werte werden durch solche Datennamen aufgerufen.

In einem speziellen Teil eines COBOL-Programms, der »DATA-DIVISION«, wird der Aufbau der benutzten Datenbestände auf eine sehr charakteristische Art beschrieben, was im folgenden kurz gezeigt werden soll. Es werden sogenannte Stufennummern verwendet. Dabei erhalten der »ranghöchste« Datenname die Stufennummer 01 und alle auf derselben nächstniederen Teilstufe gelegenen Datennamen die Nummer 02.
Wird ein Begriff dieser Stufe noch weiter geteilt, so erhalten die Teilbegriffe die nächste Stufennummer 03. Dieses Verfahren wird bis zur niedrigsten Stufe fortgesetzt. Für unser Beispiel ergibt sich folgendes Bild:

01 MASCHINE
02 MASCH-BEZ
03 MASCH-NR
03 MASCH-NAME
02 LIEF-A
03 AN-TERM
03 AN-WERT
03 LIEF-WERK
02 BISH-AUSF-Z
02 BISH-LAUF-Z
02 ABSCHR

Die Angaben der zugehörigen Zeichenzahlen erfolgt nun hinter denjenigen Datennamen, die nicht weiter gegliederte Begriffe bezeichnen. Dazu werden die beiden COBOL-Wörter PICTURE IS (das Bild ist) verwendet. Gleichzeitig wird die Verabredung getroffen, dass unter dem Symbol A ein Buchstabe und unter dem Zeichen 9 eine Ziffer verstanden werden soll und dass hinter diesem Symbol vermerkt sein soll, wie oft die durch dieses Symbol vertretene Zeichenart vorkommt. Die Maschinennummer ist beispielsweise 6stellig und besteht nur aus Ziffern. Wir notieren daher

03 MASCH-NR PICTURE IS 9(6).

Der Maschinenname ist dagegen eine Bezeichnung, die aus Buchstaben besteht. Aus diesem Grunde schreibt man

03 MASCH-NAME PICTURE IS A(15).

In einem weiteren Teil des COBOL-Programms, in der sogenannten PROCEDURE DIVISION, werden dann die eigentlichen Prozeduren aufgeschrieben, die mit den in der DATA-DIVISION erklärten Datenbeständen durchgeführt werden sollen. Einige Anweisungen ähneln dabei durchaus den Befehlen der Sprache Algol; man kann verallgemeinernd sagen, dass die Kenntnis einer bestimmten problemorientierten Sprache das Erlernen einer weiteren wesentlich erleichtert.

Neben diesen beiden bereits »klassischen« problemorientierten Sprachen Algol und COBOL entstanden eine Reihe weiterer Sprachen für die Formulierung algorithmisch fassbarer Probleme.
Darunter befinden sich einige Versuche, die Vorteile der beiden Sprachen Algol und COBOL in einer einzigen algorithmischen Sprache zu vereinigen.

Unter diesen Versuchen muss man die Sprache ALGEK erwähnen, die von sowjetischen Wissenschaftlern in den Instituten der Automatenfamilie URAL entwickelt wurde, sowie die vor allem in den IBM-Laboratorien entstandene Sprache PL-1.
Neben den problemorientierten Sprachen für die Behandlung technischer oder ökonomischer Probleme auf Rechenautomaten müssen unbedingt noch die Sprachen zur Herstellung von Programmen für numerisch gesteuerte Werkzeugmaschinen erwähnt werden, weil ihnen bei der unmittelbaren Vorbereitung der Produktion und bei der Anwendung moderner Technologien in der Industrie große Bedeutung zukommt. Bei der Verwendung solcher Maschinen und der Programmierung der Arbeiten in der Symbolsprache SYMAP erreicht man Produktivitätssteigerungen von mehreren hundert Prozent.

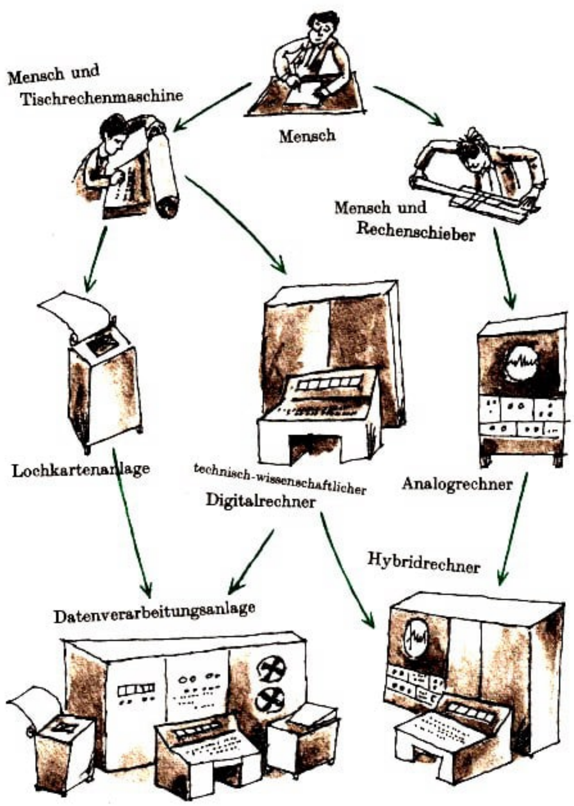
Das Gesagte deutet bereits darauf hin, wie wichtig es ist, in der Informationsverarbeitungstechnik nicht nur schlechthin die »klassischen« Rechenautomaten zu nutzen, sondern auch alle Möglichkeiten auszuschöpfen, die sich aus der Existenz programmgesteuerter Automaten ergeben.

9 Alltag und Sonntag von Elektronenhirnen

9.1 Der Stammbaum der Elektronenhirne

Mit der Erklärung des Programmierens haben wir eines der schwierigsten Kapitel über digitale Rechenautomaten hinter uns gelassen. Im folgenden sei versucht, die große Vielfalt ihrer Anwendungsmöglichkeiten zu skizzieren.
Das bisher über diese Maschine Gesagte bezog sich hauptsächlich auf den Typ des universellen Automaten, d. h. auf einen Automaten, der zu keinem speziellen Zweck gebaut wurde, der also die Bearbeitung beliebiger algorithmischer Aufgaben zuließ. Neben diesem universellen Rechner gibt es viele spezielle Typen von elektronischen Automaten, die zur Lösung verschiedener besonderer Probleme konstruiert wurden.

Nachfolgend finden wir eine Darstellung, die den wichtigsten Grundtypen von Rechenmaschinen einen Platz in der bisherigen Entwicklungsgeschichte programmgesteuerter Automaten zuweist.



Während der mit Papier und Bleistift rechnende Mensch gezwungen war, neben dem Programm auch noch die Algorithmen der Rechenoperationen im Gedächtnis zu haben, sind diese Algorithmen in den Systemen Mensch-Tischrechenmaschine und Mensch-Rechenschieber bereits mechanisiert.
In der Lochkartenanlage kommen die Mechanisierung der Ein- und Ausgabe und eine minimale Programmierbarkeit hinzu. Auf dem technisch-wissenschaftlichen Digitalrechner können Algorithmen numerischer Rechnungen in großem Maße gespeichert werden.

Der Analogrechner löst Aufgaben, die als Differentialgleichungen formuliert werden können. In der Datenverarbeitungsanlage vereinen sich die Eigenschaften der Lochkartenmaschine, große Zahlenmengen zu verarbeiten, mit der Fähigkeit des ursprünglich entwickelten technisch-wissenschaftlichen Digitalrechners, beliebige Algorithmen abzuarbeiten. Der Hybridrechner stellt schließlich eine Vereinigung des Ziffernrechners mit dem Modellrechner dar.

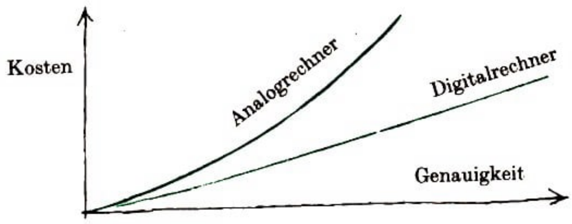
Es existieren natürlich noch weit mehr spezielle programmgesteuerte Automaten. An Bord von Flugzeugen und Raketen befinden sich Kursrechner; die sowjetische Überschallverkehrsmaschine TU 144 wird fast ausschließlich von Computern gesteuert.
Die Medizin verwendet Messgeräte, die die Messwerte speichern und vielfältig umwandeln können. Das Gemeinsame aller dieser Apparaturen ist das Prinzip der Programmsteuerung. Es ist die Fähigkeit dieser Maschinen, algorithmisch gefasste Aufgaben, die auf irgendeinem Medium gespeichert vorliegen, selbständig zu bearbeiten.

Jede Beschreibung der Anwendungsmöglichkeiten muss daher mit einer Skizze der benutzten Maschine verbunden sein.

9.2 Der Streit zwischen zwei Automaten

Betrachten wir zunächst einmal, wie die so gegensätzlichen Tendenzen des analogen und digitalen Rechnens in einem einzigen Automatentyp vereinigt worden sind.

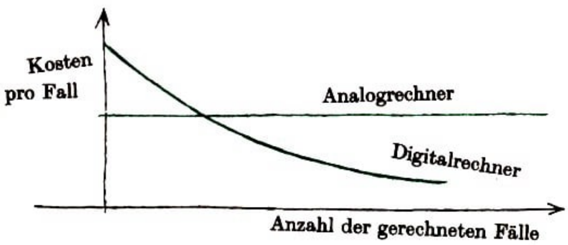
In vielen Fällen erfolgt die Gegenüberstellung der Analog- und Digitaltechnik mit etwas Böswilligkeit. Es wird versucht, Digitalrechner und Analogierechenmaschinen als rivalisierende Parteien hinzustellen. Vergleicht man etwa die Kosten beider Rechengeräte in Abhängigkeit von der erreichbaren Genauigkeit, so zeigt sich, dass die Kosten für einen Analogrechner großer Genauigkeit die Kosten eines Digitalrechners mit derselben Genauigkeit in um so größerem Ausmaß übersteigen, je höher die erzielte Genauigkeit sein soll.



Ein ähnliches Bild erhalten wir, wenn wir die Kosten betrachten, die bei der wiederholten Lösung einer für beide Typen geeigneten Aufgabe entstehen. Hier zeigt sich ebenfalls eine anfängliche Überlegenheit des Analogrechners gegenüber dem Digitalrechner.

Wenn wir aber immer wieder dieselbe prinzipielle Aufgabe lösen, wobei wir lediglich die Eingabewerte ändern, schwindet die Überlegenheit des Analogrechners. Das resultiert aus der einfachen Tatsache, dass neue Eingabewerte beim Digitalrechner lediglich eine Neufüllung der für sie reservierten Speicherplätze erfordern.

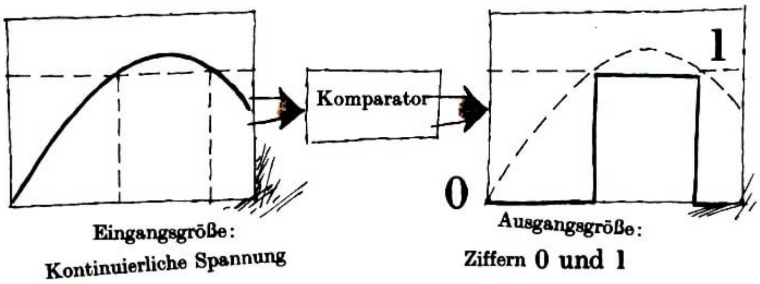
Das kann mit elektronischer Geschwindigkeit erfolgen, sobald alle Eingabewerte für alle zu rechnenden Fälle auf nicht für die Rechnung benötigten Teilen des inneren Speichers zu finden sind. Dagegen ist beim Analogrechner eine Neueinstellung von Schaltern und Drehknöpfen erforderlich, was sicherlich einen größeren Zeitaufwand bedeutet.



Wir wollen hier zwischen beiden Typen von Rechenautomaten keinen Streit heraufbeschwören, sondern wiederholen, dass beide ihre typischen Anwendungsgebiete besitzen. Der Analogrechner findet überwiegend dort seine Anwendung, wo Probleme anfallen, die sich mit Differentialgleichungen beschreiben lassen und keine übermäßigen Forderungen bezüglich Genauigkeit stellen.
Der Ziffernrechner ist dagegen zur Lösung von Aufgaben geeignet, die auf vielen Entscheidungen und komplizierten numerischen Prozeduren beruhen und Ergebnisse großer Genauigkeit verlangen.

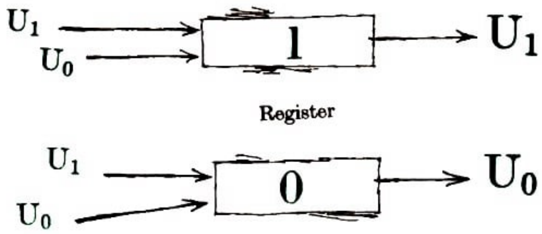
Der Streit zwischen beiden Automatentypen fand mit der Entstehung des Hybridrechners sein Ende. Im Hybridrechner verschmelzen beide Techniken zu einem Automaten neuen Typs. Wir kennen das Wort »hybrid« aus der Biologie, wo es im Zusammenhang mit Kreuzungen verwendet wird.
In diesem Sinne ist der Hybridrechner eine Kreuzung zwischen einem Digitalrechner und einem Analogrechner, die durch Umwandlungskanäle miteinander verbunden sind.

Zu den bereits bekannten Bausteinen der Ziffernrechner und der Analogrechner kommen hier die »hybriden« Grundelemente dazu. Sie erlauben den Übergang von Zahlen zu Spannungen und umgekehrt die Umwandlung kontinuierlich verlaufender Spannungen in ziffernmäßig dargestellte Zahlenwerte.



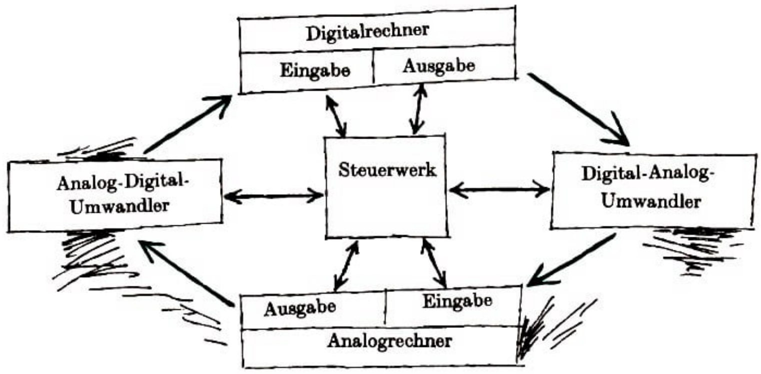
Das Grundelement zur Umwandlung von Spannungen in digitale Impulse ist der Komparator, dessen Prinzip in unserer Skizze erklärt ist. Auf den Eingang des Komparators wirkt eine kontinuierlich verlaufende Spannung. Als Ausgangsgröße erscheint ein normierter Spannungssprung, wenn die Eingangsspannung einen bestimmten Bezugswert über- oder unterschreitet.

Für die Umwandlung von Dualzahlen in Spannungen ist hauptsächlich der elektronische Schalter zuständig. Darunter verstehen wir ein Bauelement, auf dessen Eingang zwei verschiedene Spannungen U_0 und U_1 wirken, während an seinem Ausgang nur jeweils eine dieser beiden Spannungen erscheint.



Den Kern des elektronischen Schalters bildet ein Register für eine Dualziffer. Ist dieses Register mit der Dualzahl 1 gefüllt, so erscheint am Ausgang des Elementes die Spannung U_1.Liegt dagegen die Dualzahl 0 vor, ist also das Register leer, so erhält man am Ausgang die andere Spannung U_0.

Mit diesen beiden Elementen lassen sich vollständige Digital-Analog- und Analog-Digital-Umwandler realisieren. Die Ergebnisse des Analogrechners, die ja als kontinuierliche Spannungen vorliegen, werden durch den Analog-Digital-Umwandler in Zahlen umgeformt, die dem Ziffernrechner eingegeben werden können.
Umgekehrt wandelt der Digital-Analog-Umsetzer die Ergebnisse des Digitalrechners aus der Zahlenform in Spannungen um, die dem Analogrechner als Eingangsinformation dienen.



Für die Programmierung erweisen sich die Hybridrechner als äußerst komplizierte Maschinen. So müssen z. B. alle Zeitfragen berücksichtigt werden, die beim Informationstransport zwischen den beiden Teilen des Rechners auftreten. Viele dabei entstehende Probleme können erst nach einer probeweisen Programmierung geklärt werden.

Das wichtigste Anwendungsgebiet der Hybridrechner sind der Luftverkehr und die Raumfahrt. Bei der Berechnung des Fluges einer Rakete überlässt man die Simulation der Flugdynamik, des Autopiloten und der Regelsysteme dem Analogrechner. Der Digitalrechner übernimmt die Flugbahnberechnung und die Nachahmung der Erdumdrehung.

Eine weniger bekannte Einsatzmöglichkeit für den Hybridrechner ist in der Medizin die Untersuchung von Narkosefragen.
Hier übernimmt der Analogrechner die Aufgabe, die vegetativen Vorgänge darzustellen, was allerdings voraussetzt, dass wenigstens einige dieser Vorgänge und ihr Zusammenwirken mathematisch formuliert werden können. Der Digitalrechner kontrolliert mit

Hilfe der ihm zugehenden Informationen die Vorgänge im »Körper« und berechnet die jeweils erforderliche Menge eines oder mehrerer Narkosemittel.
Nach der Umwandlung dieser Mengenangaben aus der Zahlenform in Spannungen werden sie dem Analogrechner im Sinne einer Regelung als neue Einflussgrößen eingegeben. Solche Untersuchungen eignen sich zum Beispiel zur Erforschung der Narkosetiefe.

Auch die elektronischen Hybridrechner besitzen mechanische Vorläufer. Bereits im 18. Jahrhundert gab es Geräte, die digitalanalogen Charakter hatten.

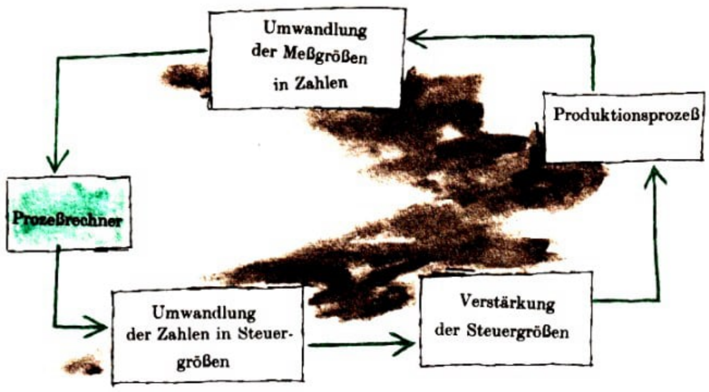
Im Jahre 1756 wurde Friedrich Kraus aus der uns bereits bekannten Uhrmacherfamilie an das physikalische Hofkabinett nach Wien berufen. Hier widmete er sich ganz seiner Lieblingsidee, einen Schreibautomaten zu bauen, der dann auch 1760 fertiggestellt wurde.
Auf einer Erdkugel stand eine nach damaliger Tracht gekleidete Sekretärin und führte eine Feder, die sie nach je 4 Buchstaben in ein Tintenfass tauchte, über ein Stück eingespanntes Papier. Sie konnte Texte bis zu 68 Buchstaben schreiben, deren Folge auf einer Stifttrommel gespeichert war. Jedes Zeichen wurde durch eine bestimmte Anordnung aus der Trommel herausragender Stifte kodiert.

Das Schreiben der Buchstaben hat große Ähnlichkeit mit der Ausgabe von Kurven auf einem modernen Analogrechner. Das Innere des Schreibautomaten entspricht dagegen den Prinzipien eines Digitalrechners. Damit lag hier bereits eine voll funktionierende Umwandlung digitaler Signale in eine kontinuierliche Form vor.

9.3 Der automatische Betriebsingenieur

Ein anderer wichtiger Typ eines Rechenautomaten, bei dem die - gegenseitige Umwandlung digitaler und analoger Größen eine Rolle spielt, ist der sogenannte Prozessrechner.

Seine Aufgabe besteht in der Steuerung und Überwachung von Produktionsprozessen. Er ist vor allem in der chemischen Industrie, im Hüttenwesen und in der Energiewirtschaft verbreitet. Die Arbeitsweise eines Prozessrechners zeigt unser Schema.



Kernstück des Prozessrechners ist ein mittelschneller Digitalrechner, der im Hinblick auf besondere Betriebssicherheit konstruiert wurde. Der wichtigste Unterschied gegenüber den universellen Rechenautomaten besteht beim Prozessrechner in der Organisation der Ein- und Ausgabe.

Die Eingabewerte des Prozessrechners sind Mess- und Kontrollwerte aus dem Produktionsprozess, die mit Hilfe von Analog-Digitalumwandlern in Dualzahlen überführt werden. Auf diese Weise wird ein ständiger Informationsfluss aus dem Betriebsgeschehen in das steuernde Organ, den Prozessrechner, ermöglicht.

Das befähigt den Rechner, nach festen programmierten Entscheidungsregeln die interessierenden Vorgänge ständig zu überwachen, entstehende Gefahren frühzeitig abzuwenden und Qualitätsschwankungen schnell auszugleichen. Der Rechenautomat muss natürlich neben dem kontinuierlichen Messwerteingang einen ebensolchen Ausgang von Befehls- und Steuergrößen haben, die aus einer Umwandlung der Dualzahlen des Automaten hervorgehen und nach einer geeigneten Verstärkung durch verschiedene Hilfsmechanismen direkt auf den Produktionsprozess einwirken können, wo sie Zuflussmengen variieren oder Umdrehungsgeschwindigkeiten ändern.

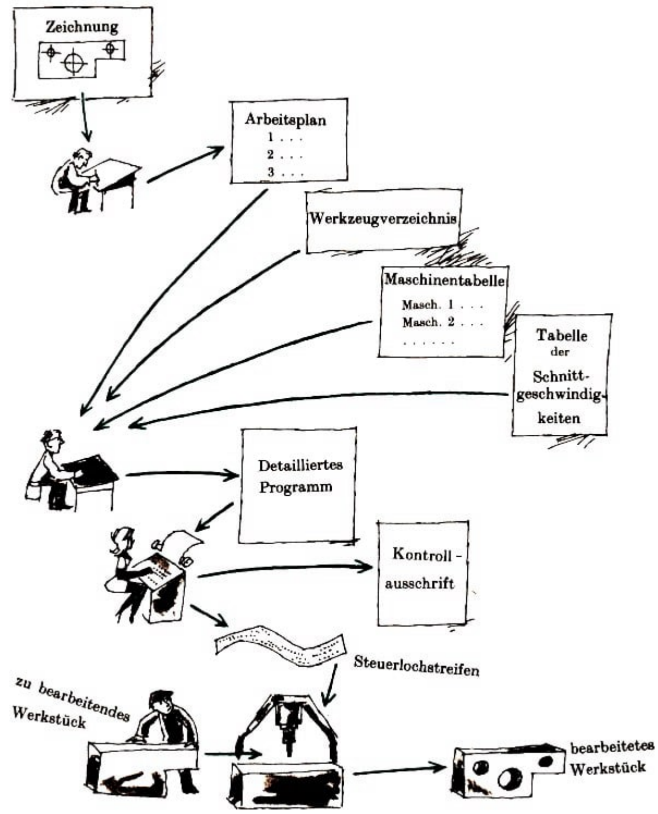
Wir erkennen in diesem Vorgang die Struktur eines einfachen Regelkreises, doch ist das Niveau höher als das der einfachen Regelung.
Das zentrale Regelgerät, der Prozessrechner, fällt hier wesentlich kompliziertere Entscheidungen als beispielsweise ein einfacher Proportionalregler, der nur die Aufgabe hat, bei einer zahlenmäßig bestimmten Abweichung von der Norm in das zu überwachende System einzugreifen.
Ein Prozessrechner übt die Funktion eines Betriebsingenieurs aus, während jetzt dem Ingenieur die Aufgabe gestellt ist, das System Prozessrechner-Produktionsprozess zu überwachen.

9.4 Elektronenhirne mit Muskeln

Während der Prozessrechner gewissermaßen noch neben der Produktion steht und diese aus einiger Entfernung steuert, sind andere digitale Automaten bereits selbst Werkzeuge geworden. Die informationsverarbeitende Anlage verschmolz mit der stoffverarbeitenden Maschine zu einem hochproduktiven Automatisierungsmittel der Gegenwart, der numerisch gesteuerten Werkzeugmaschine.

Auch der Automat für die Großserienfertigung, wie wir ihn heute in den Betrieben finden, ist mit seinem fest eingebauten Arbeitsablaufspeicher eine programmgesteuerte Maschine. Die Folge der Bewegungen und Tätigkeiten der Maschine ist in einem mechanischen System aus Hebeln, Nockenwellen und Zahnrädern gespeichert.
Dieser starre Speicher wird nun bei einer numerisch gesteuerten Maschine durch einen leicht auswechselbaren Programmträger ersetzt. Meist wird der Lochstreifen verwendet, auch das Magnetband ist geeignet. Mit der leichten Auswechselbarkeit des Arbeitsablaufspeichers vergrößert sich der Einsatzbereich der Maschine in einem fast unbeschränkten Ausmaß.

Die einzelnen Zeichen des einlaufenden Lochstreifens werden von einem elektronischen Steuerwerk entschlüsselt und in Steuerfunktionen der Werkzeugmaschine übersetzt. Dabei müssen auch Rechenoperationen ausgeführt werden, wenn beispielsweise die Ermittlung von Bahnkurven für Schnittwerkzeuge nötig ist. Die numerisch gesteuerte Werkzeugmaschine ist damit nichts anderes als eine Werkzeugmaschine mit einem kleinen eingebauten programmgesteuerten Rechenautomaten; es wäre vielleicht besser, sie aus diesem Grunde programmgesteuerte Werkzeugmaschine zu nennen.

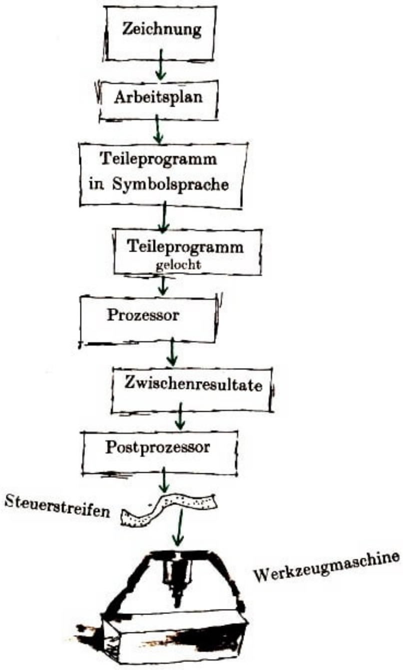


Beim manuellen Programmieren, d. h. beim Programmieren in der speziellen Sprache der Werkzeugmaschine, wird ausgehend von der Konstruktionszeichnung ein Programmblatt ausgefüllt, in das alle Arbeitsgänge der Maschine eingetragen werden, die zur vorgesehenen Bearbeitung des Werkstücks in ihrer zeitlichen Reihenfolge notwendig sind.
In jedem Programmschritt muss ein Teil der oft recht beträchtlichen Zusatzinformationen über Werkstoffeigenschaften, Abmessungen und Stellungen der Werkzeuge, Schnittgeschwindigkeiten und dergleichen in maschinenspezifische Befehle übersetzt werden. Der Text des entstehenden Programms wird dann auf einen Steuerstreifen übertragen, wie das die nebenstehende Zeichnung zeigt.

Mit wachsender Kompliziertheit der zu bearbeitenden Werkstücke und mit Vergrößerung der Zahl der Arbeitsgänge steigt der Zeitaufwand und die Fehleranfälligkeit der manuellen Programmierung beträchtlich. Diese Erscheinung ist uns bereits aus dem Kapitel über die Programmierung der Rechenautomaten bekannt. Dort wurden auch die Symbolsprachen für Werkzeugmaschinen erwähnt.
Bei der Anwendung solcher Sprachen unterscheidet man zwei wichtige Phasen des Übersetzungsprozesses.

Zuerst muss das in einer symbolischen Sprache geschriebene Programm in einen Rechenautomaten eingegeben und von einem Umwandlungsprogramm, dem Prozessor, in eine Folge interner Codes übersetzt werden. Dabei erfolgt u. a. die Ermittlung und

Speicherung standardisierter Normalformen der in der technischen Zeichnung verwendeten geometrischen Elemente.



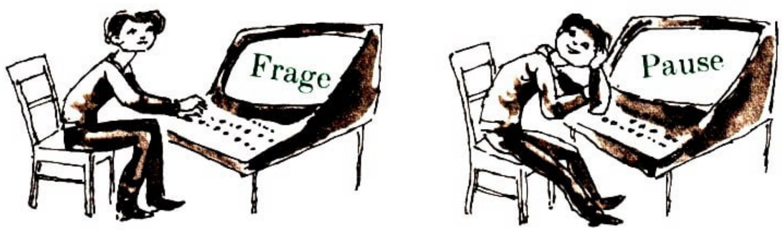
In dieser Phase erfolgt auch die Berechnung des Werkzeugweges längs der vom Konstrukteur vorgegebenen Schnittkurven und Begrenzungsflächen. Als Ergebnis entsteht eine Zeichenfolge, die noch nicht der Werkzeugmaschine angeboten werden kann, da sie keine Informationen über die Sprache dieser Werkzeugmaschinen und über ihre konkreten Arbeitsoperationen enthält.
Erst in der zweiten Phase der Übersetzung erfolgt die Umwandlung der noch allgemeinen Aussagen in die Sprache einer konkreten numerisch gesteuerten Werkzeugmaschine.

Das dafür vorgesehene Anpassungsprogramm (der Postprozessor) muss die spezifischen Informationen über die Werkzeugmaschine wie mögliche Vorschübe, Drehzahlen und Arbeitsstellungen auswerten.
Als Ergebnis gibt der Rechenautomat einen Lochstreifen aus, der in der Werkstatt die Steuerung der Arbeitsmaschine übernimmt.

Wie alle programmgesteuerten Automaten stellen auch die numerisch gesteuerten Werkzeugmaschinen hohe Forderungen an die Qualifizierung des Technologen. Es erfüllt uns deshalb mit Genugtuung, wenn wir in folgendem Beispiel erfahren, dass die modernen Automatisierungseinrichtungen ihren eigenen Beitrag dazu leisten, dem Menschen das Wissen zu vermitteln, das er zur Beherrschung der neuen Technik benötigt.

9.5 Schüler ohne Lehrer

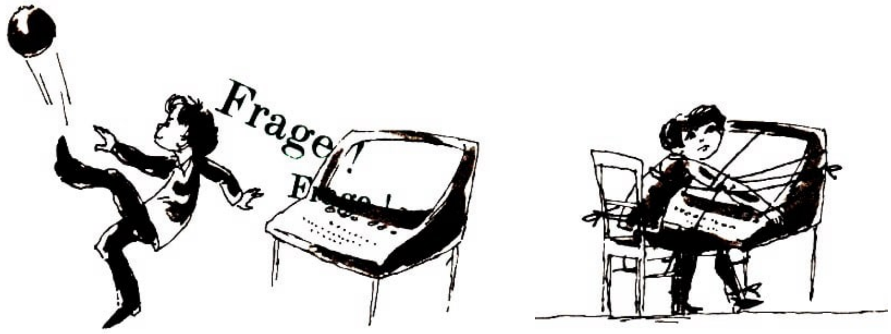
Die Automatisierung und Rationalisierung der Lern- und Lehrmethoden hat bereits eine große Anzahl spezieller kybernetischer
Apparaturen und Einrichtungen hervorgebracht, die unter dem Namen »Lernmaschinen« anzutreffen sind. Sie sollten aber besser Lehrmaschinen genannt werden. Sie verwirklichen die Automatisierung des programmierten Unterrichts, den wir am Anfang dieses Buches kennenlernten.



Der Lernende setzt sich vor diesen Automaten wie vor einen Lehrer und bekommt von ihm entweder als unmittelbar gedruckten Text oder über die Angabe eines bestimmten

Lehrbuchs und der aktuellen Seitenzahlen den zu erlernenden Stoff vorgelegt. Nach einer gewissen Zeit stellt der Automat eine Testfrage, um die Aneignung zu überprüfen.

Hält er die vom Schüler gewöhnlich über Schreibmaschine gegebene Antwort für unzureichend, so verweist er ihn auf ein nochmaliges Durcharbeiten des Stoffes oder stellt ihm zusätzliches Material zur Verfügung.
Aus weiteren Fragen erkennt der Automat den Grad des Fortschritts, entdeckt Ermüdungserscheinungen, verordnet Pausen und stellt sich so ganz auf die Lerngeschwindigkeit seines Schülers ein, wozu ein Lehrer, der eine Klasse von 20 Schülern unterrichtet, nicht in der Lage ist.

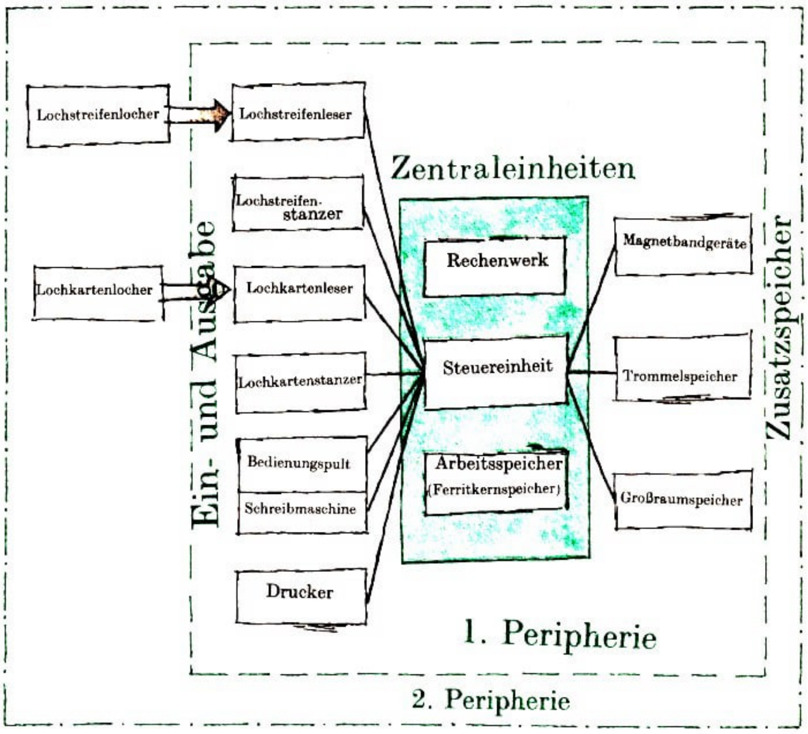


Selbstverständlich wird der Lehrer von der Lehrmaschine nie verdrängt werden. Wenn man will, kann man dies bereits der Tatsache entnehmen, dass die beste Lehrmaschine versagt, wenn sie vergeblich auf die Antwort ihres Schülers warten muss, der ihr einfach davongelaufen ist, um Fußball zu spielen.
Es hieße, den Automatisierungsgrad entschieden zu weit treiben, wollte man diesem Übelstand und dem Schüler mit einem System von Lederriemen und Stricken zu Leibe rücken.

9.6 Der restlos ausgebeutete Automat

Nach einem flüchtigen Blick auf einige spezielle Automaten soll noch einmal auf die bekannteste programmgesteuerte Maschine, die elektronische Datenverarbeitungsanlage, eingegangen werden.
Man kann sie sich aus einem universellen Digitalrechner und aus einer Vielzahl sogenannter Peripheriegeräte aufgebaut denken; also aus Druckern, Lochstreifgeräten, Magnetbandeinrichtung und ähnlichem. Der elektronischen Speicherung von Daten dient nicht nur der innere Hauptspeicher des zentralen Automaten.

Durch eine Reihe äußerer Zusatzspeicher vergrößert sich die gesamte Kapazität in beträchtlichem Maße. Das versetzt die Datenverarbeitungsanlage in den Stand, jederzeit auf eine große Informationsmenge zurückgreifen zu können.
Unsere grobe Prinzipskizze einer EDVA (wie man gern abkürzend für elektronische Datenverarbeitungsanlage sagt) macht deutlich, dass ein so kompliziertes Maschinensystem eine gute Koordinierung der Arbeit der Einzelgeräte erfordert.

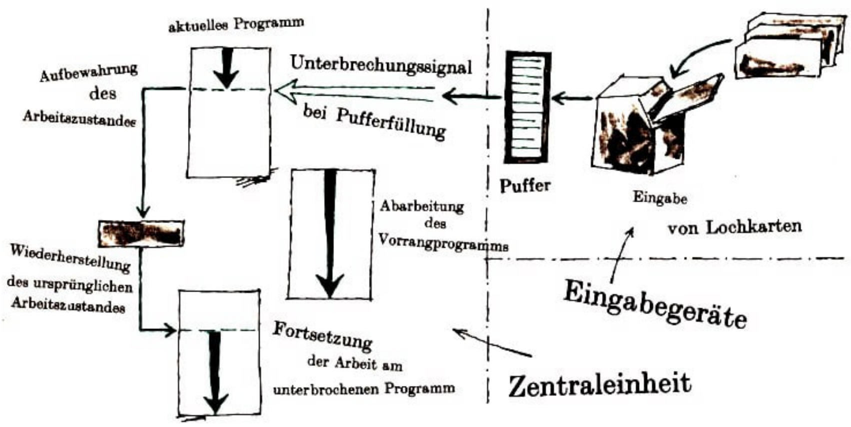


Viele elektronische Datenverarbeitungsanlagen arbeiten deshalb nach dem Prinzip der Vorrangsteuerung. Was ist darunter zu verstehen?
Während des Eingabevorgangs einer auf der Lochkarte befindlichen Zahl vergeht eine relativ lange Zeit, ehe diese Zahl vollständig in ein elektronisches Register eingelaufen ist, um von dort in den Hauptspeicher des Automaten überführt zu werden.

Während des »mechanischen« Teils der Eingabe hat also die innere logische Struktur des Automaten nichts weiter zu tun als zu warten.
Obwohl diese Wartezeiten meist nur wenige tausendstel Minuten betragen, billigt man sie den Rechenmaschinen nicht mehr zu, da sie sich im Verlauf der Gesamtnutzungszeit in Stunden, Tage und Jahre verwandeln.

Bei einem Automaten, der nach dem Prinzip der Vorrangsteuerung arbeitet, befinden sich gleichzeitig mehrere Programme auf dem inneren Hauptspeicher. Der Einfachheit halber wollen wir nur zwei Programme betrachten. Das eine dieser Programme soll dabei stark mit Ein- und Ausgabevorgängen durchsetzt sein. Wir nennen es das Vorrangprogramm. Während der Pausen, die bei der Abarbeitung dieses Vorrangprogramms durch die Ein- und Ausgabe aktuelles Programm entstehen, übernimmt die Rechenmaschine automatisch die Abarbeitung des zweiten Programms, das in der Regel eine langwierige Rechenprozedur ist, die keine peripheren Geräte beansprucht.

Die Rechnung im zweiten Programm wird sofort unterbrochen, sobald das Vorrangprogramm die innere logische Schaltungsstruktur benötigt.
Das geschieht mit Hilfe sogenannter Unterbrechungssignale, die die peripheren Geräte aussenden, wenn die in ihren Pufferspeichern angehäuften Informationen ein »Abholen« fordern. Der Automat wird so gezwungen, seine letzten Reserven in den Dienst des Menschen zu stellen.



In ein solch wohlorganisiertes Programmzusammenspiel kann bei modernen Anlagen ein kilometerweit entfernter Kunde eingreifen und beispielsweise die Lösung einer Gleichung verlangen. Rechenanlagen, die das gestatten, arbeiten als sogenannte Teilnehmerrechensysteme.
An die Rechner werden geeignete Kommunikationsgeräte, wie Fernschreiber, Bildschirmeinheiten oder Fernbedienpulte, angeschlossen.

Ein kompliziertes, die Maschine beherrschendes Steuerprogramm, das man auch Betriebssystem nennt, sorgt für die Sicherung der laufenden Programme und der von ihnen erzielten Ergebnisse, stellt dem plötzlichen Anrufer Speicherplatz und Rechenzeit zur Verfügung, führt die gewünschten Berechnungen durch und übermittelt die Resultate. An große Rechenanlagen können viele Benutzer angeschlossen werden, von denen jeder den Eindruck hat, dass der Rechenautomat nur ihm allein dient.

Nicht alle Automaten verfügen über denselben technischen Komfort, haben dieselben Geschwindigkeiten und Speicherkapazitäten. Die Skala dieser Maschinen reicht von den am häufigsten anzutreffenden kleinen Anlagen über mittlere Konfigurationen bis zu den seltenen großen, nicht mehr in Serie gebauten Maschinen. Die im folgenden genannten Aufgaben sind aber im Prinzip von allen Datenverarbeitungsanlagen lösbar.

9.7 Alltagsarbeit

Der Alltag programmgesteuerter Rechenautomaten besteht aus harter, ununterbrochener Arbeit. Eingezwängt in Rechenzentren, von den Ingenieuren zu höchster Leistung angetrieben, werden sie maßlos und unfein beschimpft, wenn es ihnen einmal in den Sinn kommt, wegen Ausfall einiger Elemente nicht funktionsfähig zu sein.
Während dann die Techniker fieberhaft nach dem Fehler suchen, geraten Arbeitspläne durcheinander und Termine ins Wanken, verzögern sich wichtige, auf Grund von Rechenergebnissen zu treffende Entscheidungen im Produktionsprozess.

In keiner anderen Situation tritt uns so deutlich vor Augen, wie unentbehrlich Rechenautomaten heute schon geworden sind. Welche Aufgaben sind es nun, die von Automaten ständig gelöst werden?
Beginnend bei wissenschaftlichen Berechnungen in Instituten und Hochschulen, reichen sie bis zum täglichen Routineeinsatz in Großbetrieben. Hier haben sie ihren Anfang

in den Konstruktionsabteilungen bei der Überprüfung von Einzelteilen in Bezug auf Festigkeits- und Schwingungseigenschaften oder andere interessierende Merkmale.

Daran schließen sich Untersuchungen kompletter Produkte an. Beispielsweise können mehrere Varianten eines Motors unter Veränderung verschiedener Einflussgrößen berechnet und sogar konstruiert werden. Neben diesen technischen Aufgaben warten die ökonomischen Aufgaben auf ihre Lösung, die vor allem eine Bearbeitung großer Datenmengen umfassen. Das alles geschieht gewissermaßen in der Stille, als Alltagsarbeit des Automaten.

Weit interessanter sind natürlich solche Probleme, von denen man nicht von vornherein annehmen kann, dass sie zur Bearbeitung mit programmgesteuerten Rechenmaschinen geeignet sind, oder solche Aufgaben, die noch nicht zur täglichen Routine geworden sind.
Dazu zählt der Einsatz der Anlage als materielle Grundlage für integrierte Informationsverarbeitungssysteme, was im nächsten Kapitel noch ausführlicher dargestellt werden soll.

Bei der Betrachtung der folgenden Sonntagsarbeiten, die wahllos zusammengestellt sind, dürfen wir aber nicht aus dem Auge verlieren, dass der Löwenanteil des Nutzens, der uns durch die Anwendung programmgesteuerter Automaten erwächst, immer noch in der schnellen und sicheren Erledigung ständig anfallender geistiger Alltagsarbeit besteht.

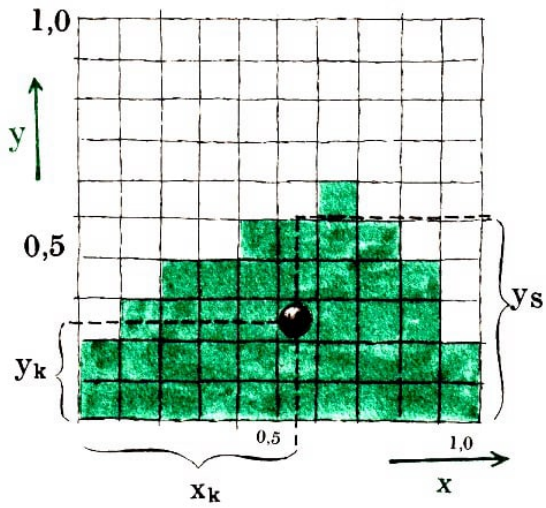
9.8 Ein Besuch in Monte-Carlo

Mit dem Namen der Stadt Monte-Carlo verbinden sich in unserem Bewusstsein die Vorstellungen von mondänen Salons, in denen sich frackbehängte Herren mit bleichen Gesichtern an großen Tischen gegenübersitzen und den Lauf von Glücksrädern verfolgen oder gespannt auf die nächste Spielkarte warten, um sich dann je nach Ausgang des Experiments bequem zurückzulehnen oder unauffällig eine Pistole an die Stirn zu setzen.

Zunächst bleibt unbegreiflich, was eine so unstete, ganz auf den Zufall begründete Angelegenheit wie das Glücksspiel mit strengen Rechenverfahren zu schaffen hat, die von Automaten beherrscht werden.
An einem Beispiel soll gezeigt werden, dass es durchaus einen Zusammenhang zwischen einer streng formelmäßigen Lösung einer Aufgabe und der Lösung mit Hilfe von statistischen Methoden geben kann.

Es sei der Flächeninhalt des schraffierten Gebietes in der folgenden Abbildung gesucht, wobei eine Seite des Quadrates die in irgendeinem Längenmaß angegebene Länge von 1 haben soll. Das Problem lässt sich zunächst auf die althergebrachte Weise durch Auszählen der Teilquadrate lösen.

Wir finden 39 Teilquadrate, von denen jedes einen Flächeninhalt $f = 0{,}1 \cdot 0{,}1 = 0{,}01$ hat, so dass sich für das gesamte schraffierte Gebiet eine Fläche von $A = 39 \cdot 0{,}01 = 0{,}39$ ergibt.



Wir stellen uns nun eine Versuchsbedingung vor, bei der jedes kleine Quadrat mit derselben Wahrscheinlichkeit von einer kleinen Kugel getroffen werden kann. Die Wahrscheinlichkeit dafür, dass die Kugel in das schraffierte Gebiet fällt, ist in diesem Fall zahlenmäßig so groß wie der Flächeninhalt des Gebietes, was wir den Betrachtungen bei der Einführung des Wahrscheinlichkeitsbegriffs entnehmen können.
Werfen wir viele Male eine Kugel auf das große Quadrat und zählen, wie oft sie unterhalb des gezeichneten Streckenzuges liegen bleibt, so wird sich der Quotient aus dieser Zahl m und der Gesamtzahl n der Würfe um so mehr der Wahrscheinlichkeit w annähern, je größer die Anzahl der Würfe ist. Genauer gesagt, gibt es eine gewisse Wahrscheinlichkeit

$$p\left(\frac{m}{n} \to w\right)$$

dafür, dass sich die Häufigkeit $\frac{m}{n}$ mit der die Kugel in das schraffierte Gebiet fällt, der Wahrscheinlichkeit w annähert. Die Wahrscheinlichkeit der Annäherung der Zahlen $\frac{m}{n}$ und w ist um so größer, je größer die Gesamtzahl n der Würfe ist.

Auf diesem fundamentalen Satz beruhen alle wahrscheinlichkeitstheoretischen Berechnungsverfahren, die auf Rechenautomaten durchgeführt werden und den Namen »Monte-Carlo-Methoden« erhalten haben.
Natürlich kann ein Rechenautomat keine Kugeln werfen. Wenn er aber in der Lage wäre, rein zufällige Zahlen zu erzeugen, die zwischen 0 und 1 liegen, so hätten wir nur zu prüfen, ob je zwei dieser Zahlen x_k und y_k einen Punkt im Inneren des schraffierten Gebietes bezeichnen oder nicht.

Der Quotient aus der Anzahl m dieser Paare, für die der entsprechende Punkt unterhalb des Streckenzuges liegt, und der Anzahl n aller erzeugten Zahlenpaare ergibt dann für großes n einen Näherungswert für den gesuchten Flächeninhalt. Dieser Näherungswert wird um so besser, je mehr Zahlenpaare (x_k, y_k) erzeugt werden.

In jedem Rechenschritt eines solchen Verfahrens hätten wir die Frage zu entscheiden, ob die zu einem gewürfelten x_k-Wert gehörende Höhe y_k des Streckenzuges größer ist als die gleichfalls »gewürfelte« Zufallszahl y_k. Das ist aber ein ganz klar gefasster, auf einer Rechenmaschine abzuarbeitender Algorithmus.

Die Erzeugung von echten Zufallszahlen ist auf dem Automaten natürlich nicht möglich, da er nur Algorithmen abarbeiten kann und sich ein Algorithmus eben dadurch auszeichnet, dass er feste reproduzierbare Ergebnisse und keineswegs irgendwelche zufälligen Zahlen liefert.
Man kann sich aber mit sogenannten Pseudozufallszahlen begnügen, die etwa auf folgende Art algorithmisch erzeugt werden.

Wir denken uns 2 zehnstellige Zahlen a und b, die miteinander multipliziert werden. Vom Ergebnis streichen wir die ersten 10 Stellen. Wenn sich dabei eine Zahl ergibt, die weniger als 10 Stellen besitzt, nehmen wir die letzten Stellen der Zahl b hinzu.
Diese so entstandene Zahl multiplizieren wir wieder mit der Zahl a und verfahren mit dem Ergebnis wie bei der ersten Multiplikation. Aus jeder so entstehenden zehnstelligen Zahl greifen wir nun beispielsweise die zweite, fünfte und achte Stelle heraus und schreiben die drei Ziffern zu einer neuen, dreistelligen Zahl.
Diese Zahlen sind offensichtlich als »gewürfelte« Zahlen ansprechbar.

Echte Zufallszahlen kann man der Rechenmaschine von außen zuführen, indem man gewisse physikalische Prozesse mit völlig statistischem Charakter, wie den radioaktiven Atomzerfall, dafür ausnutzt.

Im Gegensatz zu anderen Rechenverfahren haben wir bei den Monte-Carlo-Methoden keine Gewähr dafür, dass wir das richtige Ergebnis erhalten. Lediglich die Wahrscheinlichkeit, das richtige Ergebnis zu bekommen, wächst mit der Anzahl der Versuche.
Glücklicherweise wissen wir aber, dass diese Wahrscheinlichkeit für eine genügend große Zahl von Versuchen auch wirklich sehr groß sein kann. Der Nachteil der dennoch verbleibenden Unsicherheit wird bei den Monte-Carlo-Methoden durch einen Vorteil ausgeglichen, den andere Rechenverfahren nicht kennen. Ein Rechenfehler des Automaten hat bei statistischen Methoden praktisch keine Auswirkung auf das Ergebnis.

Mit der Verbreitung immer schnellerer Rechenautomaten werden die statistischen Verfahren zunehmend an Bedeutung gewinnen. Ein vierfaches Integral lässt sich bereits auf mittelschnellen Automaten mit statistischen Methoden schneller als mit Hilfe traditioneller Verfahren auswerten.
Nach diesem Ausflug in das Reich des Zufälligen wollen wir uns wieder bestimmten Dingen zuwenden und einmal einen Blick auf maschinelle Übersetzungen werfen.

9.9 Schlechte Übersetzungen

Der Gedanke, maschinell zu übersetzen, ist so alt wie die Rechenmaschine selbst. Bis zum Jahre 1950 hatte die Arbeit auf diesem Gebiet rein theoretischen Charakter. Heute sind die Kosten pro Wort einer Maschinenübersetzung bereits 10 mal geringer als die einer Übersetzung durch den Menschen.

Die meisten bekanntgewordenen Untersuchungen auf diesem Gebiet betreffen die beiden Sprachen Russisch und Englisch und beschränken sich höchst sinnvoll auf technische und wissenschaftliche Texte.

Natürlich können von einer Maschine keine vollkommenen Übersetzungen erwartet werden. Stilistisch müssen sie zwangsläufig unsauber sein. Aber auch gute menschliche Übersetzer sind selten.
Jeder, der schon selbst übersetzt hat, kennt zum Beispiel die ungeheuren Schwierigkeiten bei der Übertragung aus einem ihm fremden Fachgebiet. Wie oft haben solche Übersetzungen bei fachkundigen Lesern weniger das Wissen erhöht als die Lachmuskeln erregt. Man kann getrost sagen, dass Maschinenübersetzungen eine bessere Qualität haben werden, auch wenn der Lesende genötigt sein wird, kleine Korrekturen vorzunehmen und Ergänzungen anzubringen.
Dies aber kann ja von einem Leser, der mit dem Fachgebiet vertraut ist, erwartet werden.

Jedoch selbst wenn wir annehmen, dass maschinelles Übersetzen nie in großem Ausmaß zur Anwendung gelangen wird, können wir bereits einen Nutzen aus der Beschäftigung mit diesem Thema erkennen. Im Laufe der Vorbereitungen auf das maschinelle Übersetzen wurden detaillierte Sprachstudien getrieben, die vor allem den logischen Aufbau der Sprache betreffen.
Nicht von geringem Wert ist dabei die Tatsache, dass sich Sprachwissenschaftler mit Mathematik und Mathematiker mit Sprachwissenschaft befasst haben.

Grundlage jeder Übersetzung ist ein in irgendeiner Weise gespeichertes Wörterbuch. Bei der Maschinenübersetzung ergibt sich die Notwendigkeit, das Wörterbuch im Hauptspeicher anzulegen, um die Wörter schnell griffbereit zu haben. Die Anlage des Wörterbuches selbst erfolgt wie üblich in alphabetischer Reihenfolge. Auch das Nachschlagen in einem elektronischen Wörterbuch vollzieht sich wie beim normalen Wörterbuch. Ein zu übersetzendes Wort wird so lange mit den Wörtern der gespeicherten Liste verglichen, bis Übereinstimmung vorliegt. Das dazugehörige anderssprachige Wort wird dann zur weiteren Bearbeitung bereitgestellt.

Vor der Aufstellung eines elektronischen Wörterbuches muss Klarheit darüber bestehen, ob die Wörter in allen Formen, mit allen Beugungen und Abwandlungen vertreten sein sollen, wobei diese Abwandlungen jeweils eine gesonderte Eintragung bedingen, oder ob man sich der Methode »Stamm und Endung« bedienen will, bei der das Wort nur in seiner Grundform, als Stamm, gespeichert wird, während die Endungen durch Programme grammatischen Inhalts hinzugefügt werden.

Im zweiten Fall müssen die zu übersetzenden Wörter vor ihrem, Vergleich mit den Wörterbuchwörtern ebenfalls durch ein Programm auf ihrem Stamm verkürzt werden. Diese Methode ist also programmtechnisch komplizierter als die erste, hat aber den Vorteil, dass sie zur Unterbringung des Wörterbuches wesentlich weniger Speicherplatz benötigt. Fast alle elektronischen Wörterbücher sind auf dem Prinzip »Stamm und Endung« aufgebaut.

In jeder Sprache gibt es Redewendungen, die nicht Wort für Wort übertragen werden können. Das sind die sogenannten idiomatischen Konstruktionen. In den Wörterbüchern müssen sie gesondert gespeichert werden, bei der Übersetzung sind die Texte auf das Vorkommen solcher idiomatischer Wortgruppen zu untersuchen.

Ein Problem ist allerdings bis heute noch nicht befriedigend gelöst.
Es handelt sich um die Frage der Mehrdeutigkeit. Gewöhnlich verwenden die Sprachen für einen Begriff mehrere Ausdrücke; aber es kann auch der Fall sein, dass für verschiedene Begriffe ein und dasselbe Wort erscheint. Die beste Lösung dieses Problems bestände zumindest bei technischen und wissenschaftlichen Texten in der völligen Vermeidung von Mehrdeutigkeiten durch die Verwendung einheitlicher Begriffe und wohldefinierter Fachwörter.

9.10 »Die nächste Lochkarte bitte«

Die Vereinheitlichung von Begriffen und Bezeichnungen ist auch in der Medizin für den Einsatz elektronischer Rechenautomaten bedeutungsvoll. Eine wichtige Anwendung ist unter dem Namen automatische Diagnose bereits gut bekannt.

Jede Krankheit zeigt ein für sie charakteristisches Bild. Der Versuch, die Krankheit aus den äußeren Merkmalen und Anzeichen zu bestimmen, stößt aber oft auf Schwierigkeiten. Viele Krankheiten zeigen in der Anfangsphase dieselben Symptome, einige Anzeichen werden nicht bemerkt oder für Merkmale anderer Krankheiten gehalten.

So erscheint zum Beispiel sowohl bei der Angina als auch bei der Diphtherie ein gelblichweißer Belag auf den Mandeln. Der Arzt ist also verpflichtet, an alle möglichen Krankheiten zu denken, die gleiche oder ähnliche Symptome am Patienten hervorrufen. Dabei kann ihm ein Automat wertvolle Dienste leisten, indem er wirklich alle in Verdacht kommenden Erkrankungen aufzählt, die in seinem Speicher zu finden sind, während es einem Arzt durchaus unterlaufen kann, dass ihm eine bestimmte Krankheit einfach nicht einfällt.
Bei allen bisher unternommenen Versuchen hat sich gezeigt, dass der Automat immer an mehr Krankheiten gedacht hat als der Arzt.

Voraussetzung für eine automatische Diagnose ist eine Systematisierung der Krankheiten nach ihren Symptomen. Das ist natürlich eine sehr große, aber keineswegs unlösbare Aufgabe. Unterstellen wir, dass ein Komplex von Krankheiten, zum Beispiel die Gruppe der Augenerkrankungen, über eine derartige Systematisierung bereits verfügt, so können wir jedem äußeren Anzeichen und jeder Krankheit Zahlen zuordnen und diese in Form einer Tabelle auf dem Hauptspeicher eines Automaten festhalten.

Bei der Voruntersuchung des Patienten werden alle beobachteten Symptome in ein Formular geschrieben und anschließend auf Lochkarten übertragen. Nach der Eingabe dieser »Befund-Karten« in den Automaten ermittelt ein spezielles Suchprogramm alle Krankheiten, auf die die Symptome zutreffen oder für die sie »ähnlich« sind.
Die Nummern dieser Krankheiten werden ausgedruckt. Erst mit dem Ergebnis der Voruntersuchung geht der Patient zum Arzt, der dann entscheiden muss, welche Krankheit wirklich vorliegt.

Aus dem Dargelegten ist zu erkennen, dass alle Vorurteile gegen die automatische Diagnose unberechtigt sind. Keineswegs wird das persönliche Verhältnis zwischen Patient und Arzt zerstört, und ebenso wenig kann man davon reden, dass der Arzt durch den Automaten ersetzt wird.

9.11 Die verblüfften Gangster

Genauso unberechtigt wäre es, zu behaupten, dass die Kriminalbeamten überflüssig werden, nur weil sie bei ihren Untersuchungen neuerdings programmgesteuerte Rechenmaschinen verwenden.
In St. Louis/USA soll sich vor wenigen Jahren angeblich folgende Geschichte zugetragen haben.

Drei gefährliche Einbrecher waren schwerbewaffnet nachts in ein Warenhaus eingedrungen. Sie hatten die Alarmanlage außer Betrieb gesetzt und begaben sich recht zuversichtlich auf den Weg in den Kassenraum. Ihr Unternehmen war absolut geheim, sie besaßen keine Mitwisser.
Ihre früheren Tatorte waren nach einem ausgeklügelten System gewählt, ein umfangreicher Informationsdienst unterrichtete sie über alle möglichen Gefahrenpunkte. Nach menschlicher Voraussicht konnten sie daher mit einem glücklichen Gelingen ihrer Operation rechnen.

Sie erreichten wie gewohnt ohne Schwierigkeiten die Tür des Kassenraums, öffneten sie und blickten grenzenlos überrascht in die Augen einiger Polizisten. Ohne den geringsten Widerstand zu leisten, ließen sich die Gangster festnehmen.

Eine Art Rührung überkommt uns, wenn wir die Verblüffung und die gleichzeitig einsetzende Denklähmung der drei so siegesgewiss ausgezogenen Helden nachempfinden. In der Tat hat sie kein Kommissar Smith zu Fall gebracht. Bereits ihre ersten erfolgreichen Unternehmungen hatten einen Gegner auf den Plan gerufen, der ihnen überlegen war. Eine Datenverarbeitungsanlage hatte alle ihre Untaten registriert und analysiert. Auf ein umfangreiches Informationsmaterial gestützt, ermittelte die Maschine den Ort, für den die größte Wahrscheinlichkeit bestand, als nächster Tatort gewählt zu werden. Auch für den Zeitpunkt der voraussichtlichen Tat erhielt man, wie sich später erwies, eine hinreichend genaue Angabe.

In Chikago wird eine Datenverarbeitungsanlage zur täglichen Analyse der Verbrechen in der Stadt verwendet. Dadurch besitzt die Polizeizentrale eine ständig aktuelle und gewichtige Unterlage für den Einsatz von Streifen und Kriminalisten.

9.12 Aufstellung von Stundenplänen

Nicht so aufregend wie die Pläne zum Einsatz von Polizisten sind die Pläne zur Verteilung von Lehrern auf Schulklassen.
Zu Beginn jedes Unterrichtsabschnitts werden an allen Schulen Stundenpläne aufgestellt. Diese Aufgabe übernehmen oft erfahrene »Stundenplanspezialisten«, die durch ein individuelles Probierverfahren versuchen, die vorhandenen Lehrer den Klassen und

Unterrichsstunden so anzuordnen, dass zumindest keine unmöglichen Situationen entstehen.
Diese Arbeiten beanspruchen oft viele Stunden. Das scheint noch vertretbar.

Bilden wir aber die Summe dieser Zeiten für alle Schulen und bedenken, wie oft Stundenpläne aufgestellt werden müssen, so erkennen wir, dass hier wertvolle menschliche Arbeitszeit vergeudet wird.
Einer Automatisierung auf diesem Gebiet muss eine gründliche Untersuchung der Forderungen, die an einen Stundenplan gestellt werden, vorausgehen. Man unterscheidet zwischen Bedingungen, die bei der Aufstellung des Stundenplanes unbedingt eingehalten werden müssen, und solchen Bedingungen, die soweit wie möglich erfüllt werden sollen.
Zu der ersten Gruppe zählen unter anderem folgende Forderungen:
Für den Unterricht steht eine feste Anzahl von Wochenstunden zur Verfügung, und von Woche zu Woche soll sich ein gleicher Ablauf ergeben. Die Anzahl der Wochenstunden für jedes Fach und jede Klasse sind fest vorgegeben.
Einige Fächer, wie Chemie und Physik, können nur in entsprechenden Fachräumen unterrichtet werden. Die Schüler dürfen keine »Zwischenstunden« haben.

Die zweite Gruppe von Bedingungen enthält etwa folgende Wünsche: Die einzelnen Fächer sollen so gleichmäßig wie möglich über die Woche verteilt werden. Hauptfächer sollen in den frühen Unterrichtsstunden liegen. Die Anzahl der Zwischenstunden eines Lehrers soll so gering wie möglich sein.

Für die Belange einer automatischen Aufstellung von Stundenplänen werden alle Bedingungen in Form von Gleichungen und Ungleichungen formuliert. Zur Lösung des Problems existieren bereits einige mehr oder weniger gute Verfahren.
Ein allgemeiner Algorithmus, der so frühzeitig wie möglich später eintretende Widersprüche auf dem eingeschlagenen Weg anzeigt und erkennen lässt, wodurch sie verursacht werden, ist zur Zeit noch nicht bekannt. Wollte man diese Tatsache zum Anlass nehmen, die automatische Aufstellung von Stundenplänen abzulehnen, so ist dem entgegenzuhalten, dass bereits die mit den vorliegenden Verfahren erreichten Stundenverteilungen im Sinne der einzelnen Forderungen weit besser sind als die vom Menschen aufgestellten.

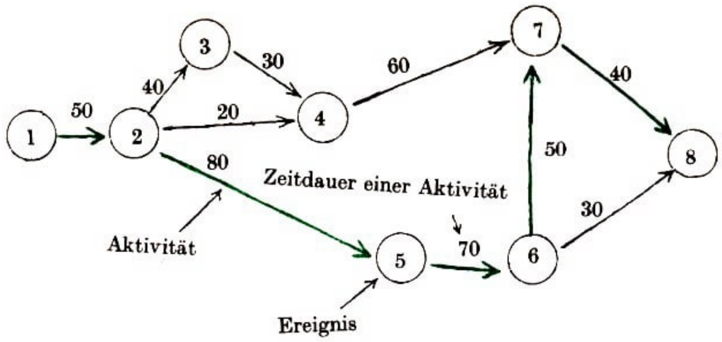
Zum Schluss sei in groben Zügen ein Verfahren skizziert, das die Aufstellung eines Stundenplans ermöglicht. Es beruht darauf, dass den einzelnen Fächern vor Beginn der Verteilung gewisse Bewertungen erteilt werden.
So können zum Beispiel Mathematik, Physik und Chemie hohe Bewertungsziffern erhalten. Die Verteilung beginnt mit dem Fach, das die höchste Bewertungsziffer besitzt. Unter Berücksichtigung aller auferlegten Bedingungen werden eine oder mehrere Unterrichtsstunden mit diesem Fach belegt. Dadurch erniedrigt sich die Bewertung dieses Faches, ein anderes Gebiet ist nun an der Reihe. Auf diese Weise wird eine recht gleichmäßige Verteilung der am höchsten bewerteten Fächer erreicht.

9.13 Kritische Wege

Die Planung von Stundenverteilungen an einer Schule ist im Vergleich zur Planung des Ablaufs umfangreicher und komplizierter Bauvorhaben und vieler anderer Projekte eine noch recht einfache Aufgabe.
Denken wir nur an die Vielzahl miteinander verflochtener Einzelarbeiten, so wird deutlich, dass die Planung solcher Unternehmungen mit großer Sorgfalt erfolgen muss.

Ein Verfahren, das zunehmend bei der Planung und Kontrolle komplizierter Projekte zur Anwendung gelangt, ist unter dem Namen »Methode des kritischen Weges« bekannt geworden. Diese Methode ist aus dem Gedanken entstanden, die in der Technik verbreitete Analyse von Netzwerken auf die Untersuchung von Planungsvorgängen anzuwenden.

Es sei zum Beispiel ein umfangreiches Bauvorhaben zu verwirklichen.
Alle anfallenden Arbeiten werden in der Form eines Netzes zusammengestellt, das genau einen Anfangspunkt und einen Endpunkt besitzt. Dabei bedeuten die Pfeile die einzelnen Arbeiten und die Kreise den Beginn und das Ende einer Arbeit. Die Pfeile werden in der Netzwerktechnik Aktivitäten und die Kreise Ereignisse genannt. Es sei weiterhin vorausgesetzt, dass die zur Realisierung jeder Arbeit oder Aktivität erforderliche Zeit bekannt ist.



Sie wird in der Netzdarstellung unmittelbar an der jeweiligen Aktivität vermerkt. Zum Verständnis eines Netzwerkes verhilft folgender Satz: »Eine Aktivität kann erst dann beginnen, wenn die vorhergehenden Aktivitäten, die im Anfangspunkt der betrachteten Aktivität enden, abgeschlossen sind.«
Auf diese Weise wird die technologische Abhängigkeit der einzelnen Arbeiten sichtbar.

Man kann sich leicht überzeugen, dass es innerhalb des Netzes mehrere Möglichkeiten gibt, vom Anfangspunkt, der den Beginn des Projektes darstellt, zum Endpunkt zu gelangen, der den Abschluss aller Arbeiten bezeichnet.
Summieren wir die Zeiten aller Aktivitäten, die auf einem bestimmten Weg vom Anfangspunkt zum Endpunkt liegen, so erhalten wir die Länge dieses Weges oder, was dasselbe ist, die Gesamtzeitdauer aller Arbeiten, die in dieser Reihenfolge durchgeführt werden. Der längste in einem Netzwerk mögliche Weg wird kritischer Weg genannt. Er bestimmt die Gesamtzeitdauer des Projektes und legt somit den Endtermin fest.

Auf der Zeichnung erkennt man den kritischen Weg 1-2-5-6-7-8.
Für alle Arbeiten, die auf diesem kritischen Weg liegen, gibt es keine zeitlichen Reserven oder sogenannte Pufferzeiten. Die Kenntnis des kritischen Weges erlaubt festzustellen, wo bei der Durchführung der Arbeiten besondere Termintreue zu herrschen hat und welche Arbeiten zu beschleunigen sind, falls eine Terminverkürzung für das gesamte Projekt erreicht werden muss.
Stellte man früher bei einem Bauvorhaben Terminverzögerungen fest, so wurden plötzlich irgendwelche und nicht selten alle noch zu erledigenden Arbeiten hektisch beschleunigt, obwohl es ausgereicht hätte, nur die auf dem kritischen Weg liegenden Arbeiten schneller voranzutreiben.
Vergleicht man ständig die geplanten Werte der Aktivitäten mit den Werten, die in der Wirklichkeit anfallen, und berechnet in angemessenen Abständen den kritischen Weg des Netzes mit den in der Praxis aufgetretenen Werten, so hat man eine wirksame Kontrolle über alle entscheidenden Arbeiten und kann jederzeit in den Ablauf des Geschehens eingreifen, wenn sich eine Überschreitung des Endtermins ankündigt.
Die inzwischen weit verbreitete Methode des kritischen Weges ermöglicht also eine ständige Vorhersage des Endtermins auf Grund vorliegender aktueller Daten.

Im folgenden Abschnitt werden wir eine andere Vorhersage betrachten, die nach böswilligen Ansichten viel seltener zutrifft.

9.14 Selbst 64000 Wetterfrösche sind zu wenig

Der amtliche Wetterbericht ist heute noch oft ein Produkt, dessen Zustandekommen den durch viele Beobachtungen entstandenen Erfahrungsregeln der Meteorologen zu verdanken ist.
Bereits am Anfang unseres Jahrhunderts wurde eine objektive Vorhersagemethode entworfen, die sich auf umfangreiche Rechnungen stützen sollte.

Das Wettergeschehen wird durch allgemeine Gesetze der Thermodynamik und der Hydrodynamik beschrieben, die ein kompliziertes Differentialgleichungssystem ergeben. Für das Vorhersagegebiet müsste dieses System nun zahlenmäßig gelöst werden. Zur Erledigung aller dabei anfallenden Arbeiten wären täglich 64000 Menschen als Rechner erforderlich, die obendrein in einem Stadion untergebracht sein müssten, da sie nicht unabhängig voneinander arbeiten könnten.

Die Entwicklung moderner Datenverarbeitungsanlagen hat das Projekt eines derartigen Wetterbüros gegenstandslos gemacht. Bei der numerischen Wettervorhersage werden die über Fernschreiber eintreffenden Angaben gleichzeitig auf einen Lochstreifen übertragen und dem Automaten eingegeben.
Die Maschine berechnet dann die Linien gleichen Luftdrucks und gibt diese über ein spezielles Zusatzgerät in Form von meteorologischen Karten aus. Danach beginnt sie

mit der Berechnung der künftigen Zustände und gibt die Ergebnisse wiederum in Form von Wetterkarten aus.
Diese Berechnung bezieht sich im wesentlichen nur auf den Luftdruck, die wichtigste meteorologische Zustandsgröße. Die sekundären Erscheinungen, wie Bewölkung, Windrichtung oder Niederschläge, müssen von den Meteorologen noch selbst bestimmt werden.

Auch die numerische Wettervorhersage ist nicht »astronomisch genau«. Ihre Ergebnisse werden von zwei Tatsachen entscheidend beeinflusst.
Zunächst muss man wegen der komplizierten Beschaffenheit der Atmosphäre mit einem stark vereinfachten Modell arbeiten. Dazu kommt die Beeinflussung des Gebietes, auf das sich die Rechnung bezieht, durch das Wettergeschehen, das sich außerhalb dieses Gebietes abspielt.

So kann vom Einsatz elektronischer Rechenautomaten in der Meteorologie noch keine exakte langfristige Wettervorhersage für kleine Gebiete erwartet werden. Der Nutzen dieser Maschinen liegt zunächst in der Automatisierung aller anfallenden geistigen Routinearbeiten.
Während die weitreichende Bedeutung des Einsatzes elektronischer Rechenautomaten in den meisten vorangegangenen Beispielen ohne weiteres einzusehen war, kann bei den jetzt folgenden nicht mehr in jedem Fall von einer wirklichen Notwendigkeit gesprochen werden. Wenn solche Beispiele dennoch angeführt werden, so geschieht das, um die Vielfältigkeit in der Anwendung programmgesteuerter Rechenautomaten zu zeigen.

9.15 Matt in zwei Zügen

Ein beliebtes Problem ist das Suchen nach Algorithmen, die einer Rechenmaschine das Schachspielen ermöglichen. Schachspielende Automaten gab es bereits in der Vergangenheit.
Einige beruhten auf einem geschickt verborgen gehaltenen Betrug; anstelle der Maschine spielte ein Mensch. Wieder andere waren sinnreich konstruierte, mechanisch arbeitende Apparate, die ein Spiel mit etwa 3 bis 4 Figuren ermöglichten.

In der letzten Zeit sind für Elektronenrechner einige Programme entstanden, die die Schachaufgabe »Matt in zwei Zügen« lösen, wenn die Anzahl der Figuren nicht zu groß ist.
Diese Programme beruhen auf der systematischen Untersuchung aller möglichen Züge, deren Anzahl bei einer geringen Zahl von Figuren noch nicht ins Uferlose wächst. Die Automaten lösen solche Probleme gegenwärtig in einer Zeit von nur wenigen Minuten.

Natürlich entspricht ein derartiges Vorgehen nicht dem, was wir eigentlich unter Schachspielen verstehen. Auf einem höheren Niveau stehen Versuche, Bewertungsfunktionen zu finden, die das Gesamtspiel beurteilen und die bei »optimalem« Spiel zu einem Minimum oder Maximum gemacht werden müssen.

Besonders reizvoll scheint auf den ersten Blick das Spiel zweier Automaten gegeneinander. In Wirklichkeit spielen aber nicht zwei Maschinen, sondern zwei Programme. Sind

diese voneinander verschieden, liegen ihnen also verschiedene Algorithmen zugrunde, so wird sich recht schnell herausstellen, welches Verfahren das bessere ist.
Beim Spiel zweier völlig gleicher Programme dürfte es in der Regel zu einem uninteressanten Schlagabtausch kommen.

9.16 Die Feststellung von Schwätzern

Erwähnt man in Gegenwart literarisch interessierter Menschen die Möglichkeit, einen vorgelegten Text von einem Automaten auf seinen Informationsgehalt prüfen zu lassen, so stößt man auf eine nicht geringe Abwehr.
Wir wollen diese zunächst gelten lassen und für das Gebiet wissenschaftlicher Texte einige Feststellungen treffen, die die Protestierenden nachdenklicher stimmen sollen.

In einer wissenschaftlichen Bibliothek ist es wünschenswert, die eingehenden Bücher und Zeitschriftenartikel nicht nur zu katalogisieren, sondern auch jede Karteikarte mit einer kurzen, charakteristischen Inhaltsangabe der betreffenden Veröffentlichung zu versehen. Das setzt voraus, das jedes Buch und jeder Artikel gelesen werden. Nur selten stehen aber für jedes Gebiet die entsprechenden sachkundigen Bearbeiter in der gewünschten ausreichenden Anzahl zur Verfügung.

Man kann sich nun überraschenderweise mit Erfolg für den entgegengesetzten Weg entschließen und Bearbeiter auswählen, die nicht die geringste Sachkenntnis besitzen, dafür aber schnell und ausdauernd arbeiten können. Wir wissen bereits, dass auch hier wieder Automaten gemeint sind.
Von diesen Rechnern muss aber zusätzlich verlangt werden, dass sie »lesen« können, d. h., dass sie in der Lage sind, die einzelnen Buchstaben und Zeichen auf dem gedruckten Blatt zu unterscheiden. Andernfalls müsste eine Übertragung auf ein geeignetes Zwischenmedium erfolgen, was natürlich die Aufgabe sofort unrentabel, aber keineswegs unmöglich macht.

Man geht von einem Algorithmus aus, der die Berechnung der Informationsmenge einer beliebigen Folge von Buchstaben und Zeichen ermöglicht. Dabei verwendet man die Häufigkeit der einzelnen Zeichen als Maß für die Wahrscheinlichkeit ihres Auftretens.
Mit einem solchen Programm wird nun Satz für Satz des vorgelegten Textes untersucht und festgestellt, welche Sätze die größten Informationsmengen besitzen. Diese Sätze werden von der Maschine ausgedruckt und erweisen sich in der überwiegenden Anzahl der Fälle so charakteristisch für die Veröffentlichung, dass jeder Fachmann mit genügender Sicherheit über deren Wert urteilen kann.

Wir erkennen hier einen berechtigten Ansatzpunkt für die Kritik an der Übertragung dieser Methode auf Texte nicht wissenschaftlicher Art. Wir besitzen beispielsweise noch keine allgemeinen Kriterien, nach denen der Wert oder der Unwert eines Gedichtes beurteilt werden könnten. Ein Roman mit großem Informationsgehalt kann uns langweilen.

Das mag für diejenigen, die mathematische Untersuchungen auf dem Gebiet der Kunst ablehnen, sehr freundlich klingen. Indessen befasst sich die Informationstheorie seit einiger Zeit mit Fragen, die sich nicht nur auf die Menge der Information beziehen, sondern

auch den »Wert« der Information zum Gegenstand haben.
Die Grundlage dieser Untersuchungen bildet die Bemühung, eine exakte Definition für den Begriff »Wert der Information« zu finden.
Man geht dabei von der Voraussetzung aus, dass der Empfänger von Informationen mit der Lösung eines Problems beschäftigt ist. Je mehr ihm die erreichenden Informationen dabei behilflich sind, desto wertvoller sind sie.

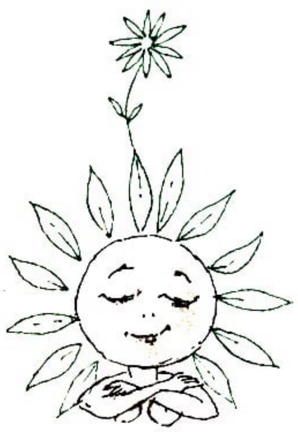
Natürlich muss auf diesem Gebiet noch viel Arbeit geleistet werden, ehe eine maschinelle Feststellung von Schwätzern stattfinden kann.

9.17 In der Akademie von Lagado

Völlig berechtigt ist dagegen die Kritik an Versuchen, den Automaten die Rolle von Künstlern zu erteilen. Als Kuriosum sollen die Umrisse eines Programms angegeben werden, das in der Lage ist, Gedichte herzustellen. Vorläufig handelt es sich dabei nur um reimlose Lyrik.

In den Hauptspeicher der Maschine wird eine gewisse Menge von Hauptwörtern eines beliebigen Themenkreises eingegeben. Das können Worte wie Sonne, Knospe, Veilchen und Wiese sein. Dazu nimmt man noch eine Liste passender Tätigkeitswörter wie »erwachen«, »blühen«, »scheinen« usw. Damit die Gedichte nicht eintönig werden, kann man sich noch für die schmückenden Beiwörter »schön«, »hell« und »freundlich« entscheiden. Weiterhin benötigt man die Artikel »der«, »die« und »das« sowie das Wörtchen »ist« und das verbindende »und«.

Dann wird eine Liste von zugelassenen Satzkonstruktionen aufgestellt. Diese Liste enthält alle Satzarten, die in den Gedichten vorkommen sollen. Als Beispiel lassen wir folgende Satzstrukturen zu.

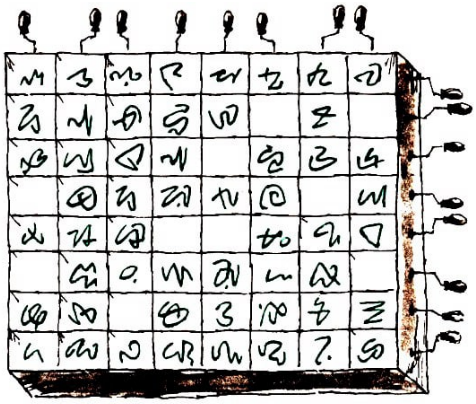


1. Das Veilchen ist schön.
2. Das Veilchen blüht.
3. Das Veilchen ist schön und blüht.
4. Das schöne Veilchen blüht.

Nun kann das eigentliche Dichtprogramm in Aktion treten. Mit Hilfe von Zufallszahlen wird zunächst eine Satzform gewürfelt. Das sei zum Beispiel die Form 2. Daraufhin erfolgt auf dieselbe zufällige Weise die Ermittlung eines Substantivs; das sei hier »Sonne«.
Ein Unterprogramm ermittelt dazu das passende Wörtchen »die«. Somit ist unser Kunstwerk bereits bis »Die Sonne ...« gediehen.

Jetzt tritt wieder das Zufallszahlenprogramm in Aktion und ermittelt ein beliebiges Tätigkeitswort. Wir nehmen an, dass die Maschine hier auf das Word »blühen« stößt. Ein Programm grammatischen Inhalts ermittelt die richtige Form, und der erste Satz des Gedichts lautet: »Die Sonne blüht.«

Mit der Zulassung vieler Wortarten, Wörter und Satzkonstruktionen können nun recht originelle Gedichte entstehen. Es sind Fälle bekannt, wo solche Werke nach ihrer Veröffentlichung in Tageszeitungen begeisterte Zuschriften ausgelöst haben, obwohl die Leser nicht wussten, dass die Gedichte maschinell hergestellt worden waren.
Ein derartiges Vorgehen ist eine reine Spielerei, die das eigentliche Anliegen der Rechenautomaten, algorithmisch fassbare geistige Arbeit auszuführen, nicht in Verruf bringen kann. Dass man auf den Gedanken kam, solche Programme zu schreiben, mag vielleicht daran liegen, dass bei der Entstehung von Gedichten durchaus zufällige »Eingebungen« eine wesentliche Rolle spielen können.



Das angedeutete Programm zur Erzeugung von Texten hat eine große Ähnlichkeit mit dem Verfahren eines Professors der berühmten Akademie von Lagado, die Jonathan Swifts Gulliver im Verlauf seiner Reisen aufgesucht hat.
Innerhalb eines quadratischen Rahmens von 20 Fuß Länge waren auf Holzstücken die einzelnen Wörter der Landessprache geschrieben. Durch eine Drehung seitlich angebrachter Kurbeln konnte das ganze Wortsystem plötzlich geändert werden. Sinnvolle Wortkombinationen wurden abgeschrieben. Auf diese Weise wollte der Professor der Welt »einen vollständigen Inbegriff aller Künste und Wissenschaften« geben.

9.18 Erzeugung von Zwölftonmusik

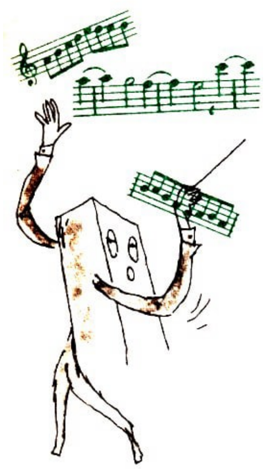
Auch die Musik blieb von den Automaten nicht verschont. Wie bei der Literatur wollen wir zunächst über Analysen bestehender Werke sprechen, ehe wir das elektronische Komponieren selbst betrachten.
Bei einer mathematischen Musikanalyse soll natürlich nur versucht werden, rein äußerliche Gesetzmäßigkeiten zu finden, die nichts über den Sinngehalt des Musikstückes aussagen. Wegen der Kompliziertheit musikalischer Werke beschränkt man sich von vornherein auf die Untersuchung einfacher Elemente, wie Tonhöhe, Dauer der Einzeltöne usw. Diese Elemente werden mit statistischen Verfahren maschinell analysiert.

Eine wichtige Aussage liefert dabei unter anderem der sogenannte Korrelationskoeffizient, der den Grad der gegenseitigen Abhängigkeit zweier Größen angibt, wenn von beiden eine Reihe von »Messwerten« vorhanden ist.

Nach solchen statistischen Kriterien wurden nun einige Orchesterwerke aus verschiedenen Stilepochen mit Hilfe von Rechenanlagen untersucht. Man stellte zum Beispiel

fest, dass die Streuung der Tonhöhe in den Jahren zwischen 1600 und 1900 von 4% auf 10% ständig zunahm.
Bei aufeinanderfolgenden Tönen herrschen in der klassischen und romantischen Musik die kleineren Intervalle vor.

Eine deutliche Unstetigkeit konnte man bei der Zwölftonmusik beobachten, bei der die statistischen Merkmale sich kaum von denen unterscheiden, die rein zufällige Prozesse besitzen. Man könnte also solche Musik von Automaten verfertigen lassen, indem man ein Zufallszahlenprogramm für den Aufruf der Töne schreibt.
Das soll natürlich kein Werturteil darstellen, zumal der Automat in der Lage ist, auch »schönere« Musik zu erzeugen.



Ein bekannter Professor der angewandten Mathematik pflegte gegen Ende seiner Einführungsvorträge über Rechenautomaten mit Hilfe eines Tonbandgerätes etwas klassische Musik vorzuführen.
Sein Publikum, das mit Interesse und sichtlichem Vergnügen lauschte, erriet natürlich sofort, dass es sich um Musik handelte, die von Rechenmaschinen komponiert war, obwohl man zumindest beim ersten Hören nichts finden konnte, was sie als solche gekennzeichnet hätte.
Genau wie bei der maschinellen Herstellung von Gedichten handelt es sich beim elektronischen Komponieren um einen Spaß, den man den oft geplagten Programmierern zugestehen muss, ohne auf den Gedanken zu verfallen, die Automaten könnten, weil sie viel billiger arbeiten, die Komponisten verdrängen.

Beim elektronischen Komponieren unterscheidet man zur Zeit zwischen zwei Methoden. Bei der ersten werden eine große Menge verschlüsselter Tonfolgen in den Speicher des Automaten gegeben. Ein Programm, das Zufallszahlen erzeugt, stellt diese Folgen von Tönen in verschiedener Reihenfolge wahllos zusammen. Für den Spezialfall, dass nur Einzeltöne eingegeben werden, entsteht dann die Zwölftonmusik.

Bei der zweiten Methode werden die einfachsten Gesetze der Harmonik als Algorithmen dem Automaten mitgeteilt. Ein Zufallsprogramm variiert nun die Töne innerhalb der durch die Gesetze festgelegten Grenzen.

Durch Anschluss geeigneter elektrisch betriebener Musikinstrumente kann sogar erreicht werden, dass der Automat seine Kompositionen selbst aufführt.

Nach dem flüchtigen Durchstreifen so vieler Anwendungsgebiete der maschinellen Rechentechnik wollen wir jetzt den entgegengesetzten Weg gehen und ein einziges Anwendungsgebiet genauer betrachten. Wir wollen die elektronische Datenverarbeitungsanlage als Instrument der Planung und Leitung im volkseigenen Betrieb kennenlernen.

10 Der durchsichtige Betrieb

10.1 Wie der Lohn errechnet wird

In der Vergangenheit wurde eine inzwischen ausgestorbene Methode der Vorbereitung für den Einsatz einer Datenverarbeitungsanlage häufig angewandt. Wenn ein Betrieb eine solche Anlage bestellen wollte oder bereits bestellt hatte, wurde zunächst darauf gesehen, dass einige periodisch auftretende Aufgaben zumeist aus dem Abrechnungsbereich schnell programmiert wurden, um die Anlage so früh wie möglich auszulasten. Zu dieser Art von Problemen gehört die Lohnabrechnung eines Betriebes. Wir wollen zeigen, wie eine derartige Aufgabe von einer elektronischen Datenverarbeitungsanlage gelöst werden kann.

Zunächst muss der sogenannte Ist-Zustand des Problems ermittelt werden: Jeder Arbeiter füllt nach der Beendigung einer Arbeit an der Maschine einen Lohnschein aus, auf dem die Art der Arbeit, die dafür benötigte Zeit, die Lohngruppe und einige weitere Angaben festgehalten werden. Die Lohngruppe gibt an, wie hoch eine Arbeitsstunde zu bezahlen ist.

Dazu wird ein sogenannter Lohnfaktor L_G für jede Lohngruppe vorgegeben. Die Multiplikation dieses Faktors mit der für die Arbeit benötigten Zeit t_A ergibt den dazugehörigen Bruttoverdienst B_A

$$B_A = t_A \cdot L_G$$

Hat nun ein Arbeiter im Laufe des Monats mehrere Arbeiten A_1, A_2, ..., A_N erledigt, so ist sein gesamter Bruttoverdienst die Summe der einzelnen Bruttoverdienste, was in Formeln folgendermaßen geschrieben werden kann:

$$B = B_{A1} + B_{A2} + ... + B_{AN} = L_G \cdot t_{A1} + L_G \cdot t_{A2} + ... + L_G \cdot t_{AN}$$

Mit dem Summenzeichen $\sum$ lässt sich das noch kürzer darstellen:

$$B = \sum B_{AN} = L_G \cdot \sum t_{AN}$$

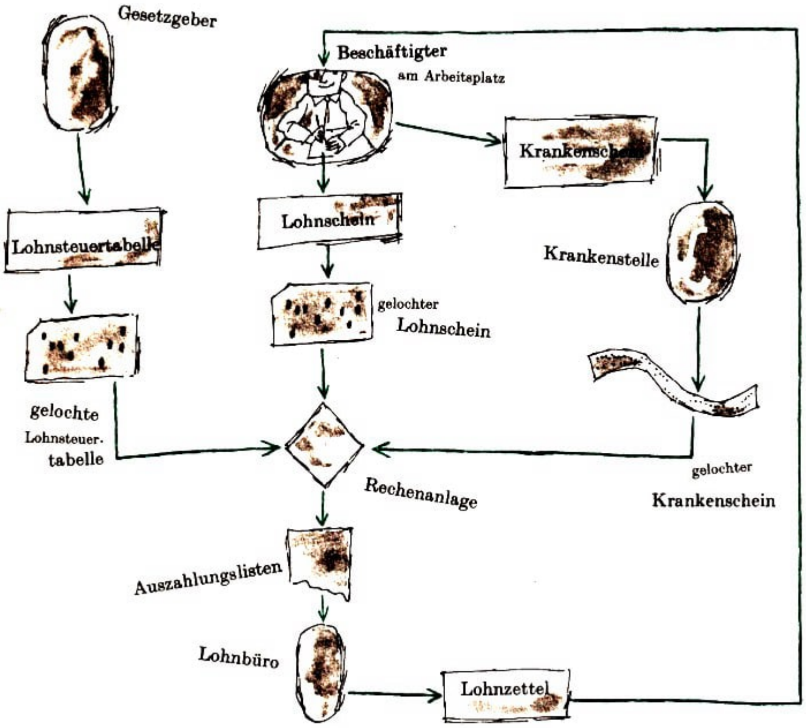
Liegt ein Krankheitsfall vor, so werden weniger Lohnscheine vorhanden sein, dafür zahlt aber die Sozialversicherung für jeden Krankheitstag 90% des durchschnittlichen Bruttoverdienstes. Somit ergibt sich der Bruttoverdienst eines Arbeiters im allgemeinen

$$B = \begin{cases} L_G \cdot \sum t_{AN} & \text{ohne Krankheit} \\ L_G \cdot \sum t_{AN} + 0{,}9 \cdot n_T \cdot D_T & \end{cases}$$

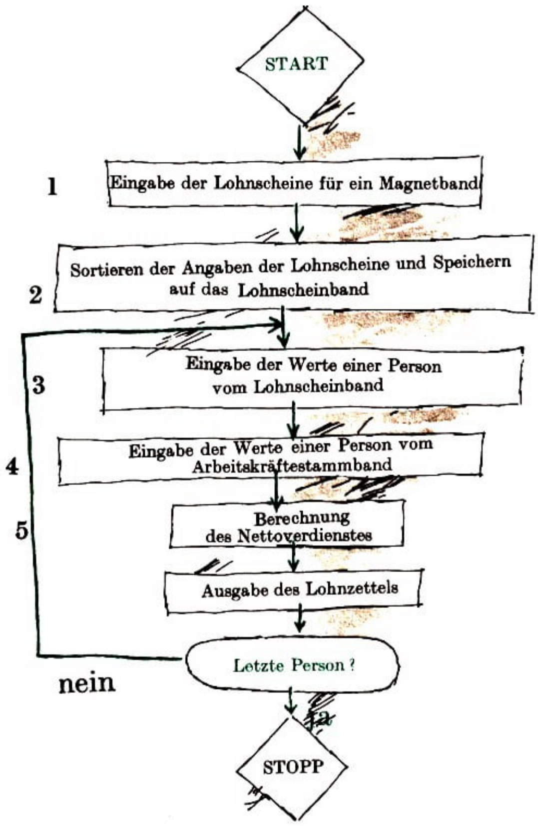
Dabei ist n_T die Anzahl der Krankheitstage und D_T der durchschnittliche tägliche Bruttoverdienst, der zum Beispiel aus den Löhnen des vergangenen Jahres ermittelt werden kann.

Aus dem Bruttoverdienst muss dann nach der gesetzlichen Lohnsteuertabelle die dazugehörige Lohnsteuer ermittelt werden, die von der Steuerklasse abhängt, wobei das Krankengeld nicht versteuert werden soll. Nach Abzug der Lohnsteuer von der Bruttosumme ergibt sich der Nettoverdienst, der vom Lohnbüro ausgezahlt werden muss.

Das ist in groben Zügen der Sachverhalt der Lohnabrechnung, der nun mit Hilfe eines Programms auf die elektronische Datenverarbeitungsanlage übertragen wird.



Zunächst wollen wir ein Ablaufdiagramm der Vorgänge zeichnen. Dabei bedienen wir uns am Anfang einer Darstellungsform, aus der lediglich der Zusammenhang der erforderlichen Informationen und ihr zeitlicher Fluss erkennbar sein sollen. Man nennt eine solche Übersichtsdarstellung mit genormten Symbolen einen Datenflussplan.



Jetzt kommt es darauf an, den Algorithmus der eigentlichen Berechnung zu formulieren, die im Inneren des Automaten ablaufen soll. Das Programm soll so eingerichtet werden, dass die gelochten Lohnscheine in das Karteneingabegeräte gegeben und mit Hilfe der Zentraleinheit auf ein Magnetband gespeichert werden.
Diese Eingabe soll unsortiert erfolgen, die Lohnscheine können in einer beliebigen Reihenfolge einlaufen. Mit Hilfe eines Sortierprogramms sollen sie dann in der Reihenfolge steigender Kontrollnummern auf ein anderes Magnetband übertragen werden. Die Kontrollnummer ist eine Kennzeichnung jedes einzelnen Werktätigen im Betrieb.

Auf einem anderen Magnetband sollen alle Personalien, die zur Berechnung des Lohnes erforderlich sind, wie Name, Kontrollnummer, Lohngruppe und Steuerklasse, gespeichert sein. Dieses Magnetband möge Arbeitskräftestammband heißen. Seine Angaben seien ebenfalls nach steigenden Kontrollnummern sortiert.

Der Berechnungsvorgang laufe nun so ab, dass das Lohnscheinband und das Arbeitskräftestammband parallel zueinander einlaufen und so alle Daten des betreffenden Falles laut programmierten Anweisungen ausgewertet werden können. Die Lohnsteuertabelle soll sich dabei als Bestandteil des Programms im Hauptspeicher der Anlage befinden.

Schließlich verbleibt als letzter Schritt der Problemaufbereitung die genaue Darstellung des eigentlichen Berechnungsalgorithmus und die Übersetzung desselben ins Maschinenprogramm. Da aber dazu die genaue Kenntnis der Darstellung der Daten auf den benutzten Datenträgern Lochkarte und Magnetband notwendig ist, müssen wir hier eine diesbezügliche Darstellung einflechten.

10.2 Daten auf laufendem Band

Die Lochkarte, die den Inhalt des einzelnen Lohnscheines aufnehmen soll, möge die im Bild gezeigte Gestalt haben. Das ist die uns bereits bekannte Darstellung von Daten auf der Lochkarte. Noch nicht besprochen ist dagegen die Speicherung der Information auf dem Magnetband.

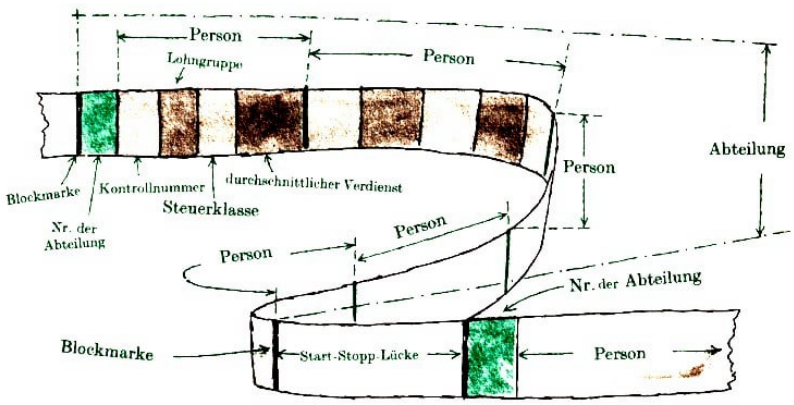
Magnetbänder haben stattliche Längen bis zu 750 m. Die Dichte der Zeilen ist ebenfalls groß: Auf einem Bandstück von 1 Millimeter Länge können etwa 20 Zeilen untergebracht werden. Demzufolge ist die Informationskapazität eines solchen Bandes recht hoch

$$750 \text{ m } : 20 \text{ Z/mm } = 750 \cdot 100 \cdot 10 \text{ mm } \cdot 20 \text{ Z/mm } = 1{,}5 \cdot 10^7 \text{ Z}$$

Eine solche Magnetbandkapazität von $1{,}5 : 10^7$ Zeichen übersteigt in der Regel die des Hauptspeichers eines Rechenautomaten. Erinnern wir uns an die gedachte Rechenmaschine GR-01, deren Hauptspeicher eine Kapazität von 10000 Wörtern zu 6 Tetraden hatte, was umgerechnet $10000 \cdot 3 = 30000 = 3 \cdot 10^4$ Zeichen ausmacht, so sehen wir, dass die Magnetbandkapazität 500 mal größer ist als das Fassungsvermögen des Hauptspeichers.

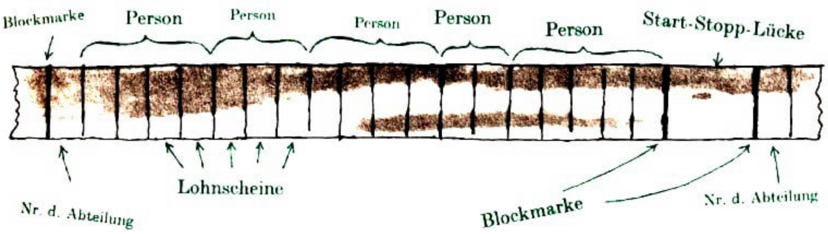
Es wäre also sinnlos, den gesamten Informationsgehalt eines Magnetbandes auf den inneren Speicher eines Automaten übertragen zu wollen.
Alle Magnetbandeinheiten geben daher ihren Inhalt nur abschnittsweise in den Hauptspeicher ein. Zwischen diesen mit Informationen beschriebenen Abschnitten muss das Magnetband anhalten. Da dies aus mechanischen Gründen nicht in der Zeit 0 erfolgen kann - wegen der Trägheit des Bandes und der Spule läuft das Band noch eine Zeitlang mit sich vermindernder Geschwindigkeit weiter, ehe es zum völligen Stillstand kommt -, entstehen auf dem Band gewisse Lücken, die man Start-Stopp-Lücken genannt hat und die nicht zur Informationsspeicherung ausgenutzt werden können.

Man muss darauf bedacht sein, mit wenig Start-Stopp-Lücken auszukommen, um die theoretisch mögliche Kapazität des Bandes nicht unnütz zu verschleudern, ohne aber auf der anderen Seite den Hauptspeicher so zu überfüllen, dass er keinen Raum für die Programme lässt.

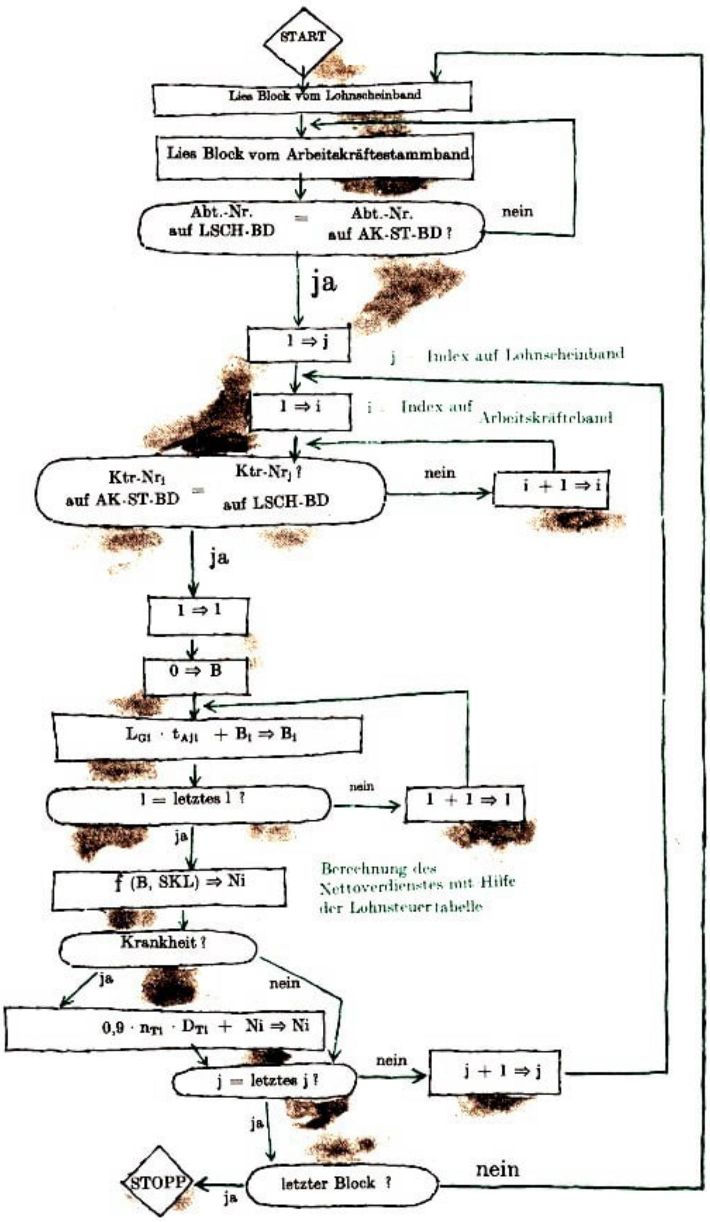


In der Praxis verfährt man so, dass man alle zu einem übergeordneten Sortierbegriff gehörenden Daten in einem Block zusammenfasst, der dann mit einem Mal in den Hauptspeicher gebracht wird, um von dort aus verarbeitet zu werden.
Im Beispiel der Lohnrechnung kann das Arbeitskräftestammband so aufgebaut werden, dass alle Angaben aller Beschäftigten einer einzelnen Abteilung zu einem Block zusammengefasst werden. Jeder Block soll durch sogenannte Blockmarken begrenzt sein. Diese Marken müssen aber echte Sonderzeichen sein und dürfen im Text des Blockes nicht auftauchen.

Das Lohnscheinband in der folgenden Abbildung könnte einen ähnlichen Aufbau besitzen, nur muss hier bedacht werden, dass unter Umständen sehr viel mehr Angaben zu einem Block gehören, da von jedem Arbeiter einer Abteilung mehrere Lohnscheine abgegeben werden.
Nach dem Eingeben eines Blockes vom Lohnscheinband und vom Arbeitskräftestammband erfolgt im abgebildeten Ablaufplan die Frage, ob die Abteilungsnummern auf beiden Bändern übereinstimmen.



Dabei ist vorausgesetzt, dass die Reihenfolge der Abteilungen auf beiden Bändern identisch ist. Es kann also nur der Fall eintreten, dass von einer Abteilung keine Lohnscheine eingetroffen sind. Bei der weiteren Durchmusterung des Arbeitskräftestammbandes muss der Automat schließlich auf die Abteilung stoßen, deren Lohnscheinangaben gerade in die Maschine gespeichert wurden.



Innerhalb der Abteilung müssen dann gleiche Kontrollnummern gesucht werden. Danach können die Lohnscheine einzeln ausgewertet werden. Am Ende liegt der gesamte Nettoverdienst N vor, der mit Anschreibung des Namens und der Kontrollnummer des

Beschäftigten ausgedruckt wird. Die Krankmeldungen sollen vorher gesammelt worden sein und nach ihrer Ablochung sortiert nach Abteilungen und Kontrollnummern im Hauptspeicher zum Abruf bereitstehen.

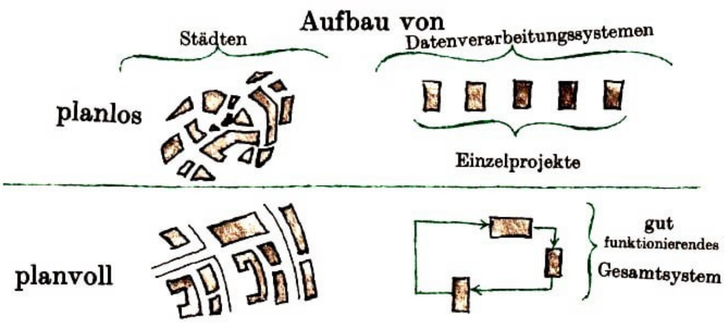
Dieser Programmablaufplan wird nun in die Maschinensprache übersetzt oder mit weniger Anstrengung in die Sprache COBOL übertragen. Nach der Anfertigung der Lochkarten können wir mit der monatlichen Lohnrechnung im Betrieb beginnen.

10.3 Planlos oder planvoll?

Nach der Übertragung der Lohnrechnung auf die Rechenanlage können wir daran gehen, weitere Probleme zu lösen. Beispielsweise könnte man als nächstes die Fragen der Arbeitskräftestatistik untersuchen. Sehr bald würde man erkennen, dass für diesen Problemkreis ein ähnliches Arbeitskräftestammband erforderlich ist, wie wir es von der Lohnrechnung her bereits kennen.
Man könnte sich dazu entschließen, die beiden Bänder der zwei Projekte zu vereinheitlichen oder mit zwei voneinander verschiedenen Bändern zu arbeiten, obwohl verschiedene Angaben auf beiden Datenträgern, wie Name und Kontrollnummer, völlig identisch sind und daher überflüssigerweise doppelt geführt werden.

Früher oder später führt der Weg der Ausarbeitung von Einzelprojekten zu der Einsicht, dass es notwendig ist, diese Einzelteile in ein Gesamtsystem der Datenverarbeitung einzubeziehen. Die Anstrengungen, die dann unternommen werden, ein solches System aus den bereits vorliegenden Bruchstücken zu schaffen, sind oft um ein Vielfaches größer als diejenigen, die zum Aufbau eines von vornherein geplanten, gründlich durchdachten einheitlichen Systems nötig wären.

Aus dieser Erkenntnis heraus bemühen sich heute viele Betriebe von Anfang an um den Aufbau integrierter Datenverarbeitungssysteme.
Der Name integrierte Datenverarbeitung soll dabei all das bezeichnen, was zu einem sinnvoll funktionierenden Komplex gut aufeinander abgestimmter Programme der Datenverarbeitungsanlage gehört, wobei diese Programme die wichtigsten Informationsbeziehungen des Betriebes widerspiegeln.



Ähnlich wie ein großzügiger, den Bedürfnissen der Einwohner Rechnung tragender Plan zum Aufbau oder zur Erweiterung einer Stadt nicht mit einem Schlage verwirklicht werden kann, wird man auch selten alle im Betrieb vorkommenden oder wichtigen Probleme gleichzeitig für die Lösung auf einer Datenverarbeitungsanlage vorbereiten können. Man muss sicherlich schrittweise vorgehen.

Die Existenz eines Planes für das zu schaffende Gesamtsystem bewahrt aber vor Verbauungen, Unstimmigkeiten und höheren Kosten, die durch Änderung »historisch gewachsener« Lösungswege entstehen. Die Stadt wie das Datenverarbeitungssystem müssen einen gut funktionierenden, lebensfähigen Organismus bilden.
Wie muss man vorgehen, wenn man dieses Ziel erreichen will?

10.4 Die Wissenschaft von den Strukturen, Modellen und Systemen

Die Antwort auf die eben gestellte Frage kommt seltsamerweise nicht aus dem Fachgebiet Datenverarbeitung oder aus den Fachgebieten, denen die zu lösenden Aufgaben angehören; sie kommt aus den Erkenntnissen einer Wissenschaft, die nicht viel älter ist als die elektronischen Rechenmaschinen und die man dennoch als deren Mutter bezeichnen kann. Diese Wissenschaft heißt Kybernetik.

Man kann sie in gewisser Weise als eine Fachdisziplin bezeichnen, die mit spezifischen Mitteln gemeinsame Strukturen in anderen Fachgebieten untersucht, Ähnlichkeiten in verschiedensten Bereichen feststellt und begründet und vor allem Aussagen über das Verhalten komplizierter Systeme macht.

Begriffe wie Modell, System, Information und Rückkopplung gehören in den Sprachschatz dieser Wissenschaft. Elektronische Rechenmaschinen werden oft als kybernetische Maschinen bezeichnet. Man spricht aber auch von kybernetischen Modellen in der Volkswirtschaft.
Tierisches Verhalten wird mit kybernetischen Methoden untersucht. Es entstehen kybernetische Modelle des bedingten Reflexes und der höheren Lernvorgänge. Selbst in den »traditionellen« Geisteswissenschaften wie der Sprachwissenschaft wird die Kybernetik mit ihren zum neuen Durchdenken altbekannter Tatsachen provozierenden Methoden längst nicht mehr als Eindringling, sondern als willkommener Helfer zur Erforschung verwickelter Zusammenhänge angesehen.

Zur Darstellung der vollen Bedeutung dieser neuen Wissenschaft muss man ein dickes Buch schreiben. Im Grunde aber ist auch das vorliegende schon ein kleiner Einblick in die Kybernetik, da es sich mit Maschinen befasst, die selbst kybernetische Strukturen realisieren.
Raum genug ist im Rahmen dieses Kapitels dagegen für eine kleine Geschichte, die über den Begründer dieser Wissenschaft erzählt wird.

Im Jahre 1909 verweigerte man dem 15jährigen Studenten der Hovard-Universität Norbert Wiener den Eintritt in die Verbindung »Phi-Beta-Kappa«. In Amerika haben viele Studentenverbindungen solche aus griechischen Buchstaben zusammengesetzte Namen, hinter denen sich der Wahlspruch der jeweiligen Organisation verbirgt.
Phi-Beta-Kappa war die Abkürzung von »Philosophia Bion Kybernetes«, was soviel wie »die Philosophie ist der Steuermann des Lebens« bedeutet.

Man muss weiter wissen, dass Norbert Wiener ein Wunderkind war. Mit 6 Jahren kam er in die dritte Klasse, die, wie sich aber bald herausstellte, seinen Fähigkeiten und Kenntnissen nicht entsprach. Noch im selben Jahr musste er in die vierte Klasse versetzt werden.
Wenig später nahm ihn sein Vater - ein Professor für neue Sprachen - aus der Schule und gab ihm Privatunterricht. Es ist leicht möglich, dass ein solches Wunderkind, das mit 18 Jahren seine Doktorarbeit schrieb, die Kränkung, die man ihm mit der Verweigerung der Aufnahme in jene Verbindung antat, nicht sobald vergessen konnte.

In der Zeit seines Privatunterrichts las er neben vielen anderen auch einen Aufsatz über Nervenströme. Als Sohn eines Professors musste Norbert Wiener die Trennung und Spezialisierung der Wissenschaften besonders merklich und störend empfunden haben. So reifte in ihm der Gedanke einer Wissenschaft, die die gemeinsamen Strukturen der Einzeldisziplinen untersuchen sollte. Er gab dieser Wissenschaft vielleicht in lächelndem Gedenken der ihm zugefügten Schmach den Namen Kybernetik. Damit beantwortete er wohl als erster in der Geschichte eine Beleidigung mit der Aufstellung einer neuen Wissenschaft.

Der Kybernetik erging es zunächst ähnlich wie dem Rotkäppchen.
Das Rotkäppchen wurde von Wölfen, die Kybernetik von den Robotern gefressen. Der höchst liebenswerte Mensch, der diese Ungeheuer schuf, war der tschechische Schriftsteller Karel Capek.
In einem Drama beschrieb er 1920 menschenähnliche Maschinen oder maschinenähnliche Menschen, die dazu verdammt waren, ständig ohne Ruhepause zu arbeiten.

Capek wandte sich damit gegen die kapitalistische Ausbeutung, die durchaus in der Lage ist, Menschen in Maschinen zu verwandeln. Bei der Übersetzung des Stückes entstand schließlich aus dem tschechischen Wort »robota« der Begriff »Roboter«.

Als die ersten Modelle kybernetischer Strukturen entstanden, hatten es die Sachunkundigen leicht, die Kybernetik als Wissenschaft von den Robotern zu bezeichnen. Gray Walter stellte seine berühmte Schildkröte vor.
Aus Laboratorien und Instituten krochen künstliche Motten, Wanzen und Mäuse; von den letzteren war die Maus von Shannon die geistig am höchsten stehende, weil sie sich in einem Labyrinth zurechtfinden konnte.

Alle diese Konstruktionen hatten den Zweck, tierisches Verhalten am technischen Modell zu veranschaulichen, sie beanspruchten keineswegs, echtes Leben zu sein; und doch brachten diese Modelle die Kybernetik in den Ruf, keine Ansätze für eine ernsthafte Wissenschaft zu enthalten.

Inzwischen hat sich diese Anschauung insbesondere unter dem Eindruck der theoretischen und praktischen Arbeiten sowjetischer Wissenschaftler grundlegend gewandelt. Die Kybernetik ist eine ernsthafte Wissenschaft geworden, deren Bedeutung für die Gesellschaft in der wissenschaftlich-technischen Revolution täglich deutlicher zu erkennen ist.

10.5 Der schwarze Kasten

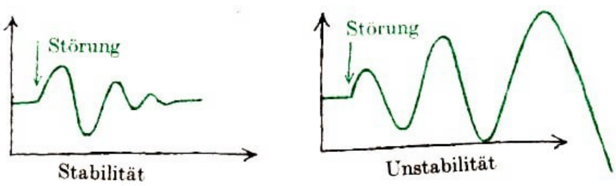
Einer der zentralen Begriffe der Kybernetik ist der Begriff des Systems. Wir wollen darunter einen funktionierenden Mechanismus im weitesten Sinne des Wortes verstehen, der durch gewisse Ein- und Ausgangsgrößen mit der übrigen Außenwelt verbunden ist. Geradezu berühmt ist die Darstellung eines Systems als schwarzer Kasten, auch blackbox genannt, in dem gewisse Eingangsgrößen (Inputs) münden und der mit bestimmten Ausgangsgrößen, die man auch Outputs nennt, auf seine Umwelt reagiert.

So gesehen ist der Mensch ein System, das sich in allen seinen inneren chemischen, physikalischen und informationstheoretischen Vorgängen manifestiert und durch die Sinnesorgane und die Gliedmaßen mit der Umwelt korrespondieren und auf diese einwirken kann.

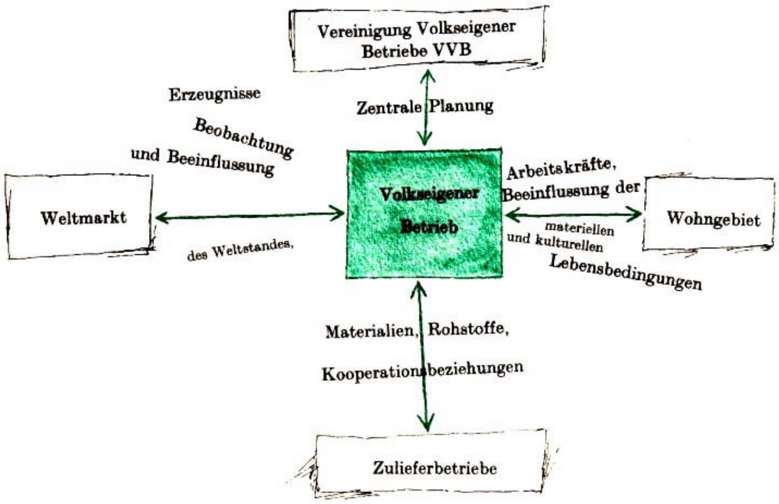
Mit genau definierten Einschränkungen dürfen kybernetische Gesetze auch bei der Betrachtung gesellschaftlicher Erscheinungen verwendet werden. Beispielsweise hat unser ökonomisches Grundkonzept in der DDR den Namen Ökonomisches System des Sozialismus erhalten.
Es stellt ein System im kybernetischen Sinne dar, dessen innere Vorgänge aus den weitverzweigten Beziehungen des gesamten volkswirtschaftlichen Reproduktionsprozesses bestehen und das nach außen durch vielfältige ökonomische, gesellschaftliche und politische Verknüpfungen besonders mit den Wirtschaftssystemen anderer sozialistischer Länder verbunden ist. Dieses System besitzt eine relativ große Eigenständigkeit.

Unsere zentralen Planungsstellen sind erfolgreich bemüht, ihm eine große Stabilität zu sichern, damit es seine Aufgabe - die Erreichung des maximalen Nationaleinkommens - jederzeit erfüllen kann. Zwei wichtige Merkmale eines Systems wollen wir festhalten:

Die relative Eigenständigkeit und die Erfüllung einer »inneren« Aufgabe. Letzteres wird im allgemeinen nur dann erreicht werden können, wenn das System stabil ist.
Der Begriff Stabilität kommt aus der Regelungstechnik, die selbst ein Teilgebiet der Kybernetik geworden ist. Ein einfaches technisches Regelsystem heißt dann stabil, wenn diejenige Größe, an deren Konstanthaltung wir interessiert sind, nach jeder beliebigen möglichen Störung von außen wieder in ihre Ruhelage zurückkehrt.
Natürlich ist die Formulierung der Stabilität eines ökonomischen Systems viel komplizierter; das technische Beispiel kann lediglich eine gewisse Grundvorstellung vermitteln.



Jedes System kann in Teilsysteme abgegrenzt werden und selbst Teilsystem eines übergeordneten Systems sein. Die Volkswirtschaft der DDR, als System aufgefasst, setzt sich aus den Teilsystemen der einzelnen Betriebe zusammen, während sie selbst gleichzeitig ein Teilsystem des Wirtschaftsgefüges aller sozialistischen Länder darstellt.



Das System »volkseigener Betrieb« als Teilsystem der sozialistischen Wirtschaft hat die Grundaufgabe der sozialistischen Ökonomie zu erfüllen:
Es muss für einen ständigen materiellen Zuwachs zum gesellschaftlichen Eigentum sorgen. Dabei darf es nicht nur schlechthin produzieren, sondern muss mit einem beschränkten Aufwand den höchstmöglichen Nutzen erreichen.

Je schneller es uns gelingt, die Produktion in der Wirtschaftseinheit Betrieb optimal zu gestalten, um so schneller wird sich unsere weitere Entwicklung zum Nutzen aller vollziehen. Bei dieser optimalen Gestaltung des betrieblichen Reproduktionsprozesses nach den Prinzipien des ökonomischen Systems des Sozialismus werden die elektronischen Datenverarbeitungsanlagen eine unschätzbare Hilfe sein.

Es ist daher folgerichtig, wenn in unserer Republik die Devise »elektronische Datenverarbeitungsanlagen in die Betriebe« gilt.
Die in der DDR entwickelten und gebauten Anlagen vom Typ R 300 werden deshalb in einer hohen Stückzahl in viele Betriebe gelangen. Hier schließt sich der Kreis unserer Gedanken: Wer verstanden hat, dass programmgesteuerte Rechenautomaten nicht nur rechnen, sondern auch entscheiden können, und wer die Fülle ihrer Anwendungsmöglichkeiten auf allen Gebieten, die mit »gewöhnlichen« Rechenarbeiten nichts zu tun haben, überflogen hat, wird den Einsatz der Anlagen zur Planung und Leitung von Betriebsprozessen nicht für unmöglich halten, sondern ihn als natürliche, den Fähigkeiten der Automaten entsprechende Aufgabe ansehen.

10.6 Die Modellierung der Zusammenhänge

Diese Aufgabe muss aber zunächst in mühseliger Arbeit genau formuliert werden. Es gilt, die wesentlichen Zusammenhänge im Betrieb zu erkennen und in mathematisch gefassten Modellen darzustellen.

Für den Prozess der unmittelbaren Produktionsvorbereitung und -kontrolle müssen zum Beispiel Modelle der optimalen Maschinenauslastung gefunden werden. Die Zusammenhänge zwischen der langfristigen, vorausschauenden Planung und der Betriebsabrechnung müssen erkannt und berücksichtigt werden. Viele Tatsachen werden in starker Abstraktion, andere wieder als wirkliche Vorgänge in das zu schaffende Gesamtmodell des Betriebes eingehen.

Diese Phase der Modellierung ist die am schwersten zu bewältigende Etappe der Verbesserung der Leitungs- und Informationsbeziehungen im Betrieb. Jeder Betrieb, der hier versäumt, moderne Verfahren der Mathematik, Kybernetik und Ökonomie zu verwenden, und sich lediglich auf eine Formulierung seiner Abrechnungsarbeiten beschränkt, wird zwar eine bedeutende Rationalisierungsmaßnahme seiner Verwaltung mit dem Rechenautomaten verwirklichen, er wird aber nie zu einer qualitativ höheren Stufe seines Leitungs- und Informationsgefüges gelangen.

Die Schwierigkeit der Aufgabe macht dabei eine echte sozialistische Gemeinschaftsarbeit aller Beteiligten notwendig. Sie verlangt Ökonomen, denen mathematische Denkgewohnheiten vertraut, und Mathematiker, denen die ökonomischen Gesetzmäßigkeiten geläufig sind. Das gemeinsame Wissen der mit der fachlichen Lösung dieser Aufgaben betrauten Kollektive wird zu einer entscheidenden Triebkraft des Gesamtvorhabens.
Aber auch dieses Wissen wird nichts ausrichten können, solange die Leitung des Betriebes nicht ebenso schnell, wenn nicht schneller die Erkenntnisse der Kybernetik beherrschen lernt und die volle Bedeutung eines wissenschaftlichen Leitungs- und Informationssystems erkennt.

Für das gesamte Betriebskollektiv, das sich auf den sinnvollen Einsatz einer Datenverarbeitungsanlage vorbereitet, gilt das viel zitierte Wort von Lenin: Lernen, lernen und nochmals lernen.
Es gilt dabei unter den Bedingungen einer sich stürmisch entwickelnden Wissenschaft, unter den Bedingungen einer ständig wachsenden Verflechtung der Wirtschaft und unter den Bedingungen eines gewaltigen Fortschritts der Technik, der uns letztlich die neuen Maschinen, die programmgesteuerten Datenverarbeitungsanlagen zur Verfügung gestellt hat.

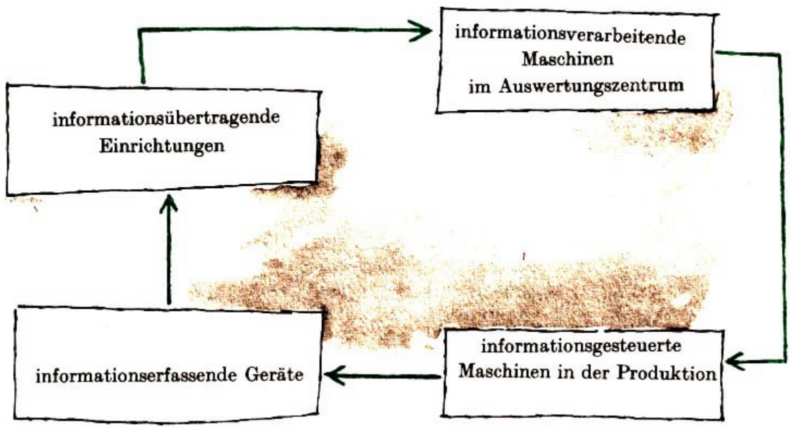
Wir müssen noch einen Augenblick in der Etappe der Modellierung der betrieblichen Vorgänge verbleiben und uns noch einmal vor Augen führen, dass es hier darauf ankommt, das Betriebsgeschehen wissenschaftlich zu durchleuchten und mathematisch zu formulieren, so dass es später möglich wird, Algorithmen der Entscheidungs-, Leitungs- und Planungsprozesse aufzustellen, die in Programme für die Datenverarbeitungsanlage umgeschrieben werden können.
Dieses Durchdringen ökonomischer Tatsachen mit mathematischen und kybernetischen Methoden ist selber wieder zu einer Wissenschaft geworden. Man nennt diese Wissenschaft Operationsforschung.
Sie beinhaltet neben der bereits bekannten Modellierung der ökonomischen Praxis die Systemanalyse, die sich darauf konzentriert, die Stabilität der modellierten Systeme zu untersuchen und zu garantieren.

Eine typische Aufgabenstellung aus der Operationsforschung ist die Optimierung von Prozessen und Situationen. Ein Betrieb, der eine gewisse Anzahl unterschiedlicher Erzeugnisse herstellt, die unterschiedliche Aufwendungen an Material und Lohnkosten beanspruchen und die auf der anderen Seite verschieden hohe Gewinne zur Folge haben, ist sicherlich daran interessiert, einen Produktionsplan aufzustellen, der ihm unter Einhaltung der vorhandenen Lohnsumme und der zur Verfügung stehenden Materialien den höchsten Nettogewinn bringt.
Solche und ähnliche Aufgaben lassen sich mit den Methoden der Operationsforschung, etwa mit der sogenannten Simplexmethode, lösen.

Nach dem Kennenlernen so vieler neuer Wissensgebiete wie der Datenverarbeitung, der Kybernetik und nun der Operationsforschung kann ein Gefühl der Verwirrung entstehen. Wir sind in diesem Kapitel von der Übertragung eines Abrechnungsvorganges auf eine elektronische Datenverarbeitungsanlage ausgegangen und inzwischen bei der Modellierung der ökonomischen Zusammenhänge im Betrieb mit Hilfe der Operationsforschung angelangt.
Aus der Kybernetik entnahmen wir den Begriff des Systems. Die Datenverarbeitungsanlage lernten wir als gehorsames Werkzeug zur Ausführung gegebener Befehle kennen. Was ist hier wichtig, was hat hier besonderen Vorrang?

An dieser Stelle kann jeder, dem die Dialektik vertraut ist, die Antwort geben: Datenverarbeitung, Kybernetik und Operationsforschung bilden eine dialektische Einheit. Die Datenverarbeitungsanlage ist die materielle Basis eines zu schaffenden Leitungs- und Informationssystems, zugleich muss sie aber auf Grund ihres programmierten Entscheidungsvermögens unmittelbares Glied in der Kette der Informationsbeziehungen sein.

Die Kybernetik lehrt, wie ein solches Leitungsgefüge als gut funktionierendes System aufzubauen ist, und die Operationsforschung liefert Modelle und Verfahren, die den Komplex des Betriebsgeschehens in algorithmischer Sprache beschreibbar machen.



Dabei dürfen wir den Begriff Datenverarbeitung nicht zu eng interpretieren. Wir wissen, dass die Rechenanlage eine informationsverarbeitende Maschine ist; wir kennen die numerisch gesteuerten Werkzeugmaschinen als informationsgesteuerte Anlagen, und wir haben uns mit einigen Möglichkeiten der Informationsübertragung beschäftigt.
Beim Entwurf von Leitungs- und Informationssystemen müssen die vorhandenen Gegebenheiten und alle erkennbaren Entwicklungstendenzen auf den Gebieten der infor-

mationsverarbeitenden, informationsgesteuerten, informationserfassenden und informationsübertragenden Geräte berücksichtigt werden.

Das bedeutet, dass man die Leitungs- und Planungsprozesse nicht isoliert von den Vorgängen in der Projektierung, Konstruktion, Technologie und in der Produktion selbst betrachten darf. Die Automatisierung und die Anwendung moderner wissenschaftlicher Verfahren muss sich auf das gesamte Betriebsgeschehen auswirken.

Der in der DDR geprägte Begriff des integrierten Systems automatischer Informationsverarbeitung (ISAIV) stützt sich auf diese Auffassung. Ein ISAIV ist das perspektivische Ziel jedes großen volkseigenen Betriebes: Wir verstehen darunter ein einheitliches System für die Planung, Leitung und Organisation, für die Vorbereitung und die Durchführung der Produktion.

Dabei darf nie der Sinn und die Aufgabe eines solchen Gesamtsystems vergessen werden: Es dient einer optimalen Gestaltung des betrieblichen Reproduktionsprozesses nach den Prinzipien, die das übergeordnete System, das ökonomische System des Sozialismus, in der DDR festlegt.
Die gesamte Informationstechnik ist also nur das materielle Hilfsmittel bei der Erreichung neuer informationsverarbeitender Systeme, die das Wirken der ökonomischen Gesetze des Sozialismus voll zur Geltung bringen müssen.

Wenn man gegenwärtig davon liest, dass in der Phase der wissenschaftlich-technischen Revolution die Wissenschaften mehr und mehr zur unmittelbaren Produktivkraft werden, so denkt man zweifellos in erster Linie an die technischen Wissenschaften. Beim Komplex der Einsatzvorbereitung der elektronischen Datenverarbeitung erweist sich dies auch für die Gesellschaftswissenschaften mit aller Deutlichkeit.
Der materielle Effekt, der Gewinn, der mit einer Rechenanlage in einem Betrieb erreichbar ist, hängt entscheidend davon ab, wie die Erkenntnisse aller Wissensgebiete, die sich hier überschneiden, effektiv genutzt werden.

10.7 Ein Automat erwacht

Was ist nun der Lohn für all die Arbeit, die mit dem Aufbau eines modernen Informationsverarbeitungssystems auf der Basis einer elektronischen Datenverarbeitungsanlage verbunden ist? Wie funktioniert ein solches System, oder wie kann es funktionieren, wenn es nach langer Zeit intensiven Bemühens fertiggestellt ist?

Im Organisations- und Rechenzentrum, der Informationszentrale des Betriebes, befindet sich als Kernstück, als unermüdlicher Roboter, die Anlage selbst. Ausgestattet mit umfangreichen Speichereinrichtungen, ist sie das »Gedächtnis« des Betriebes. Wie sieht ihr Tagewerk aus?

Der schichtführende Operateur, ihr unmittelbarer Betreuer, und die verantwortlichen Techniker, gewissermaßen die Ärzte der Maschine, überzeugen sich zunächst von ihrem ausgezeichneten Gesundheitszustand:
Ein Fehler, den der Lochstreifenleser signalisiert, wird als unwesentlich erkannt und kann in einer halben Stunde behoben sein. Das »kranke« Einzelgerät wird von der Zentraleinheit getrennt; die Reparaturarbeiten beginnen. Inzwischen hat der Operateur eine einzige gewöhnliche Lochkarte eingegeben, auf der lediglich das Datum des Tages vermerkt ist; noch kann die Anlage die Kalenderblätter nicht selbst abreißen.

Diese unscheinbare Lochkarte hat überraschende Wirkungen: Sie weckt den Automaten. Magnetbänder beginnen zu laufen, der Drucker erwacht aus seiner Geräuschlosigkeit und produziert Übersichten und Analysen. Die elektrische Schreibmaschine auf dem Bedienungspult druckt kurze Anweisungen an den Operateur zum Einlegen neuer Magnetbänder oder zur Eingabe bestimmter Kartensätze. Was ist vorgefallen?

Nach der Eingabe des Kalenderdatums hat ein zentrales Überwachungsprogramm die Liste der periodisch zu erledigenden Aufgaben durchmustert, diejenigen Programme ausgewählt, die an diesem Tage abzuarbeiten sind, und nach einer festgelegten Reihenfolge mit den Berechnungen begonnen.
Da auf jedem Programm vermerkt ist, wo die benötigten Grundinformationen zu finden sind, d. h., auf welchem Magnetband oder in welchem Kartensatz sie gespeichert sind, kann die Maschine selbständig nach diesen Datenbeständen verlangen und dies dem Operateur mitteilen.

Diese wohlgeordnete Arbeit muss das Datenverarbeitungssystem unterbrechen können, sobald besonders wichtige Nachrichten oder Gefahrenmeldungen eintreffen. Lässt die gerätetechnische Ausführung der Rechenanlage es zu, so können diese Mitteilungen wie folgt ausgewertet werden.

Über einen in der Fertigungsabteilung stehenden Fernschreiber kommt die Meldung »Auftrag 371610 kann nicht bearbeitet werden, da Material 7853926 fehlt«.
An einigen für den Betrieb besonders wichtigen Stellen, wie in der Produktionsabteilung und beim Werkdirektor, werden Geräte installiert, die den direkten Eingriff in den Ablauf der Arbeiten auf der Datenverarbeitungsanlage gestatten. Die Signale dieser Geräte gelangen auf eine ständig aktive »Beobachtungsschaltung« des Automaten und bewirken die Unterbrechung der gerade bearbeiteten Aufgabe.

In unserem Beispiel sei die eingetroffene Gefahrenmeldung dadurch ausgewertet worden, dass der Rechenautomat die Antwort »Bearbeiten Sie zunächst Auftrag 031 876. Materialien 7945628, 3050341 und 45371200« an die Produktionsabteilung gesandt hat. Danach sei die Rechenanlage zur Erledigung ihrer ursprünglichen Arbeiten zurückgekehrt.

Wie kam es zu einer solchen Antwort? Der Automat durchmusterte die Liste der Aufträge und stellte den mit der nächstfolgenden Dringlichkeit fest, suchte die dazugehörigen

Materialnummern und übermittelte schließlich das Ergebnis.
Bei der Übersendung dieses Ergebnisses kann es nicht bleiben. Wie konnte es zu dem Materialmangel kommen? Zunächst wird das Magnetband des Materialbestandes durchmustert. Unter der Nummer 7853926 findet der Automat tatsächlich die Menge 0.
Daneben findet er das Datum des Tages, an dem er eine Mitteilung an die Materialbeschaffungsabteilung ausgefertigt hatte. Diese Mitteilung war eine bereits unterschriftsreife Materialbestellung an den Herstellerbetrieb. Das möge am 6. 4. 1970 geschehen sein. Der heutige Tag sei der 29. 6. 1970. Die gespeicherte Lieferfrist betrage 4 Wochen, vom Bestelltermin an gerechnet. Hat die entsprechende Abteilung vergessen, die Bestellung abzusenden ?

Nein; auf dem Magnetband befindet sich die Bestätigung, dass die Bestellung ordnungsgemäß erfolgte. Auch die Antwort des Zulieferbetriebes, in der er die Bestellung bestätigte und die Lieferung bis zum 4. 5. 1970 zusagte, ist auf dem Magnetband gespeichert.
Was bleibt also zu tun?

Es muss eine Mahnung geschrieben werden. Dazu gibt die Anlage einen Lochstreifen aus, der die entsprechenden Angaben wie Anschrift des Betriebes, Nummer und Datum des Bestellauftrages und Art des Materials enthält. Gleichzeitig erhält der Operateur über die elektrische Schreibmaschine die Anweisung, den aufgegebenen Lochstreifen einem Schreibautomaten zuzuleiten. Hier kann ein vorgefertigtes Mahnschreiben in Form eines Lochstreifens benutzt werden, in dessen »Lücken« die Angaben des von der Rechenanlage hergestellten Streifens durch den Steuermechanismus des Schreibautomaten eingesetzt werden.
Darüber hinaus muss die Rechtsabteilung des Betriebes informiert werden. Die Datenverarbeitungsanlage muss daher eine entsprechende Mitteilung an diese Abteilung verfassen.

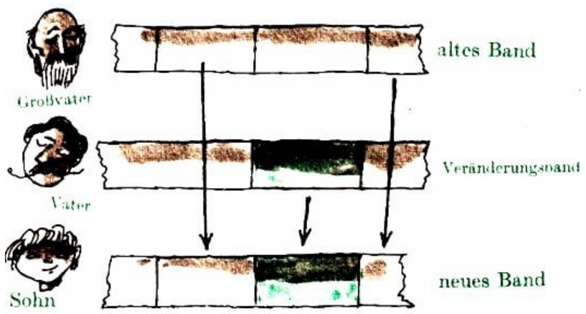
Man sieht, wie weitverzweigt die Reaktion auf eine einzige Mitteilung sein kann, und man kann ahnen, wie durch die Schaffung eines auf einer elektronischen Datenverarbeitungsanlage beruhenden Informationssystems der Betrieb gleichsam durchsichtig wird:
Alle Vorgänge werden in ihren Zusammenhängen erkannt, die Auswirkungen von Entscheidungen treten klar hervor, und die Leiter können sich auf ein fundiertes, aktuelles Datenmaterial stützen.

10.8 Vom schnellen Zugriff

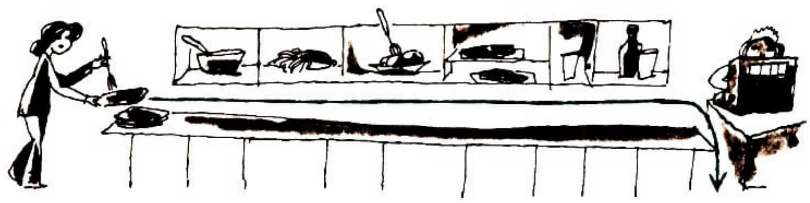
Das ständige Aufrechterhalten des wirklichen aktuellen Zustandes der Informationen ist eine der wichtigsten Aufgaben, die von der Datenverarbeitung gelöst werden müssen.
Betrachten wir einmal, wie die ständige Aktualisierung von Informationen auf dem Magnetband gehandhabt wird: Treten Änderungen gewisser Teile der gespeicherten Daten auf, so werden sie auf ein sogenanntes Veränderungsband geschrieben. Die Daten des alten Bandes werden daraufhin mit den Daten des Veränderungsbandes kombiniert und

auf ein neues Band übertragen. Die Änderung erfolgt also nicht durch unmittelbares Überschreiben oder Korrigieren des alten Bandes. Das ist eine vorausschauende Maßnahme der Datensicherung. Beim Schreiben auf das ursprüngliche Band könnten Fehler entstehen, die unter ungünstigen Umständen die Vernichtung des gesamten, sorgsam zu hütenden Datenbestandes zur Folge hätten.

Dabei muss das alte Band zumindest so lange aufbewahrt werden, bis die Richtigkeit des neuen festgestellt wurde. Es müssen also gleichzeitig mindestens drei Generationen von Magnetbändern »leben«, wobei die älteste Generation, der Großvater, erst abtreten darf, wenn die jüngste Generation, der Enkel, bereits »läuft«.

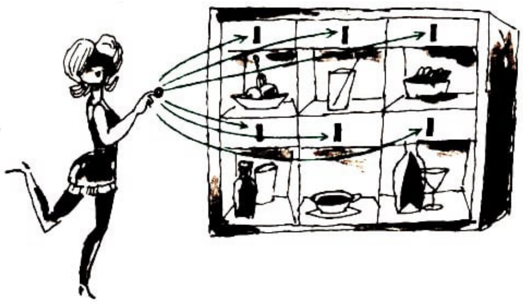


Dieses immerhin noch recht umständliche Verfahren wird vom Charakter des Magnetbandes als »sequentieller« Speicher diktiert.
Ein sequentieller Speicher oder ein »Fadenspeicher« ist dadurch gekennzeichnet, dass ein Zugriff zu einer bestimmten Stelle nur dadurch erfolgen kann, dass alle vorhergehenden Stellen durchmustert werden. In manchen Selbstbedienungsgaststätten wird uns das Prinzip des Fadenspeichers recht anschaulich demonstriert.
In einer langen Reihe sind die Gerichte nebeneinander aufgestellt; es gibt einen wohldefinierten Eingang und das Ende der Reihe mit der Kasse. Will man zu einer Tasse Kaffee gelangen, so muss man notgedrungen am Eisbein und an der Tomatensuppe vorbeigehen.



Es ist erfreulich, dass wir das Beispiel des völlig anders gearteten Speichers mit »wahlfreiem« Zugriff auch in einer Schnellgaststätte finden können, und zwar dann, wenn diese als sogenanntes Automatenrestaurant ausgebildet ist. Vor einem Schrank mit den angebotenen Genüssen stehend, kann der Besucher des Lokals beliebig zu jedem Fach greifen.
Er kann seine Tasse Kaffee direkt »ansteuern« und nach dem Geldeinwurf sofort erhalten.

Als Haupt- oder Operativspeicher eines Rechenautomaten kann natürlich nur ein Speicher mit wahlfreiem Zugriff verwendet werden, wie wir ihn in der Magnettrommel oder im Ferritkernspeicher bereits kennengelernt haben, da der Zugriff zu den einzeln adressierten Zellen solcher Speicher bedeutend weniger Zeit erfordert als das hintereinanderfolgende Durchmustern des sequentiellen Speichers.



Man kann sich nun leicht ausmalen, was für Vorteile entstehen, wenn zur Speicherung von Datenbeständen wahlfrei adressierbare äußere Speicher des Automaten verwendet werden, wie das mit dem Plattenspeicher der Fall ist. Das beim Magnetband noch recht umständliche Verändern der auf ihm gespeicherten Informationen reduziert sich beim Plattenspeicher auf das adressierbare Herausholen des gewünschten Inhaltes, das Verändern desselben und auf das Zurückschreiben auf die alte Stelle.
Mit solchen Speichern lässt sich das Prinzip der ständigen Aktualisierung von Informationen noch wirkungsvoller als mit Magnetbändern realisieren.

10.9 Die Ausnahme als Regel

Inzwischen sind in unserem Informationsverarbeitungszentrum die ökonomischen Berechnungen weitergelaufen, wobei sie mehr oder weniger »Lärm« verursachten, weil sie wiederholt auf umfangreiche Datenbestände auf den Magnetbändern oder in großen Sätzen von Lochkarten zurückgreifen mussten.
Fast unbemerkt hat sich in aller Stille im Innern der Anlage ein weiteres Ergebnis gebildet. Auf einer niedrigeren Vorrangstufe war ein technisches Problem, beispielsweise die Berechnung der Festigkeit eines Werkstückes, eingegeben worden. Die wenigen Eingabedaten und das Programm befanden sich ständig auf dem Hauptspeicher der Maschine.

Jede Sekunde, in der ökonomische Daten eingegeben wurden, nutzte das technische Programm zu seiner eigenen Abarbeitung. Dabei gelangte es schließlich zu seinen Endresultaten und meldete dies nun dem zentralen Steuerprogramm. In Abhängigkeit von der Dringlichkeit der anstehenden Aufgaben kann das Steuerprogramm die Ausgabe der technischen Ergebnisse über den Schnelldrucker anordnen oder die Überführung auf ein Magnetband organisieren, um den Druck später erfolgen zu lassen.

Wie sehen nun die Tabellen aus, die der Schnelldrucker auf Grund seiner hohen Geschwindigkeit in großer Zahl und mit beliebiger Länge anfertigen kann? Sind es endlose Zahlenkolonnen, die ein bloßes Abbild der gespeicherten Grunddaten darstellen, aus denen sich der Leiter oder der Bearbeiter einer Aufgabe die wichtigsten Angaben heraussuchen muss?

Die konventionelle Lochkartenanlage ließ keine andere Form der Tabellen zu. Bei der

elektronischen Datenverarbeitungsanlage können wir ein ganz anderes Prinzip verwirklichen. Es ist das Prinzip der Ausnahme, die damit zur Regel wird.
Niemand im Betrieb hat ein Interesse daran, einen Wald von Tabellen zu durchmustern, deren Werte gar nicht oder nur geringfügig von den geplanten Werten, also von der Norm abweichen. Dagegen ist es von ungeheurer Bedeutung, große Abweichungen und Diskrepanzen sofort zu erkennen.

Das moderne Datenverarbeitungssystem übermittelt daher dem Leiter im allgemeinen nur Informationen über Abweichungen vom Normalfall, während die übrigen Daten wohlbehütet, aber unsichtbar und jederzeit abrufbereit in den verschiedenen Speichern der Anlage verbleiben.
Man wird einsehen, dass in diesen Speichern eine wohlüberlegte Ordnung vorhanden sein muss, wenn das gespeicherte Wissen schnell bereitgestellt und schnell geändert werden soll. Will man eine solche Ordnung erreichen, so kommt man nicht umhin, die ungeheure Zahl der benötigten Daten nach einheitlichen, zweckmäßigen Gesichtspunkten abzuspeichern. In diesem Sinne spricht man dann von einer Datenbank des Betriebes.

Inzwischen habe der Drucker eine neue Tabelle geschrieben. Es soll sich um eine Übersicht für die Personalabteilung handeln, die folgende Gestalt hat.

BERICHT 308 ANF. ABT. : PB
BETRIEBSJUBILAEEN MONAT JULI
BERICHTSANFORDERUNG 28. 6. 69
AUSFERTIGUNG 29. 6. 69

NAME	VORNAME	ABT.	EINGEST.	JAHRE IM BETRIEB
SCHULZE	FRITZ	TSZ	1.7.	40
MUELLER	RENATE	OEB	1.7.	30
LEHMANN	HEINRICH	KMB	18.7.	10

Die in der Personalabteilung beschäftigten Kollegen müssen nicht mehr in verstaubten Karteien blättern, um die Namen derjenigen zu ermitteln, die im laufenden Monat ihr Betriebsjubiläum feiern. Niemand wird mehr vergessen oder übersehen, und die Zeit, die die Rechenanlage der Personalabteilung damit einspart, kann zu einer persönlichen Gratulation ausgenutzt werden.

Gehört das noch zur Datenverarbeitung? Soll die Rechenanlage am Ende Blumen verteilen? Ja, das gehört zur Datenverarbeitung: Die Entlastung der Leiter von routinemäßiger Beschäftigung gibt ihnen Gelegenheit zu wahrer schöpferischer Arbeit und Zeit zum Gespräch mit ihren Mitarbeitern, Zeit für eine persönliche Aufmunterung und Muße, über alle vorgetragenen Probleme nachzudenken.

10.10 Ein Besuch im »Fräser«-Werk

Es existieren bereits viele Beispiele für die Anwendung der Datenverarbeitung in komplexen Informationsverarbeitungssystemen. In der DDR wurde im Uhrenkombinat Ruhla ein großer Erfolg durch die richtige Einschätzung der Datenverarbeitung als ein wichtiges Hilfsmittel bei der komplexen sozialistischen Rationalisierung erreicht.

Seit einiger Zeit ist das System »INWEMOS« bekannt, das ein sehr anspruchsvolles Leitungs- und Informationssystem für einen Werkzeugmaschinen produzierenden Betrieb darstellt.
Ein richtungweisendes Vorbild für den Einsatz der modernen Automatisierungstechnik in einem sozialistischen Betrieb ist im Moskauer »Fräser«-Werk entstanden. Dort wurde auf der Grundlage leistungsfähiger Rechenanlagen und der dazugehörigen Peripheriegeräte: ein automatisiertes Leitungssystem projektiert und seit dem Jahre 1968 eingeführt.

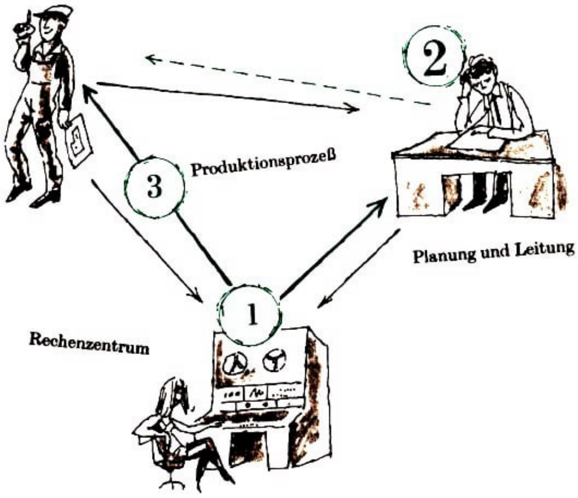
Dabei berücksichtigt man in einer ähnlichen Weise, wie es der Begriff ISAIV fordert, dass die Automatisierung auf dem Gebiet der Planung und Leitung nicht losgelöst von der Automatisierung im Produktionsprozess betrieben werden kann, wie es auch falsch wäre, moderne, hochproduktive Maschinen nur in den Werkstätten und Werkhällen einzusetzen.
Im Mittelpunkt der gesamten Informationsverarbeitung des »Fräser«- Werkes stehen die zu den Rechenanlagen gehörenden umfangreichen Speichereinrichtungen, mit deren Hilfe ein unbestechliches »Gedächtnis« des Betriebes verwirklicht ist.

Arbeitsnormen, Qualitätsmerkmale, Materialkennziffern, Maschinenzeitfonds, Planungswerte und vieles andere mehr werden auf ihnen festgehalten und laufend auf den neuesten Stand gebracht.

Das Verarbeitungssystem muss einen riesigen Informationsstrom bewältigen. Täglich fallen etwa 35 Millionen Einzelzeichen an, die die aktuelle ökonomische Situation des Betriebes beschreiben. Es zeugt von einer optimalen Organisation im Datenverarbeitungssystem, wenn trotz der großen Informationsmenge ihre Auswertung und Verwaltung in nötige Entscheidungen, Anweisungen und Maßnahmen sehr schnell erfolgt.
Schon kurz nach Beendigung einer Schicht sind die wichtigsten Produktionsdaten mit den vorgegebenen Plandaten verglichen, und bald darauf ist für jeden Arbeitsplatz bekannt, wieviel dort geleistet wurde und wieviel der einzelne Kollege verdient hat. Das hat einmal große Bedeutung für die Führung des Wettbewerbs, zum anderen werden dadurch rechtzeitig sich anbahnende Störungen im Produktionsrhythmus erkennbar und damit von vornherein vermeidbar.

Es leuchtet ein, dass die Entwicklung eines solchen automatisierten Leitungssystems nur unter aktiver Mitwirkung der Leiter entstehen kann. So hatte der Werkdirektor des Moskauer Betriebes von Anfang an Projektierung und Verwirklichung des Systems fest in der Hand. Er verstand es, durch seine persönliche Haltung das gesamte Betriebskollektiv von der Aufgabe zu begeistern. Durch umfangreiche Qualifizierungsmaßnahmen wurde jeder Werktätige auf seine künftigen Aufgaben vorbereitet.

In dieser Atmosphäre wurden die nicht geringen Schwierigkeiten beim Aufbau des Systems überwunden, und nur so konnten die schnellen und wirkungsvollen Erfolge erzielt werden, die sich vorläufig in der imponierenden Zahl von 700000 Rubel widerspiegeln, die das »Fräser«-Werk jährlich erwirtschaftet.
Damit wurde hier bewiesen, wie hochproduktiv und nutzbringend sich die Datenverarbeitungsanlagen im Betrieb erweisen können, wenn ihr Einsatz unter Beachtung der Erkenntnisse der Operationsforschung und der Kybernetik nach den Prinzipien sozialistischer Planungs- und Leitungssysteme erfolgt.



Das »Fräser«-System gleicht schon in seiner derzeitigen Realisierungsstufe einem komplizierten lebenden Organismus, der Signale und Veränderungen seiner Umwelt mit höchst sinnvollen Reaktionen beantwortet, obwohl es doch eigentlich recht »dumme« Maschinen verwendet, die lediglich Schritt für Schritt das ausführen können, was der Mensch ihnen auferlegt hat.

Man kann nun zu Recht fragen, ob das immer so bleiben muss.
Lassen sich - mit anderen Worten - nicht auch kompliziertere Maschinen ersinnen, die verwickeltere Aufgaben lösen können? Wie steht es um die Frage nach dem »elektronischen Gehirn«? Wird es eines Tages möglich sein?

11 Automaten beginnen zu lernen

11.1 Der Versuch, einen künstlichen Menschen zu bauen

Die Nachbildung von Lebensvorgängen durch Automaten reizte schon die Mechaniker des 18. Jahrhunderts.
Der geniale Erfinder Jacques de Vaucanson stellte sich das Ziel, einen künstlichen Menschen zu bauen. Schon als Schüler zeigte er großes mechanisches Geschick. Mit einem künstlichen Engel erregte er naturgemäß das Missfallen seiner geistlichen Obrigkeit und musste deswegen das Kloster, in dem er erzogen wurde, verlassen.

Nun verschrieb er sich ganz der Idee, ein Modell eines lebenden Menschen zu schaffen. Es sollte nicht nur äußerlich und in den Bewegungsformen mit dem Original übereinstimmen, sondern auch Atmung, Blutkreislauf, Verdauung und die Fähigkeit zum Sprechen aufweisen.
Da er alle seine Zeit und alle Mittel für diesen Zweck verwendete, geriet er sehr bald in eine missliche Lage. Krankheit und wirtschaftliche Not zwangen ihn, sein zu hoch gestecktes Ziel zugunsten erfüllbarer Aufgaben aufzugeben.

So baute er einen Flötenspieler und einen Schalmeienbläser. Der erste hatte ein Repertoire von 12 Musikstücken, der Schalmeienbläser konnte sogar 20 Tonstücke zu Gehör bringen. In den Salons drängte man sich. Niemand wollte glauben, dass die beiden Figuren tatsächlich in die Instrumente bliesen.

Noch größer aber war das Erstaunen, als Vaucanson eine Miniaturausführung seines Lieblingsgedankens zeigen konnte. Es war ein Enterich aus vergoldetem Kupfer, der auf einem kleinen Schränkchen stand, schnatterte, ab und zu mit den Flügeln schlug, Wasser trank und Weizenkörner pickte. Die Körner wurden in seinem Innern mittels verschiedener chemischer Stoffe »verdaut« und in überzeugend naturgetreuer Form ausgeschieden. Ein Flügel des Enterichs soll aus über 400 Einzelteilen bestanden haben.

11.2 Ein magnetisches Gehirn

Die Betrachtung lebender Organismen aus dem Gesichtswinkel heutiger Erkenntnisse hat es wünschenswert erscheinen lassen, die große Fähigkeit dieser Systeme zur Anpassung an die Umwelt und die große Sicherheit und Schnelligkeit, mit der auftretende Störungen beseitigt werden, mit technischen Einrichtungen nachzuahmen.
Einer der bekanntesten Apparate, der mit technischen Mitteln eine große Anpassungsfähigkeit demonstrieren sollte, war der Homöostat von Ashby. Der Name Homöostat

setzt sich aus den griechischen Ausdrücken für die Wörter »gleich« und »Zustand« zusammen.

Die Maschine besteht aus einer Anzahl von Elektromagneten, von denen jeder eine bestimmte Menge unterschiedlicher Schaltstellungen einnehmen kann. Die Gesamtzahl aller möglichen Zustände, in denen sich das System befinden kann, wird dadurch sehr groß. Das System kennt aber nur einige wenige Zustände, in denen es im Gleichgewicht bleiben kann, d. h. in einem Zustand, in dem sich alle Magneten in Ruhe befinden.

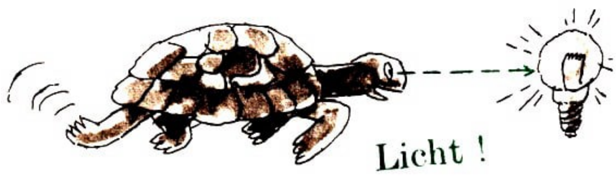
Bringt man nun einen Magneten aus seinem Ruhezustand, indem man ihn von außen verstellt, so beginnt das System sich einen neuen Gleichgewichtszustand zu suchen. Es folgt eine Serie von Schaltvorgängen bis zu dem Zeitpunkt, an dem eine neue Gleichgewichtslage, ein neuer stabiler Zustand erreicht ist.

Sicherlich kann der Homöostat nicht ernsthaft als magnetisches Gehirn bezeichnet werden. Er zeigt aber mit der Fähigkeit, stets in einen stabilen Zustand überzugehen, eine typische Eigenschaft von Lebewesen. Der Homöostat vermag sein von außen gestörtes Gleichgewicht durch eine Reihe innerer Operationen wiederherzustellen. Ein unvorhergesehenes Ereignis in der Umwelt zerstört den Organismus nicht, es hat lediglich eine Änderung verschiedener Teile desselben zur Folge und bewirkt schließlich den Übergang in einen neuen, den geänderten Umweltbedingungen angepassten stabilen Gleichgewichtszustand.

Je höher der Entwicklungsgrad des Organismus ist, desto vollkommener wird die Eigenschaft der Anpassungsfähigkeit. Beim Menschen äußert sie sich schließlich in einer bewussten Vorausschau äußerer Einflüsse und ihrer planvollen Begegnung durch wirksame Reaktionen sowie in der bewussten Veränderung seiner Umwelt und letztlich in der bewussten Gestaltung seiner selbst.

11.3 Experimente mit einer Schildkröte

Ein weiteres kybernetisches Modell, das tierisches Verhalten nachahmen sollte, ist die berühmte Schildkröte von Grey Walter. Ein dreirädriges Fahrgestell wird von einem Elektromotor angetrieben.
Ein weiterer Elektromotor verstellt das Vorderrad und dient so der Steuerung des Modells. Die Schildkröte besitzt eine Fotozelle als Tastorgan. Das innere Schaltungssystem erlaubt die Realisierung folgender Forderungen.



1. Wenn kein äußerer Reiz auf die Sinnesorgane trifft, drehen sich beide Motoren gleichzeitig. Die Schildkröte läuft auf gekrümmten Bahnen im Raum umher.

2. Wenn Licht auf die Fotozelle fällt, soll sich die Laufrichtung solange nicht ändern,

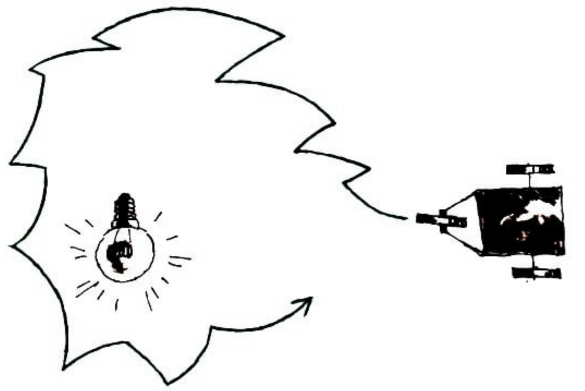
wie der Lichtreiz konstant bleibt. Nimmt er ab, so soll die Drehrichtung des steuernden Elektromotors umgekehrt werden. Das bewirkt, dass die Schildkröte auf eine Lichtquelle zuläuft.

3. Beim Berühren eines Hindernisses wird der Antriebsmotor für kurze Zeit auf Rückwärtsgang geschaltet. Auf diese Weise umgeht das Modell jedes Hindernis.

4. Ist die Lichteinwirkung zu groß, so wird ebenfalls der Rückwärtsgang eingeschaltet.

5. Die aus akustischen Reizen entstehenden, von einem Mikrofon gelieferten Spannungen sollen einen Kondensator dann aufladen, wenn gleichzeitig Licht auf die Fotozelle fällt. Wenn Ton und Licht längere Zeit auf die Schildkröte eingewirkt haben, ist der Kondensator so weit aufgeladen, dass die Spannung ausreicht, um ein Relais zu schalten, wodurch sowohl Licht als auch Ton auf die Steuerung im Sinne des Punktes 2 einwirken können. Das ist eine einfache Modellierung des bedingten Reflexes.

Diese Schildkröte verhält sich äußerlich tatsächlich wie ein primitives Tier. Schaltet man im Versuchsraum eine Lichtquelle ein, so läuft die Schildkröte »entschlossen« darauf zu, wobei sie Hindernisse umgeht.
In den Bereich zu großer Helligkeit gelangt, kriecht sie ein Stück zurück, um dann wieder vom Licht angezogen zu werden und auf diese Weise ständig die Lichtquelle zu umkreisen.
Unsere Abbildung zeigt den Weg der Schildkröte von Grey Walter.



Wesentlich verwickeltere Verhältnisse treten ein, wenn man die Schildkröte mit einer eigenen Lichtquelle versieht, die sie abschalten kann, wenn der Lichtreiz, der ihr Auge trifft, zu groß wird. Bringt man dann noch zwei solcher Schildkröten in den Versuchsraum, so wird ihr Verhalten völlig unvorhersehbar.
Die Schildkröten laufen aufeinander zu, dann schaltet eine ihr Licht aus und läuft zurück. Die andere folgt ihr. Schließlich berühren sie sich, kriechen sofort wieder in verschiedene Richtungen, schalten ihre Lampen ein und treiben viele ähnliche Späße. Sie erwecken den Eindruck gefangener Käfer in einer Schachtel.

Interessant ist, dass selbst ihr Konstrukteur, Grey Walter, nach monatelanger Beobachtungszeit nicht in der Lage war, ihr Verhalten immer richtig vorauszusagen.

Ist es angebracht, sich über diese Modelle und ihre Reaktionen zu wundern? Wenn wir

bedenken, dass sie nach eben den Konstruktionsprinzipien gebaut wurden, die diese Verhaltensweisen ermöglichen, so sehen wir sie eher etwas geringschätzig an.
Aber auch diese Betrachtungsart wird den künstlichen Tieren nicht gerecht. Wenn wir uns vor Augen halten, mit wie wenig Konstruktionsprinzipien und mit welch geringer Anzahl von Bauelementen diese verwirklichten Reaktionen hervorgerufen werden können, so liegt die Vermutung nahe, dass mit wachsender Kompliziertheit und Verflechtung der Schaltungen noch höhere Verhaltensweisen erreichbar sind.

11.4 Ein Modell des Gehirns

Besonders reizvoll ist die Simulierung von Lebensvorgängen auf digitalen Rechenautomaten, wobei als Krönung ein Modell des Gehirns angestrebt wird. Wir wissen, dass es dazu nicht genügt, eine Rechenmaschine aufzustellen und sie als Elektronenhirn zu bezeichnen. Man muss sich vielmehr um ein Programm, um einen Algorithmus bemühen, mit dessen Hilfe das Gewünschte wenigstens in grober Näherung erhalten wird.

Wir wollen zunächst nach der Speicherkapazität des Gehirns fragen. Aus psychologischen Betrachtungen weiß man, dass der Mensch in der Lage ist, sich mindestens 1000 Dinge zu merken, die in ihrer Kompliziertheit dem kleinen Einmaleins entsprechen. Da die im Einmaleins enthaltene Informationsmenge ungefähr 15000 bit beträgt, ergibt sich eine untere Abschätzung der Speicherkapazität zu

$$1000 \cdot 15000 = 1{,}5 \cdot 10^6 \text{ bit}$$

Weitere Untersuchungen haben ergeben, dass der Mensch nicht in der Lage ist, mehr als 25 bit in der Sekunde in Form von dauerhaften Eindrücken in sein Gedächtnis zu bringen. Nimmt man an, dass der Mensch 16 Stunden lang am Tag Informationen aufnimmt, so bringt er es an einem Tag auf

$$16 \cdot 60 \cdot 60 \cdot 25 \text{ bit}$$

die dauerhaft abgespeichert werden. In einem Jahr sind das dann

$$16 \cdot 60 \cdot 60 \cdot 25 \cdot 365 \text{ bit}$$

und in einem 80jährigen Leben

$$16 \cdot 60 \cdot 60 \cdot 25 \cdot 365 \cdot 80 \text{ bit}$$

Wenn man die Multiplikation ausführt, gelangt man zu der Zahl

$$4{,}2 \cdot 10^{10} \text{ bit}$$

als oberer Grenze der Speicherkapazität des Gehirns. Andere Abschätzungen ergeben 10^{12} oder sogar 10^{15} bit. Das sind ungeheuer große Zahlen.

Nur zu gut ist aber auch bekannt, dass uns von diesem riesigen Speicherinhalt in jedem Augenblick nur ein verschwindend kleiner Teil in der Größenordnung von 10^2 bit in der Sekunde zugänglich ist.

Vergleichen wir diese Zahlen, so erhalten wir hier etwa das Bild eines zehnfachen Milliardärs, der niemals mehr als einen Pfennig in der Tasche trägt.

Der »anatomische« Aufbau unseres Modellgehirns bestände also hauptsächlich aus einer großen Anzahl von Speicherzellen, von denen jeweils ein kleiner Teil besonders gekennzeichnet wäre.
Diese ausgewählten Speicherzellen sollen die augenblicklich »bewusstwerdenden« oder »angeregten« Zellen darstellen. Jede Speicherzelle beinhaltet Kombinationen von 0 und 1, über deren Sinn nichts näheres gesagt werden kann. Bestände das menschliche Gehirn aus ähnlichen Speicherzellen, so entsprächen diese Kombinationen den tatsächlichen Inhalten der Gehirnzellen.

Das Wesentliche unseres Modells besteht nun darin, dass die Beschriftungen der Speicherzellen in spezifischer Weise abgeändert werden können, wobei der Zufall eine entscheidende Rolle spielt. Des weiteren sollen in jedem Schritt nur wenige ausgewählte Speicherzellen beschriftet werden. Das sind gerade jene Zellen, die wir als die »bewusstwerdenden« angesehen haben.

Die Umwelt simulieren wir in einer gesonderten Speicherzelle, deren Inhalt sich ständig ändern soll und im wesentlichen den Eindrücken entspricht, die menschliche Gehirne über die Sinnesorgane von der Außenwelt erhalten. Zur Außenwelt zählen wir dabei die »Arme« unseres Modells, denen wir gewisse feste Wortstellen der Speicherzelle »Außenwelt« zuordnen.

Zu Beginn des »Lebens« eines solchen »Gehirns mit Armen« bestimmt lediglich ein Zufallszahlenprogramm die Reaktionen und Denkprozesse des Modells. Erst im Verlauf von »Schulungen« bildet sich ein gesetzmäßiges Verhalten heraus.
Ein Gedankenablauf in diesem Gehirnmodell hat etwa folgendes Aussehen. Man definiert zunächst ein Maß der Ähnlichkeit zwischen den Inhalten der »bewusstwerdenden« Zellen mit den gegenwärtig nicht »angeregten« Zellen.
Danach sucht man diejenigen nicht angeregten Speicherzellen, für die dieses Maß die größten Werte erreicht. Eine bestimmte Anzahl dieser Zellen wird nun als neu angeregt betrachtet, während gleichzeitig andere Zellen, die sich bis jetzt im angeregten Zustand befanden, in den nicht »bewusstwerdenden« Zustand zurückfallen.

Dabei wird für jede angeregte Zelle ermittelt, wie sie von der Summe der Inhalte aller übrigen angeregten Zellen und der Umwelt beeinflusst wird. Entsprechend dieses Einflusses wird die gerade betrachtete Zelle mit einem neuen Inhalt gefüllt, wobei wieder der Zufall mitwirken kann.

Mit diesem hier grob skizzierten Programm wurden verschiedenartige Lernversuche unternommen. Das Modell war in der Lage, nach einer gewissen Anzahl von Versuchen seine »Glieder« in eine vom Experimentator gewünschte Stellung zu bringen, wenn die Außenwelt eine vorher bestimmte Form annahm.
Auch komplizierte Versuche, in denen das Modell verschiedene festgelegte Zustände der Außenwelt unterscheiden und darauf spezifisch reagieren musste, wurden erfolgreich durchgeführt.

Bei einer flüchtigen Betrachtung solcher Modelle könnte ein Einwand gegen sie vorgebracht werden. Man vermutet, dass sie in versteckter Form die »Wenn - so« Beziehung enthalten. Es wäre doch ein leichtes, die Formel »wenn Außenwelt x, dann Reaktion y« mit wenigen Programmzeilen zu realisieren.
In Wirklichkeit enthalten die Modelle diese Beziehungen keineswegs als programmierten Bestandteil; sie entwickeln sich hier tatsächlich aus Vorgängen, die man als Lernvorgänge bezeichnen muss.

Lernprogramme, die nicht direkt auf eine Modellierung des Gehirns Bezug nehmen, sind bereits weit verbreitet. Hier sei erwähnt, wie eine Rechenmaschine eine Näherungsformel zur Berechnung der Quadratwurzel $\sqrt{x}$ selbst aufbauen kann.
Man gibt der Maschine eine Reihe von x-Werten mit den dazugehörigen $\sqrt{x}$-Werten ein und lässt sie mit Hilfe eines Zufallszahlenprogramms eine Reihe von Programmbefehlen aufstellen. Diesem gewürfelten Programm unterlegt man als Eingabewerte die vorgegebenen x-Werte und prüft nach, ob das Programm die dazugehörigen Zahlen für $\sqrt{x}$ mit einer vorgegebenen Genauigkeit berechnet.

Mit Hilfe eines Bewertungssystems werden dem von der Maschine aufgestellten Programm Noten erteilt. »Gute« Programme werden im Gedächtnis festgehalten, »schlechte« sofort wieder verworfen. Auf diese Weise gelangt die Maschine irgendwann zu einem Programm, das die gestellte Aufgabe löst.

Selbst auf sehr schnellen Automaten benötigen solche Vorgänge relativ viel Zeit, und die Programme für die Lernalgorithmen erweisen sich als recht kompliziert. So entstand die Frage, ob man Automaten bauen kann, die schneller und einfacher lernen können als die bekannten programmgesteuerten digitalen Rechner.

11.5 Der Zufall und das Lernen

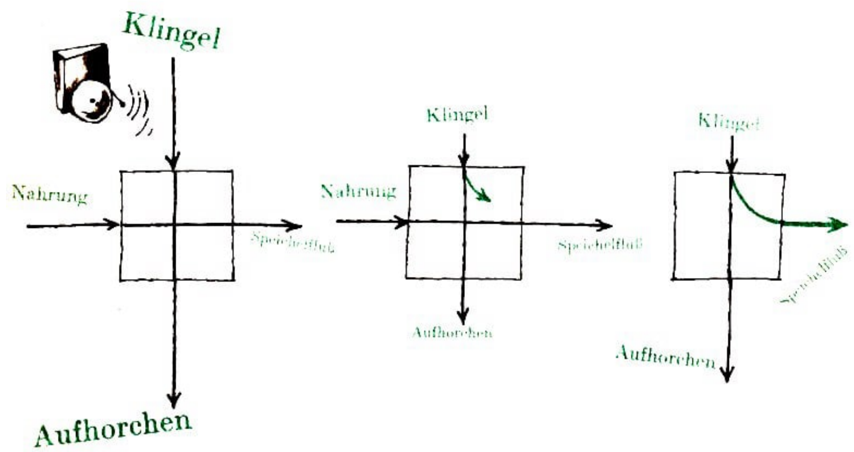
Bei der Untersuchung der in den letzten Abschnitten beschriebenen Modelle konnten wir sehen, dass sie dem Wirken von Zufällen einen breiten Raum ließen. Ihre Lernfähigkeit erhielten sie durch die Möglichkeit, zwischen vielen Zuständen zu wählen. Im Gegensatz zur Abarbeitung eines gewöhnlichen Programms, bei dem die nächste aufzurufende Speicherzelle völlig gesetzmäßig festgelegt wird, spielt bei der Auswahl der nächsten zu beschreibenden »Gehirnzelle« unseres Modells die Zufallszahl eine Rolle.

Das Modell reagiert in seiner Gesamtheit auf zwei gleiche Umweltsituationen im allgemeinen nicht mehr gleichartig. Man kann hier nur von der Wahrscheinlichkeit $W(E, A)$ sprechen, mit der die Ausgangsinformation A nach der Eingabe der Eingangsinformation E erwartet wird.
Die Sicherheit des eindeutig festgelegten Programms, das von außen nicht beeinflusst wurde, ist verlorengegangen. Dafür ist auf der Gewinnseite eine Erscheinung zu verbuchen, die bislang ausschließlich an lebende Materie gebunden schien. Es ist die Fähigkeit zum Lernen.

Bei dem Modell eines Gehirns, das wir gerade kennenlernten, wurden die mit ihm unternommenen Lernversuche als Prüfstein für seine Funktionsfähigkeit verwendet. Man kann mit Recht die Fähigkeit zum Lernen als Grundforderung an ein Modell stellen, das den Anspruch erhebt, ein Gehirn nachbilden zu wollen. Was aber ist unter dem Begriff »Lernen« zu verstehen, und ist das Lernen wirklich so streng an das Gehirn gebunden?

In den klassischen Versuchen des sowjetischen Mediziners Pawlow wurde zum ersten Mal ein erfolgreicher, über naturwissenschaftliches Gebiet führender Weg zur Erforschung der Funktionsweise des Gehirns eingeschlagen.
Bei einem Hund, dem kurz nach einem Klingelzeichen Futter gereicht wird, findet nach einer gewissen Anzahl solcher Versuche der Speichelfluss allein auf den akustischen Reiz hin statt. Pawlow fand für diesen Vorgang den Namen »bedingter Reflex«, häufig verwendete er aber auch die Bezeichnung »zeitweilige Verbindung«.

Dieser Name umschreibt die Tatsache, dass sich offensichtlich zwischen dem Klingelzeichen und dem Speichelfluss vor der Nahrungsaufnahme im Zentralnervensystem eine Kopplung gebildet hat, die deswegen zeitweilig genannt werden muss, weil sie bei ausbleibender Bekräftigung allmählich wieder erlischt.



Nimmt man die Bezeichnung »zeitweilige Verbindung« recht wörtlich, so kann man sie sich auf eine einfache Weise veranschaulichen.
In unseren drei Diagrammen ist aufgezeigt, wie das Anbieten von Nahrung Speichelfluss hervorbringt, während das Klingelzeichen ein Aufhorchen zur Folge hat. Nach einer bestimmten Anzahl von Wiederholungen dieses Vorgangs ist zwischen beiden Reizen eine zeitweilige Verbindung im Entstehen, die sich nach weiteren Versuchen schließlich ganz gefestigt hat, so dass das Klingelzeichen allein den Speichelfluss hervorruft.

Wirkt längere Zeit das Klingelzeichen ohne Bekräftigung durch die Fütterung, so bildet sich die Verbindung nach demselben Schema zurück.
Auf diesem Prinzip beruht nun eine elektronische Einrichtung, die ihr Schöpfer Karl Steinbuch »Lernmatrix« genannt hat. Die Lernmatrix ist in der Lage, zeitweilige Verbindungen zu realisieren. Sie zeigt dabei ein Verhalten, das man nur ungern anders als echtes Lernverhalten bezeichnen möchte. Ist das aber wirklich gestattet?

Bereits in den vorigen Abschnitten haben wir ohne jede kritische Bemerkung den Ausdruck Lernprogramm verwendet. So entsteht der Eindruck, dass hier ein Begriff zu schnell und vielleicht unberechtigt aus dem gewohnten Sprachbild herausgenommen und auf technische Einrichtungen übertragen wird. Wir müssen daher festlegen, was unter dem Begriff »Lernen« zu verstehen sein soll.

Gegenwärtig gibt es noch keine befriedigende Definition des Wortes Lernen. Wir müssen bedenken, dass Lernen ein äußerst vielschichtiges soziales Phänomen ist, welches historisch bestimmt und sozial determiniert ist.
Der Inhalt dessen, was wir lernen müssen, die Ziele des Lernens und die Mittel und Methoden, um diese Ziele zu erreichen, wurden schon immer durch gesellschaftliche Erfordernisse bestimmt. Die Klärung dessen, was Lernen ist und ausmacht, beschäftigt heute die Psychologen und Soziologen ebenso wie die Philosophen und Mathematiker.

Die modernen Maschinen geben dabei die Möglichkeit, bestimmte Seiten, Momente und Eigenschaften des Lernens zu simulieren, um sein Wesen tiefer zu begreifen. Im Resultat dieser Bemühungen setzte sich der Gedanke durch, die Lernprozesse in qualitativ verschiedene Gruppen einzuteilen. Als Vorstufe des Lernens sieht man das Klassifizieren an. Das ist die Fähigkeit, von einem Eingangssignal festzustellen, welcher allgemeinen Klasse von Signalen es angehört.

Die primitivste Form des Lernens ist das Auswendiglernen oder - wie man es in mehr technischer Deutung nennt - das Lernen durch Speichern.
Die nächste Stufe wird mit der Bildung zeitweiliger Verbindungen, dem Lernen durch bedingte Zuordnung erreicht.

Lernen durch Versuch und Irrtum erweist sich als nächsthöhere Stufe. Dieser Lernvorgang findet dort Verwendung, wo nicht genügend oder keine Vorerfahrungen in Bezug auf das zu Erlernende vorliegen. Das lernende System unternimmt eine Reihe von Versuchen, um sein Verhalten dem geforderten anzupassen. Die erfolgreichen Versuche werden gespeichert. Dadurch ergibt sich eine ständig verbesserte Verhaltensweise.

Mit dem Vorliegen eines Bewertungsmaßes für die erfolgreichen Versuche gerät das System in die Lage, jeweils die besten Varianten seines Verhaltens zu wählen. Diesem Vorgang soll die Bezeichnung Lernen durch Optimierung gegeben werden.
Die bisher geschilderten Lernvorgänge hatten die Tatsache gemeinsam, dass sie in einem einzigen Organismus oder Automaten abliefen. Wir kennen natürlich auch Lernsituationen, in die mehrere Organismen einbezogen sind. Der einfachste Fall besteht hier darin, dass ein lernendes System ein anderes nachahmt. Unterstützt das nachgeahmte System den Lernvorgang, belehrt es also seinen Partner, so gelangen wir zum Lernen durch Belehrung.

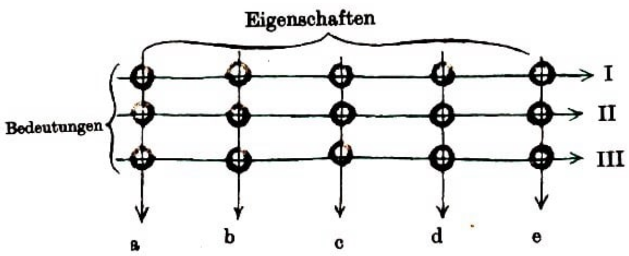
Alle »höheren« Lernvorgänge, die in der bisherigen Aufzählung nicht erschienen sind, sollen mit dem Begriff Lernen durch Erfassen belegt werden. Mit Ausnahme der letzten finden wir nun zu jeder Gruppe teilweise in ersten Anfängen, teilweise bereits deutlich ausgeprägt, Automaten als Beispiele für die entsprechenden Lernvorgänge.

Verabreden wir noch, dass wir unter »Lernen« das verstehen wollen, was in der Aufzählung erscheint, wobei das Lernen durch Erfassen ausgenommen sein soll, so haben wir uns die endgültige Berechtigung erarbeitet, von lernenden Automaten zu sprechen, ohne fürchten zu müssen, auf einen ähnlich ungenauen und verwirrenden Begriff wie den des Elektronenhirns gestoßen zu sein.

Die gegebene Erklärung befriedigt zwar noch nicht restlos. Anstelle einer exakten Definition des Lernens konnte lediglich eine Aufzählung verschiedener Lernmechanismen gebracht werden.
Sicher wird aber die Zukunft diesen Mangel abstellen, und vielleicht gelingt schon in den nächsten Jahren der Nachweis, dass einige der höheren Lernvorgänge auf einen einzigen Lernmechanismus zurückzuführen sind, womit die Liste zunächst reduziert wäre. Für diesen bevorzugten Lernmechanismus scheint das Lernen durch bedingte Zuordnungen besonders geeignet zu sein.
Dessen Realisierung in Form der Lernmatrix soll nun im Vordergrund der Betrachtungen stehen.

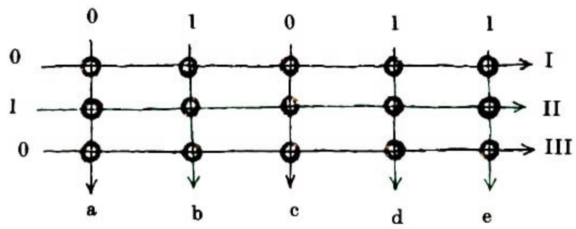
11.6 Ein lernendes Gitter

Die Gitterstruktur der folgenden Abbildung wollen wir als Lernmatrix bezeichnen. Das Bild ist eine Verallgemeinerung der Zeichnung, die den bedingten Reflex darstellen sollte. Das Gitter kann man am besten als System sich kreuzender elektrischer Leitungen auffassen, wobei in den Kreuzungspunkten besondere elektronische oder elektrochemische Vorrichtungen angebracht sind, deren elektrische Eigenschaften von den bislang durch die Leitungen geflossenen Strömen abhängen.



Mit einer solchen Lernmatrix können bedingte Verknüpfungen zwischen den als Eigenschaften und Bedeutungen gekennzeichneten Eingangssignalen hergestellt werden, wenn man gleichzeitig eine bestimmte Kombination von Eigenschaften mit einer Bedeutung zusammenwirken lässt. Formal gesehen hat dieser Vorgang eine große Ähnlichkeit mit der Einprägung eines bedingten Reflexes.

Die Eigenschaften werden in einem binären Kode angeboten, was besagen soll, dass während eines Versuches jede Eigenschaftsleitung entweder erregt oder nicht erregt sein kann. Dagegen soll nur jeweils eine Bedeutungsleitung erregt werden.



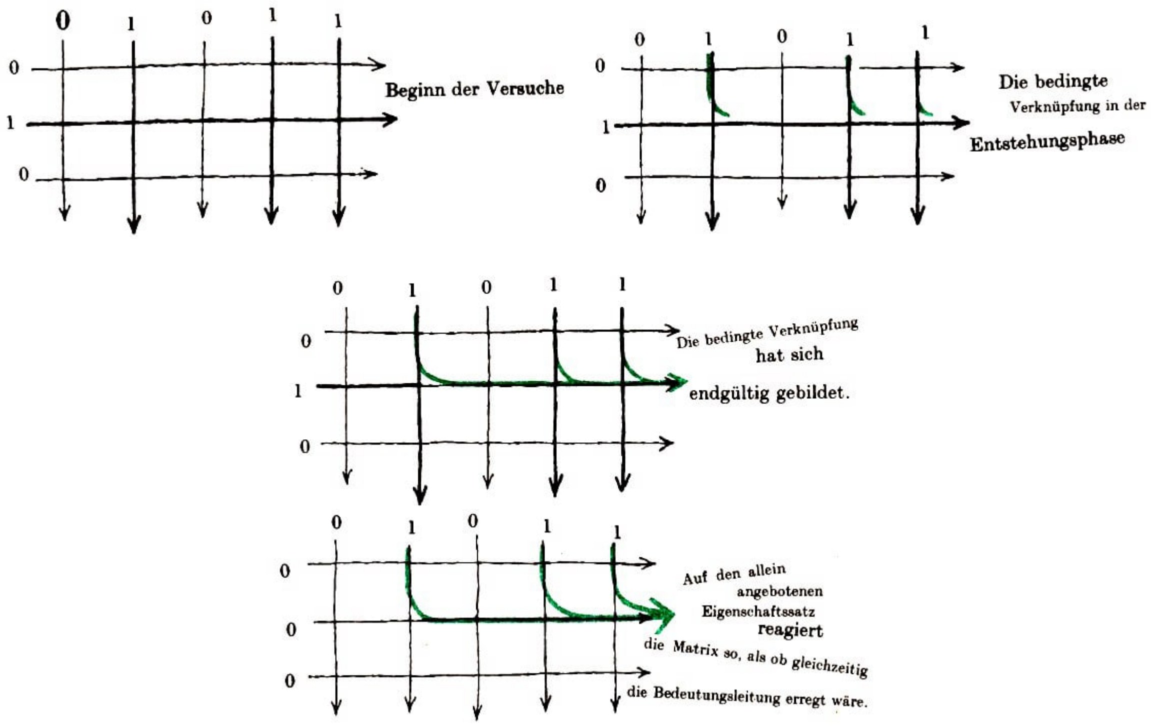
Der Lernvorgang bei der Lernmatrix besteht nun darin, dass der Matrix ein Satz von Eigenschaften mehrmals gemeinsam mit einer bestimmten Bedeutungsleitung angeboten wird. Wir wählen den Eigenschaftssatz 0, 1, 0, 1, 1 und die Bedeutungsleitung II.

Die Skizze verdeutlicht das gleichzeitige Einwirken des Eigenschaftssatzes und der Bedeutungsleitung. Dabei soll die 1 ausdrücken, dass die entsprechende Leitung erregt ist, während die 0 anzeigt, dass die Leitung nicht vom Strom durchflossen wird.
Mit wachsender Anzahl von Versuchen entwickelt sich zwischen den erregten Eigenschaftsleitungen und der erregten Bedeutungsleitung nach dem Schema des bedingten Reflexes eine bedingte Verknüpfung bei der Lernmatrix.

Ist diese Verknüpfung ausreichend gefestigt, so genügt es, wenn man nur die eingeübte Eigenschaftskombination auf die Lernmatrix einwirken lässt, um am Ausgang der Bedeutungsleitung II ein Antwortsignal zu erhalten. Der Mechanismus hat also gelernt, auf das Signal 0, 1, 0, 1,1 mit einer spezifischen Reaktion zu antworten.

Nach der Lernphase befindet sich die Matrix jetzt in der sogenannten Kannphase. Bemerkenswert ist dabei, dass die Matrix in diesem Zustand in der Lage ist, auch noch beim Ausfall einer Eigenschaftsleitung die eingeübte Bedeutung zu erkennen.
Beim gegenwärtigen »Bildungsgrad« unserer Lernmatrix ist es sogar möglich, mit einer einzigen angeregten Eigenschaftsleitung die »richtige« Bedeutung zu erhalten, weil ja nur eine Bedeutung zur Auswahl vorliegt.

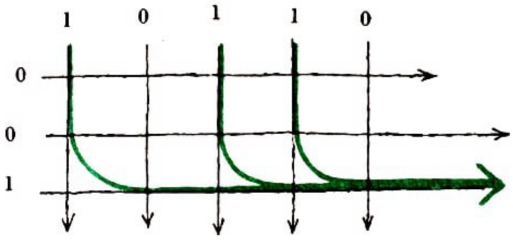
Wir wollen der Lernmatrix eine weitere Aufgabe stellen und sie veranlassen, auf die neue Eigenschaftskombination 1, 0, 1, 1, 0 mit der Erregung der Bedeutungsleitung III zu antworten. Zu diesem Zweck nehmen wir wieder eine Reihe von Versuchen vor, in



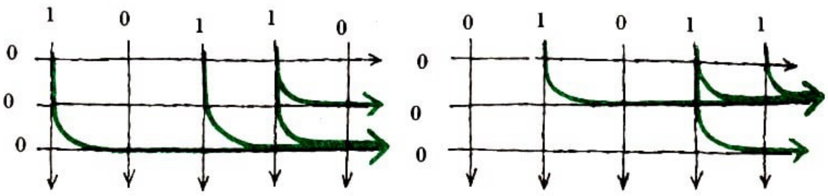
denen wir den Eigenschaftssatz 1, 0, 1, 1, 0 gleichzeitig mit der Bedeutung III eingeben. Nach einer gewissen Anzahl von Versuchen bildet sich dann auch zwischen diesem

Eigenschaftssatz und der Bedeutung III eine bedingte Verbindung heraus.

Nun ist die Lernmatrix in der Lage, auf die beiden angebotenen Eigenschaftssätze 0, 1, 0, 1, 1 und 1, 0, 1, 1, 0 mit den beiden Ausgangssignalen II und III zu antworten. In ihr hat sich eine neue bedingte Verknüpfung herausgebildet.

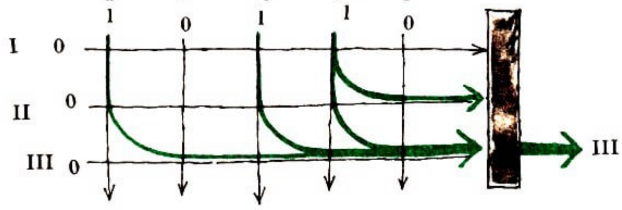


Wie man sieht, haben die beiden Eigenschaftssätze eine 1 in der Eigenschaftsleitung d gemeinsam. Da die früher einstudierte bedingte Verknüpfung noch nicht erloschen ist, wird beim Vorliegen des neuen Eigenschaftswertes 1, 0, 1, 1, 0 auch die Bedeutungsleitung II erregt, wobei das Maß dieser Erregung wesentlich kleiner als bei der Bedeutung III ist. Umgekehrt wird auch die Bedeutungsleitung III in geringem Maße erregt, wenn wir den Eigenschaftssatz 0, 1, 0, 1, 1 auf die Matrix einwirken lassen.



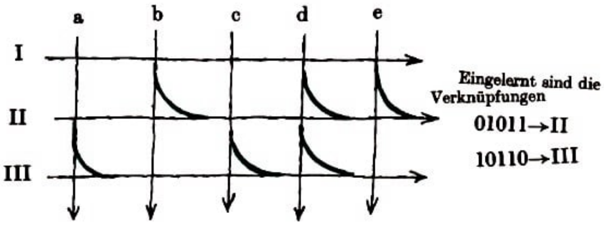
Am Ausgang der Lernmatrix erhalten wir in beiden Fällen zwei Signale, von denen jeweils nur eines wirklich gemeint war. Man kann durch den Anbau einer sogenannten Extremwertschaltung erreichen, dass nur das Signal ausgegeben wird, das am stärksten ist.

Wir deuten die Extremwertschaltung durch ein rechts von der Matrix angebrachtes Rechteck an. Mit der Anbringung der Extremwertschaltung haben wir erreicht, dass jedem angebotenen Satz von Eigenschaften nur jeweils eine Bedeutung zugeordnet wird.

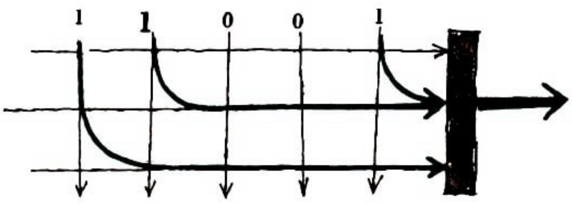


Welches Antwortsignal können wir von unserer bereits »geschulten« Lernmatrix erwarten, wenn wir keinen der beiden eingelernten Eigenschaftssätze, sondern einen neuen Eigenschaftssatz, zum Beispiel die Kombination 1, 1, 0, 0, 1, auf die Matrix einwirken lassen?

Wir zeichnen zuerst noch einmal das Bild der Lernmatrix und deuten darin die bereits vorliegenden bedingten Verknüpfungen zwischen den Eigenschaftsleitungen und den Bedeutungsleitungen durch kleine Bögen in den entsprechenden Ecken an. Wirkt nun auf diese Lernmatrix das Eingangssignal 1, 1, 0, 0, 1, so wird an allen Verknüpfungsstellen ein Strom in die entsprechenden Bedeutungsleitungen fließen.



Man sieht, dass diejenige Bedeutungsleitung am stärksten erregt wird, die für das vorliegende Eingangssignal die größte Anzahl von Durchlassstellen besitzt.
In dem vorliegenden Fall wird das stärkste Signal auf der Bedeutungsleitung II ermittelt. Die Lernmatrix verhält sich also so, als ob ihr das Eingangssignal 0, 1, 0, 1, 1 angeboten wäre. In der Tat hat dieses Signal zum zuletzt angebotenen Satz 1, 1, 0, 0, 1 eine größere »Ähnlichkeit« als das Signal 1, 0, 1, 1, 0, weil eine größere Anzahl von erregten Leitungen übereinstimmen.



Leitungen	abcde	abcde
Eingelernter Satz	01011	10110
Angebotener Satz	11001	11001
Nicht übereinstimmende Leitungen	$\underbrace{10010}_{2}$	$\underbrace{01111}_{4}$

Die Anzahl der nicht übereinstimmenden erregten Leitungen zweier auf diese Weise angebotener Signale wird ihre Hamming-Distanz genannt. Beim Vorliegen eines beliebigen Eigenschaftssatzes signalisiert die Lernmatrix also jene Bedeutung, deren eingelernter Eigenschaftssatz die geringste Hamming-Distanz zum gerade vorliegenden Eigenschaftssatz aufweist.

Dadurch gewinnt die Lernmatrix die wichtige Eigenschaft, die Bedeutung »verstümmelter« Signale noch richtig zu erkennen. Ehe wir die Tragweite dieses Satzes besprechen, wollen wir kurz auf die technische Realisierung der Lernmatrix eingehen.

11.7 Kondensatoren, die sich selbst reparieren

Wir kennen die magnetischen Ferritkerne als klassische Bausteine programmgesteuerter Rechenmaschinen. Bei diesen Kernen kam es darauf an, sie in einem einzigen Schritt von einem extremen Magnetisierungszustand in einen anderen überzuführen.

Bei der Verwendung dieser Elemente in der Lernmatrix muss die Ummagnetisierung in mehreren Einzelschritten erfolgen, wobei so viele Zwischenzustände wie nur möglich »zerstörungsfrei lesbar« sein sollen. Hier haben sich den Ferriten ähnliche magnetische Materialien wie Ultraperm oder Permolloy bewährt.

Eine weitere Realisierungsmöglichkeit der Lernmatrix ist durch die Verwendung von kapazitiven Elementen gegeben. Metallpapierkondensatoren haben die Eigenschaft, dass sie nach dem Überschreiten einer gewissen Spannung durchschlagen und die Metallfolie zerreißen. Diese Beschädigung kann aber durch die auftretende Stromwärme in einem Schweißvorgang sofort wieder rückgängig gemacht werden.

In dieser Phase verdampft ein Teil des Metallbelages, was zu einer Verringerung der Kapazität des Kondensators führt. Auf diese Weise lassen sich ebenfalls bedingte Verknüpfungen herstellen. Allerdings sind diese dann irreversibel: Eine mit diesen Elementen gebaute Lernmatrix kann nicht »umgeschult« werden, sie kann keine neuen bedingten Verknüpfungen bilden, die die alten rückgängig machen. Der Vorteil einer irreversiblen Lernmatrix besteht darin, dass sie das, was sie gelernt hat, nie wieder vergessen kann.

Große Aussichten, in lernenden Automaten Verwendung zu finden, haben auch elektrochemische Elemente, bei denen der elektrische Widerstand durch Ionentransporte geändert wird.
Die Extremwertschaltung, die Vorrichtung, die die ähnlichste Bedeutung zu einem eingegebenen Eigenschaftssatz sucht, wird am häufigsten durch eine logische Oder-Schaltung verwirklicht. Als Ergebnis erscheint dann die gewünschte Angabe der Zeile, die den größten Strom führt oder an der die größte Spannung liegt.

11.8 Die Schwerfälligkeit von Rechenautomaten

Um das neuartige Prinzip der lernenden Automaten besser würdigen zu können, wollen wir uns die Unterschiede zum programmgesteuerten Automaten vergegenwärtigen.

Während die, Eingabe eines Maschinenwortes mit einer einzigen falsch gelochten Wortstelle beim digitalen Rechenautomaten bereits katastrophale Folgen für die Rechnung hervorrufen kann, bewertet die Lernmatrix die fehlende Erregung an einer Eigenschaftsleitung als unwichtige Kleinigkeit.

Eine geschulte Lernmatrix, die bei der Eingabe eines Eigenschaftssatzes die dazugehörige Bedeutung praktisch sofort anzeigt, unterscheidet sich wesentlich von einem digitalen programmgesteuerten Rechenautomaten. Wir setzen voraus, dass auch dort alle benötigten Eigenschaftssätze bereits eingelernt sind, dass sie in das Speicherwerk des Automaten eingegeben wurden und unter bestimmten Adressen auffindbar sind. Die dazugehörigen Bedeutungen müssen ebenfalls in festgelegten Zellen abgespeichert sein.

Ist nun die Aufgabe gestellt, zu einem eingegebenen Eigenschaftssatz die dazugehörige Bedeutung zu suchen, so muss ein Vergleichsprogramm alle gespeicherten Eigenschaftssätze darauf überprüfen, ob sie mit dem gerade angebotenen Eigenschaftssatz

übereinstimmen.
Dabei kann das Programm nur schrittweise vorgehen, indem es nacheinander Satz für Satz aufruft und feststellt, ob eine Übereinstimmung mit dem eingegebenen Eigenschaftssatz vorliegt. Ist schließlich die Zelle gefunden, in der der gesuchte Eigenschaftssatz steht, so berechnet das Programm aus der Adresse dieser Zelle die Adresse der dazugehörigen Bedeutung. Diese Bedeutung kann dann gelesen und anschließend ausgegeben werden.

Die Lernmatrix arbeitet dagegen adressenlos und kommt so dem menschlichen Gehirn, bei dem das Abspeichern und das Wiederauffinden ebenfalls nicht an Adressen geknüpft sind, viel näher als die programmgesteuerten Rechenmaschinen. Natürlich ist der programmgesteuerte Automat universeller.

Es bereitet keine große Mühe, ein Programm zu schreiben, das das Verhalten der Lernmatrix nachbildet. Der Nachteil eines solchen Modells gegenüber seinem Original liegt in der längeren Zeit, die ein Lernvorgang im programmierten Modell erfordert.

An dieser Stelle lockt ein Blick auf die weitere Entwicklung. Es ist zumindest denkbar, dass in der Zukunft Automaten entstehen können, die das Prinzip der Programmsteuerung und das Prinzip der bedingten Verknüpfung in sich vereinigen. Dazu ist natürlich notwendig, dass sich die Technik der lernenden Automaten genügend weit entwickelt, so dass viele Tausende bedingter Verknüpfungen beherrschbar werden.

Die Geschichte des menschlichen Denkens lehrt, dass es keine objektive Grenze in der Erkenntnis der Natur, der Gesellschaft und des Denkens gibt, dass das, was heute als unmöglich erscheint, morgen Realität annehmen kann.
So hat die zwanzigjährige Entwicklungsgeschichte der informationsverarbeitenden Maschinen »Wunderwerke« hervorgebracht, die in ihrer Leistungsfähigkeit sicher gering erscheinen werden, wenn man sie an den Maschinen messen wird, die künftige Generationen zu entwickeln haben.

11.9 Automaten lernen lesen

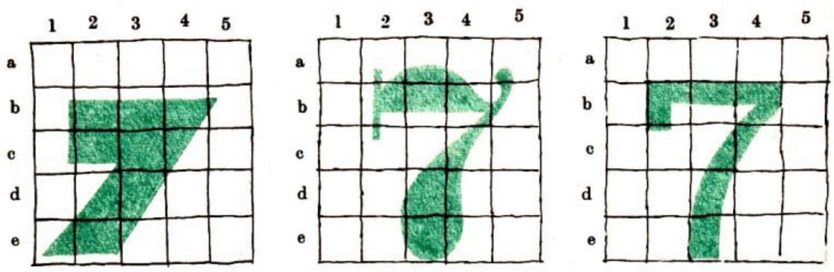
Die Vorstellungen über die Zusammenarbeit des programmgesteuerten mit dem lernenden Automaten sind für einzelne Gebiete heute bereits deutlich ausgeprägt. Bei der zunehmenden Verbreitung der Datenverarbeitung auf die verschiedensten Aufgabenbereiche wird die Eingabe der Anfangsdaten zu einem ernsthaften Problem. Der Weg über die Lochkarte erweist sich für die Zukunft als zu lang.

Auch der Behelf, spezielle, vom Automaten direkt lesbare Schriften einzuführen, wie die Magnetschrift, verspricht nicht den vollen Erfolg. Erstrebenswert bleibt das Ziel, beliebige Schriften, unter anderem auch Handschriften, dem Automaten unmittelbar einzugeben.

Diese Aufgabe ist wahrscheinlich in ihrem vollen Ausmaß mit vertretbarem technischem Aufwand ohne den Einsatz lernender Automaten nicht zu lösen. Die Lernmatrix wäre dazu bestens geeignet.

Natürlich bleiben auch hier bis zu einer endgültigen Beherrschung des Problems noch viele Fragen unbeantwortet. Eine der schwierigsten ist zum Beispiel die Frage der Zeichentrennung, die vor allem bei Handschriften auftaucht.

Am Beispiel der Entzifferung beliebig geschriebener Zahlen wollen wir die Verwendung der Lernmatrix für die automatische Zeichenerkennung etwas eingehender betrachten. Dazu sei vorausgesetzt, dass uns ein fotoelektrisches Auge mit genügend feiner Rastereinteilung zur Verfügung steht.



Der menschliche Sehapparat entziffert mühelos die im Bild gezeigten verschiedenen Darstellungen der Ziffer 7. Das Erlernen im Gehirn können wir uns so vorstellen, dass in einer großen Anzahl von Lernschritten jeweils eine geschriebene Ziffer 7 als Satz von Eigenschaften mit der Bedeutung »sieben« gekoppelt wird.

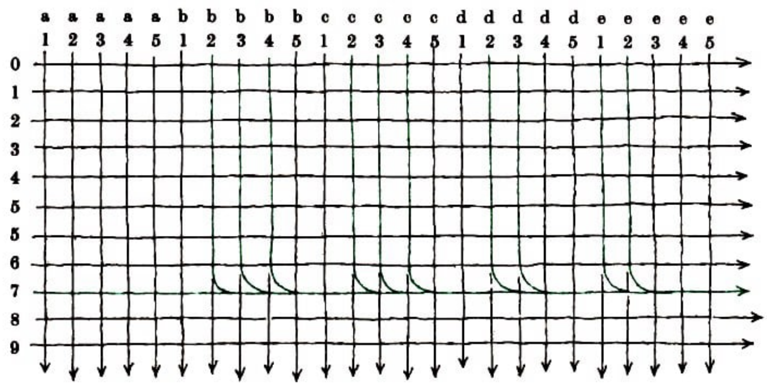
Auf dieselbe Weise vollzieht sich der Lernvorgang bei der Lernmatrix.
Wir setzen der Einfachheit halber voraus, dass alle Ziffern ungefähr in der gleichen Größe in der Mitte des Sehfeldes des elektronischen Auges erscheinen, und fordern von diesem, dass es jedem Rasterfeld einen Stromimpuls zuordnet, wenn das Rasterfeld mindestens bis zur Hälfte geschwärzt ist. Im anderen Fall soll kein Impuls ausgesandt werden.

Damit haben wir die notwendige binäre Kodierung erreicht, die bei der Verwendung der Lernmatrix vorliegen muss. Die Kodierung soll zeilenweise erfolgen. Für die erste der Ziffern 7 erhalten wir dann folgenden Satz von Dualziffern 0 und 1:

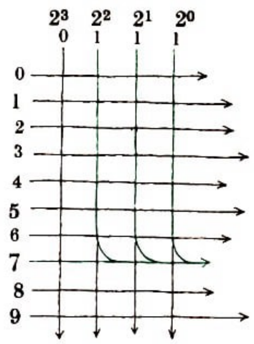
a a a a a b b b b b c c c c c d d d d d e e e e e
0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 0

Diese Zeile mit Nullen und Einsen dient uns als Eigenschaftssatz einer Lernmatrix. Durch Auswahl einer Bedeutungsleitung für die Bedeutung »sieben« sind wir in der Lage, nach Heranziehung aller denkbaren Schreibweisen der Ziffer 7 in der Lernmatrix eine Verknüpfung zwischen einer beliebig geschriebenen 7 und der dazugehörigen Bedeutung herzustellen. Auf die gleiche Art können alle Ziffern eingelernt werden.

Bis zur endgültigen Verwendung dieser so erkannten Ziffern im digitalen Rechenautomaten fehlt noch ein kleiner Schritt. Wir benötigen eine weitere Lernmatrix, die dieselben Bedeutungsleitungen wie die Matrix des »Auges« besitzt, während ihre Eigenschaftsleitungen einfach die duale Kodierung der Ziffern 0 bis 9 zulassen. In dieser Matrix hätten wir für die Ziffer 7 die Verknüpfung 0 1 1 1 einzulernen. Da die duale Schreibweise feststeht, genügt es, wenn wir hier eine Matrix verwenden, bei der das Einlernen in einem einzigen Schritt erfolgt.

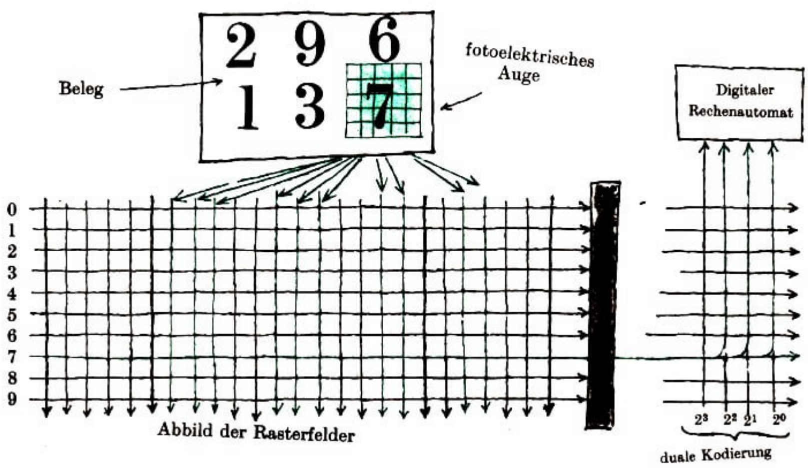


Insgesamt ergibt sich damit der Vorgang für die direkte Eingabe beliebig geschriebener Ziffern in eine digitale Rechenmaschine, so wie ihn die folgende Abbildung zeigt. Der Beleg wird vom fotoelektrischen Auge gelesen. Die Impulsfolgen der Schwarzweißverteilung gehen als Eigenschaftssätze einer belehrten Lernmatrix zu, die die entsprechenden Bedeutungsleitungen anzeigt.



Die Bedeutungen rufen in einer weiteren geschulten Lernmatrix die duale Verschlüsselung der Zahlen hervor, in der sie dann in einen programmgesteuerten digitalen Rechenautomaten eingegeben werden können.

Hierbei ist darauf zu achten, dass die zweite Lernmatrix unseres Beispiels offensichtlich etwas anders als die erste arbeitet. Wir kannten bis jetzt die Möglichkeit, der geschulten Lernmatrix einen Eigenschaftssatz anzubieten, worauf sie die dazugehörige Bedeutung signalisierte.



Umgekehrt ist es auch möglich, der Lernmatrix eine Bedeutung einzugeben, worauf sie den dazu eingelernten Eigenschaftssatz ausliefert. Diese beiden Möglichkeiten bilden die Grundlage dafür, Lernmatrizen beliebig zusammenzuschalten und auf diese Weise »höhere« Lernvorgänge zu erzielen.

Als zweites Beispiel für die Anwendung lernender Automaten sei die automatische Dokumentation erwähnt. Wir kennen das Gefühl der Hilflosigkeit, das uns in einer Bibliothek überkommt, wenn wir anstelle der Angaben von Titel und Verfasser nur wenige Stichwörter oder eine kurze, unvollständige Inhaltsangabe eines Buches vorbringen können. Selten wird ein Bibliothekar mit mehr als einem freundlichen Bedauern antworten, wenn wir das Gesuchte unter einem Berg von Büchern, die er uns als in Frage kommend bezeichnet, nicht finden können.

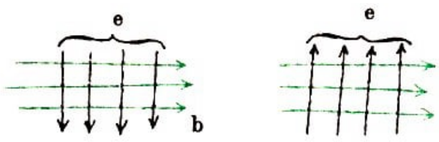
Dieses Dokumentationsverfahren, aus Stichwörtern auf bestimmte Bücher zu schließen, wird in der Tat bei der ständig wachsenden Zahl von Veröffentlichungen immer dringlicher. Die Frage »Welche Bücher gibt es, die sich mit dem Problem x beschäftigen ?« interessiert zur Zeit bereits mehr Wissenschaftler als die nach einem bestimmten Buch eines bestimmten Verfassers.

Hier können nur lernende Automaten auf die Dauer eine echte Hilfe bringen. Verwenden wir programmgesteuerte Rechenmaschinen zu diesem Zweck, so tritt uns noch einmal der grundlegende Unterschied beider Prinzipien beim Suchvorgang vor Augen. Der digitale Rechenautomat muss alle seine in Frage kommenden Speicherzellen nacheinander überprüfen, was bei umfangreichem Datenmaterial zu sehr langen Suchzeiten führen muss.
Der lernende Automat gibt dagegen sofort eine Antwort.

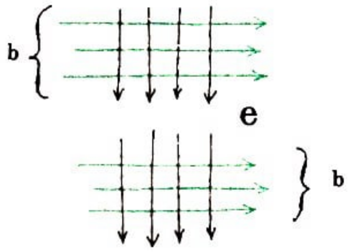
Der Lernvorgang einer Lernmatrix, die für Dokumentationszwecke eingesetzt werden soll, besteht darin, dass man mit der Katalognummer des Buches gleichzeitig einen Satz der wichtigsten darin vorkommenden Stichworte eingibt. Bei der Abfrage wird dann der Satz von Stichwörtern eingegeben, und die Lernmatrix antwortet mit der Nennung des Buches, in dem diese Stichwörter auftreten.
Durch eine Veränderung der Extremwertschaltung ist zu erreichen, dass nicht allein die ähnlichste Bedeutung, das heißt nur ein einziges Buch angezeigt wird, sondern dass mehrere Veröffentlichungen signalisiert werden, in denen die Stichwörter vorkommen. Auch können Bücher genannt werden, in denen nicht alle aufgeführten Begriffe enthalten sind.

Damit sind die Grenzen der Anwendungsmöglichkeiten der Lernmatrix längst nicht erreicht. Eine Vielfalt im einzelnen noch nicht genügend untersuchter Möglichkeiten bietet sich durch das Zusammenschalten mehrerer Lernmatrizen zu komplizierten Lernstrukturen.

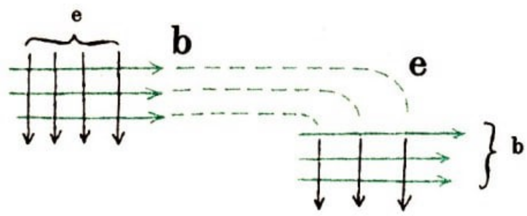
Im Beispiel des automatischen Lesens haben wir bereits eine Möglichkeit kennengelernt, Lernmatrizen zusammenzuschalten. Dort fanden wir eine Schaltung vor, in der die Bedeutungsleitungen beider Matrizen verknüpft waren. In diesem Fall spricht man von einer ebe-Schaltung, wobei der Buchstabe e die Eigenschaftssätze und b die Bedeutungen verkörpert.



Eine weitere Verknüpfung von Lernmatrizen heißt beb-Schaltung. Hier münden die Ausgänge der Eigenschaftsleitungen der ersten Matrix in die Eigenschaftsleitungen der zweiten Matrix.



Mit der Schichtung oder der ebeb-Schaltung erhalten wir schließlich eine dritte Kompositionsmöglichkeit.



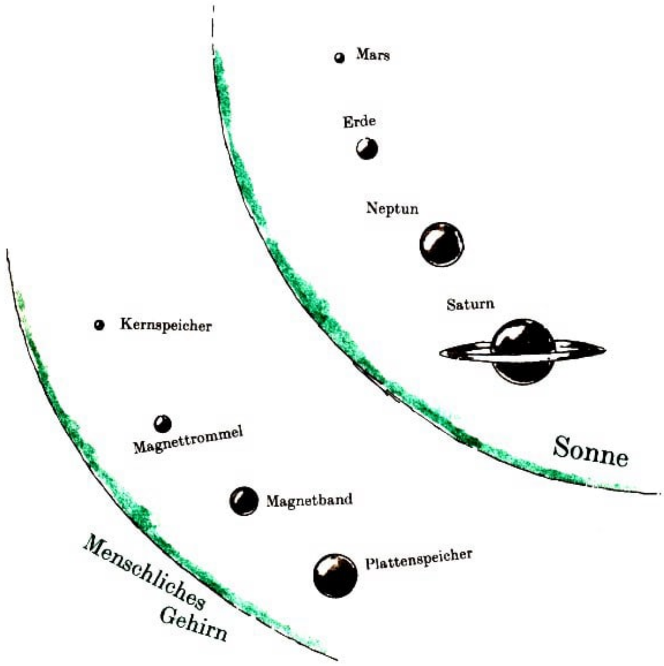
Mit diesen Grundschaltungen lassen sich dann komplexe Lernstrukturen verwirklichen, die in der Lage sind, verschiedene Abstraktionsprozesse durchzuführen, Verallgemeinerungen vorzunehmen und spezielle Ausnahmeerscheinungen zu ermitteln.

Alle diese Vorgänge kommen im Ergebnis den entsprechenden Gehirntätigkeiten sehr nahe, so dass es für die dazugehörigen Lernmechanismen in einem größeren Maße angebracht ist, von Elektronenhirnen zu sprechen, als das für die Klasse der »gewöhnlichen« programmgesteuerten Automaten der Fall war.
Die volle Berechtigung dieser Ausdrucksweise liegt sicher noch nicht vor. Im folgenden sollen einige Argumente zusammengestellt werden, die sich sowohl gegen als auch für die Möglichkeit der Existenz elektronischer Gehirne aussprechen.

11.10 Die Rakete und das Ei

Ein häufig anzutreffender Vergleich zwischen den Automaten und dem Gehirn ist die Gegenüberstellung ihrer Speicherkapazitäten.
Dieser Vergleich versucht die Unmöglichkeit oder zumindest die ungeheure Schwierigkeit der Konstruktion eines Elektronengehirns nachzuweisen: Ein Automat, der dieselbe Anzahl von Speicherzellen wie das Gehirn haben soll, müsste beim gegenwärtigen Stand der Technik die Ausmaße eines riesigen Wolkenkratzers annehmen.

Man kann die gegenwärtig existierenden Speichereinrichtungen auch als Planeten darstellen, die eine gewaltige Sonne, das Gehirn, umkreisen.



Ein solcher Vergleich mag zwar imponieren, kann aber nicht als ernsthaftes Argument gegen künstliche Gehirne gelten. Zunächst wird der Vergleich einseitig nur mit programmgesteuerten Rechenautomaten geführt.
Wie wir wissen, erfolgt die Speicherung von Informationen beim lernenden Automaten aber auf eine völlig andere Weise, die der Speicherung im Gehirn weit näher kommt.

Zum anderen hat sich schon oft gezeigt, dass Größenvergleiche bei der stürmischen Entwicklung der Technik nicht viel Aussagekraft haben. Ein ähnliches Bild war noch zu Anfang der fünfziger Jahre in Bezug auf die Raketentechnik weit verbreitet.
Darin wurde eine Rakete, die die Erdanziehung überwinden sollte, mit einem Hühnerei verglichen: Das Innere des Eies sollte die Treibstoffmenge darstellen, während der Raketenkörper durch die Schale veranschaulicht wurde.
So entstand leicht der Eindruck, dass der Bau einer solchen Rakete eine fast unmögliche Sache sei.

Als der erste künstliche Erdsatellit Sputnik I wenige Jahre später auf seine Bahn einschwenkte, erwies sich, dass ihm der Vergleich seiner Trägerrakete mit einem Ei nicht sonderlich geschadet hatte.
Ebenso gibt es keinen Beweis dafür, dass es unmöglich ist, ein Hochhaus mit elektronischen Schaltelementen auszufüllen. Man fragt sich nur, inwieweit ein derartiges Unternehmen für die Gesellschaft sinnvoll wäre. Hätte es nur den Zweck, ein elektronisches Monstrum mit den Eigenschaften des Gehirns darzustellen, so wäre es höchst unrentabel, weil ein lebendiger Mensch in einem einzigen Büroraum das Nachgebildete an qualitativer Leistungsfähigkeit übertrifft.

Wichtiger erscheint uns die Frage, ob das elektronische Hochhaus je die Eigenschaften eines Gehirns annehmen könnte. Das Gehirn stellt sich uns als derartig komplizierte

Einrichtung dar, dass wir von vornherein geneigt sind, die Frage zu verneinen. Schon der Größenvergleich mit dem Hochhaus gibt uns eine Anschauung von der ungeheuren Leistungsdichte des Gehirns.
Aber die alles entscheidende Tatsache besteht darin, dass es dem Menschen dank seines Gehirns möglich ist, seine Umwelt bewusst zu erleben und bewusst auf Umwelteinflüsse gemäß seinen Wünschen und seinem Wollen zu reagieren.

Unter Bewusstsein verstehen wir die ideelle Widerspiegelung der objektiv existierenden Realität im Gehirn des Menschen. Das Bewusstsein ist im Verlauf der Entwicklungsgeschichte irdischer Lebewesen entstanden und im Vergleich zur Materie eine sekundäre Erscheinung.

Wenn wir nun fragen, ob es elektronische Maschinen geben kann, die Bewusstsein besitzen, so müssen wir diese Frage eindeutig verneinen. Anders stellt sich das Problem, wenn man fragt, ob es jemals irgendeine, zum Teil primitive Widerspiegelung der Realität in komplizierten Automaten geben kann.

Diese Frage ist nicht ohne weiteres mit ja oder nein zu beantworten. Ein gern vorgebrachter Einwand gegen die Existenzmöglichkeit von Mechanismen, die in ihren Äußerungen Ähnlichkeit mit dem menschlichen Gehirn haben sollen, besteht in der Beschwörung der unvorstellbar großen Zahl von Eigenschaften, Fähigkeiten und Reaktionsweisen des Gehirns. Es wäre unmöglich, heißt es, dies alles in einem Rechenautomaten mit endlich vielen Speicherplätzen zu programmieren.

Dieser Einwand ist am leichtesten zu entkräften. Zunächst besitzt auch das Gehirn nur eine beschränkte Speicherkapazität. Seine hervorragende Anpassungsfähigkeit beruht vor allem auf der Tatsache, dass nicht alles starr programmiert ist.
Ohne Zweifel gibt es auch fest vorgeschriebene Handlungsabläufe im Gehirn, häufiger aber treffen wir sie in den niederen Schichten des Nervensystems.

Bei einer brütenden Gans beobachtete man die Erscheinung, dass sie alle weißen Gegenstände von der Größe ihrer Eier in der Nähe des Nestes unter ihren Körper schob. Sie »glaubte« offensichtlich, ein aus dem Nest gefallenes Ei zurückgeholt zu haben.
Die oft zitierte Dummheit der Gans kam aber noch viel deutlicher zutage, als man ihr einen solchen Gegenstand, den sie zu sich heranschieben wollte, vor ihren Augen wegnahm. Die Gans unterließ nun nicht etwa die Bewegung des Heranholens, sondern setzte sie fort, als ob das Ei noch vorhanden wäre. Erst dann setzte sie sich wieder bequem zum Brüten zurecht.

Ein einmal ausgelöster Handlungsablauf konnte hier also, obwohl er sinnlos geworden war, nicht mehr unterbrochen werden. Das entspricht der Abarbeitung eines Programms auf dem Rechenautomaten, das den Test »ist das Ei noch vorhanden ?« nicht enthält und dementsprechend nicht darauf reagieren kann.

Wollte man auf diese Weise durch Programmierung vieler Handlungsabläufe mit einer Unzahl von Testen und Möglichkeiten ein Modell eines Gehirns konstruieren, so käme man sicherlich auch mit den größten Automaten nicht aus und erreichte bestenfalls ein Modell der dummen Gans.

11.11 Automaten, die ihre Umwelt erkennen

Wir wissen, dass das Gehirnmodell des Menschen, das am Anfang dieses Kapitels beschrieben wurde, nach anderen Gesichtspunkten arbeitet. Es ist kein einziger Handlungsablauf im voraus geplant. Seine typische Verhaltensweise erwirbt sich das Modell durch eine ständige Beeinflussung durch die nicht zum eigentlichen Modell gehörende »Umwelt«.
Sicherlich reagiert das Gehirnmodell auf völlig unbekannte Umweltsituationen zunächst »falsch«. Treten sie aber häufiger auf, so lernt das Gehirn ihre richtige Einschätzung und Verarbeitung.

Auch dem Konstrukteur von Lernmatrizen sind die vielfältigen speziellen Verknüpfungen zwischen den Eigenschaften und den Bedeutungen vorher nicht bekannt. Der Automat bildet sie erst im Verlauf der Lernversuche aus.

Es erscheint sehr wahrscheinlich, dass der Weg zum Automaten, der seine Umwelt auf innere Strukturen oder Speicher abbildet, über derartige komplizierte Lernmechanismen führen wird. In einem gewissen Grad entstehen bereits heute in den Lernmodellen primitive Abbilder der Umwelt, soweit diese den Modellen zugänglich ist.

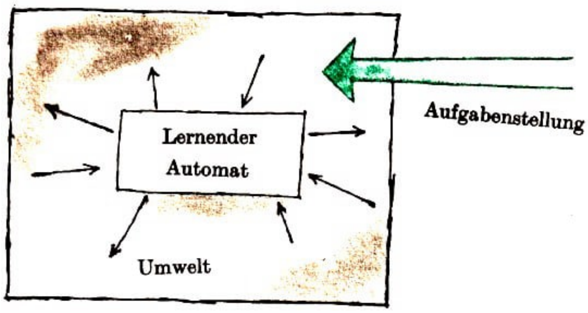
Die Zugänglichkeit zur Umwelt wirft ein weiteres Problem auf. Der klassische Digitalrechner verfügt im Vergleich zu einem Lebewesen über einen lächerlich kleinen Gesichtskreis und fast keine Möglichkeit, auf die Umwelt einzuwirken.

im Fall der Modellierung eines Gehirns auf dem Automaten ist das wesentlich anders. Hier befindet sich das Modell in einer ständigen Wechselwirkung mit der ebenfalls im Automaten erzeugten Umwelt.
Diese ändert sich dabei sowohl durch die Gesetze, die ihr innewohnen, als auch durch die Veränderungen, die das Modell an ihr vornimmt. In dieser Situation der fortwährenden gegenseitigen Beeinflussungen bilden sich am Modell jene Erscheinungen heraus, von denen man sagen muss, dass sie einem »Erkennen« der außerhalb des »Individuums« existierenden Realität sehr nahe kommen.

Ein gehirnähnlicher Automat ohne automatengerechte Umwelt wird sich nicht herstellen lassen. Hier aber sind zwei Wege möglich.
Man kann entweder einen Automaten mit einer zu ihm passenden Umwelt konstruieren oder den Automaten in die Umwelt des Menschen stellen. Die letzte Methode läuft darauf hinaus, menschenähnliche Automaten zu bauen. Es ist kaum anzunehmen, dass sie unter Verwendung rein elektronischer Bauteile zum Erfolg führen kann. Inwieweit hier die moderne Biologie eingreifen wird, ist heute nicht sicher abzuschätzen.

Sicher ist aber, dass solche wissenschaftliche Arbeiten durch die objektiven gesellschaftlichen Erfordernisse gesteuert werden und keine Liebhabereien sind. Wenn sie dennoch Überraschungen bringen können, so werden diese Erkenntnisse unsere Vorstellungen vom Menschen nur vervollkommnen, aber niemals von Grund auf ersetzen.

Eine größere Wahrscheinlichkeit besteht für die Realisierung eines Systems, das aus einem Automaten und einer ihm angepassten Umwelt besteht. Die Umwelt wäre hauptsächlich eine Fülle dem Automaten erreichbarer Informationen.



Das Gesamtsystem stellt sich dem betrachtenden Menschen wieder als einheitlicher Automat dar. Die Eingabe einer zu lösenden Aufgabe vollzieht sich dabei nicht mehr in der Form eines detaillierten Algorithmus, der streng befolgt werden muss, sondern in der weit allgemeineren Form einer bloßen Aufforderung zur Lösung der Aufgabe.
Dabei ist der Fall eingeschlossen, dass der Mensch selbst den genauen Lösungsweg nicht kennt, von dem er aber annimmt, dass er als Algorithmus mit Hilfe der dem Automaten zur Verfügung gestellten Informationen gefunden werden kann.

Eine solche Aufgabe besteht zum Beispiel in der Aufforderung, Beziehungen zwischen Festigkeitseigenschaften von Stahlsorten und den ihnen zugrunde liegenden chemischen Zusammensetzungen zu suchen. Die Umwelt des Automaten wird hierzu mit allen zur Verfügungstehenden Versuchsergebnissen, mit bisherigen Hypothesen, mit Theorien über die Chemie der Eisenverbindungen und ähnlichem ausgefüllt.

Der Automat »beschäftigt« sich dann mit diesem Gebiet, indem er Zusammenhänge feststellt, bisherige Hypothesen verwirft und neue Vermutungen aufstellt. Er verfährt dabei ähnlich wie ein Mensch, der diesen Fragenkomplex untersuchen soll, nur mit dem Unterschied, dass der Automat schneller und gründlicher arbeiten kann als der Mensch, dem es auf Grund der Vielfalt der zu untersuchenden Fakten und seiner kleinen Verarbeitungsgeschwindigkeit nicht gelingt, in angemessener Zeit neue Gesetzmäßigkeiten festzustellen.

11.12 Wer gibt die endgültige Antwort?

Bei der Betrachtung der heute bereits existierenden Gehirnmodelle auf gewöhnlichen programmgesteuerten Rechenautomaten fällt auf, in welch deutlicher Art sie das Wechselspiel zwischen dem Individuum und seiner Umwelt zeigen. Berechtigt das zu der Annahme, dass es in Zukunft Automaten geben wird, die ihre Umwelt in einem bestimmten Sinne bewusst erfassen lernen und die auf sie bewusst reagieren können?

In der sozialistischen Gesellschaftsordnung ist die humanistische Idee Wirklichkeit geworden, dass die Wissenschaft für den Menschen betrieben wird, dass sie dem Menschen zur maximalen Befriedigung seiner Bedürfnisse dient.
Es lässt sich heute kein genaues Modell davon entwerfen, welche menschlichen Bedürfnisse es in ferner Zukunft mit den Mitteln der Wissenschaft zu befriedigen gilt. Diese Fragen werden an bestimmten historischen Punkten durch die gesellschaftliche Praxis

gestellt. Die Menschheitsgeschichte bestätigt aber den Gedanken, dass der Mensch die Dinge und Erscheinungen nicht in ihrer Totalität, nicht in absoluter Vollständigkeit erkennen kann, sondern nur in einer durch die gesellschaftlichen Bedürfnisse bestimmten Näherung.

Der Mensch kann weder einen Apfel noch ein Atom, weder die menschliche Gesellschaft noch sich selbst in der Totalität erkennen, sondern nur so viel und so weit, wie es die gesellschaftliche Praxis fordert und die historisch gegebenen Bedingungen erlauben.
Demzufolge wäre auch der »denkende« Automat eine Schöpfung menschlichen Geistes, die aus gesellschaftlichen Bedürfnissen resultiert, und sie wäre immer nur eine Annäherung an die Wirklichkeit, eine angenäherte Wiedergabe der Funktionsweise des menschlichen Gehirns.

Niemand hegt einen Zweifel daran, dass die komplizierten Prozesse des Denkvorgangs, ihre historische Bestimmtheit und soziale Determiniertheit erkannt und simuliert werden können. Das aber kann nur in einem Prozess der Annäherung vor sich gehen, ohne dass jemals das menschliche Bewusstsein mit all seinen Eigenschaften technisch erzeugbar wird.

Das, was der »höhere« Automat der Zukunft zu leisten imstande ist, besteht aus der Nachbildung bestimmter Seiten, gewisser Momente und einzelner Gesetzmäßigkeiten der bewussten Widerspiegelung, aber nicht aus dem Bewusstsein in der Totalität seiner Beziehungen, das als Resultat einer andauernden Entwicklung der Materie, als Resultat einer mehrtausendjährigen Geschichte der menschlichen Gesellschaft entstanden ist.

12 Literaturhinweise

Einführung in die Datenverarbeitung

Bürger, E. und G. Wittmar: Was ist - was soll Datenverarbeitung ? Urania-Verlag, Leipzig-Jena-Berlin 1970

Fernsehkurs des Deutschen Fernsehfunks »Elektronische Datenverarbeitung«. Verlag Die Wirtschaft, Berlin 1969/1970

Hofmann, E. und D. Schreiter: Die elektronische Datenverarbeitung. Eine programmierte Einführung. Verlag Die Wirtschaft, Berlin 1968

Paulin, G.: Kleines Lexikon der Rechentechnik und Datenverarbeitung. Reihe Automatisierungstechnik, 52. VEB Verlag Technik, Berlin 1968

Technisch-wissenschaftliche Grundlagen der Datenverarbeitung Budden, F. J.: Zahlensysteme und Computer. B. G. Teubner, Verlagsgesellschaft, Leipzig 1970

Fachwörterbuch: Begriffe und Sinnbilder der Datenverarbeitung. Verlag Die Wirtschaft, Berlin 1970

Huhn, G.: Datenverarbeitung. Grundlagen. Verlag Die Wirtschaft, Berlin 1970

Jaglom, A. M. und I. M. Jaglom: Wahrscheinlichkeit und Information. VEB Deutscher Verlag der Wissenschaften, Berlin 1967

Kalex, E. und D. Mann: Wirkungsweise, Programmierung und Anwendung von Analogrechnern. Verlag Die Wirtschaft, Berlin 1970

Kämmerer, W.: Digitale Automaten. Theorie, Struktur, Technik, Programmieren. Akademie-Verlag, Berlin 1969

Klaus, G. und H. Liebscher: Was ist - was soll Kybernetik? Urania-Verlag, Leipzig-Jena-Berlin 1970

Trachtenbrot, B. A.: Wieso können Automaten rechnen ? Eine Einführung in die logisch-mathematischen Grundlagen programmgesteuerter Rechenautomaten. VEB Deutscher Verlag der Wissenschaften, Berlin 1970

Wissensspeicher - Grundlagen der Datenverarbeitung. Verlag Die Wirtschaft, Berlin 1969

Aufbau und Wirkungsweise von Rechenautomaten

Bürger, E.: Informationsspeicher für Datenverarbeitung und Rechentechnik. Reihe Automatisierungstechnik, 93. VEB Verlag Technik, Berlin 1969

Götzke, H.: Programmgesteuerte Rechenautomaten. Grundlagen, Aufbau, Arbeitsweise, Anwendung für Digitalrechner, Datenverarbeitungsanlagen, Analogrechner und Hybridsysteme. VEB Fachbuchverlag, Leipzig 1970

Murphy, J. S.: Elektronische Ziffernrechner. VEB Verlag Technik, Berlin 1968

Schubert, M.: Digitale Kleinrechner. Reihe Automatisierungstechnik, 5. VEB Verlag Technik, Berlin 1966

Smers, H.: Das maschinelle Lochkartenverfahren. Allgemeinverständliche Einführung in einen Zweig der Datenverarbeitung. VEB Fachbuchverlag, Leipzig 1969

Sydow, A.: Elektronische Analogrechner. Reihe Automatisierungstechnik, 6. VEB Verlag Technik, Berlin 1966

Programmierung digitaler Rechenanlagen

Bachmann, K.-H.: ALGOL-Programmierung mit Variante für Robotron 300. VEB Deutscher Verlag der Wissenschaften, Berlin 1969

Bachmann, K.-H.: Programmierung für Digitalrechner. Methoden und Probleme. VEB Deutscher Verlag der Wissenschaften, Berlin 1969

Hofer, A. und G. Gerhardt: Leitfaden für grafische Ablaufdarstellungen in der Organisationsarbeit. Verlag Die Wirtschaft, Berlin 1969

Saxon, J. A.: Einführung in Cobol. Ein Leitfaden zum Selbststudium. B. G. Teubner, Verlagsgesellschaft, Leipzig 1969

Schemmel: Die algorithmische Sprache ALGEK-U. Schriftenreihe Datenverarbeitung. Verlag Die Wirtschaft, Berlin 1968

Einsatzvorbereitung der Datenverarbeitung

Datenverarbeitung - Grundlagen und Einsatzvorbereitung. Informationen für leitende Kader über die elektronische Datenverarbeitung. Staatsverlag der DDR, Berlin 1969

Datenverarbeitung - Koordinierung und Leitung. Materialien einer Konferenz der Bezirksplankommission Dresden. Staatsverlag der DDR, Berlin 1967

Feinprojekte. Zur Vorbereitung des Einsatzes elektronischer Datenverarbeitungsanlagen in sozialistischen Industriebetrieben. Verlag Die Wirtschaft, Berlin 1970

Fleischmann, Gühne und Ssykor: Umstellung der Organisation in sozialistischen Industriebetrieben und Arbeitsorganisation im ORZ. Verlag Die Wirtschaft, Berlin 1970

Großprojekt zur Vorbereitung des Einsatzes von EDVA in sozialistischen Industriebetrieben. Verlag Die Wirtschaft, Berlin 1970

Organisation und Modellierung integrierter Datenverarbeitungssysteme. Staatsverlag der DDR, Berlin 1970

Datenverarbeitung und Operationsforschung

Blumenthal, B.: Die Anwendung mathematischer Methoden in der Wirtschaft. Eine Einführung für Nichtmathematiker. B. G. Teubner, Verlagsgesellschaft, Leipzig 1968

Einführung, Programmierte, in PERT. Eine Methode zur Planung und Überwachung von Projekten. Verlag Die Wirtschaft, Berlin 1968

Göttner, R.: Was ist - was soll Operationsforschung? Urania-Verlag, Leipzig-Jena-Berlin 1970

Götzke, H.: Netzplantechnik. Theorie und Praxis. VEB Fachbuchverlag, Leipzig 1969

Operationsforschung in der sozialistischen Wirtschaft. Mit bewährten Modellen aus der Praxis. Dietz Verlag, Berlin 1969

Thamm, J.: Zur Modellierung betrieblicher Reproduktionssysteme. Verlag Die Wirtschaft, Berlin 1969