

Zeitschrift:	Verhandlungen der Schweizerischen Naturforschenden Gesellschaft. Wissenschaftlicher und administrativer Teil = Actes de la Société Helvétique des Sciences Naturelles. Partie scientifique et administrative = Atti della Società Elvetica di Scienze Naturali
Herausgeber:	Schweizerische Naturforschende Gesellschaft
Band:	144 (1964)
Artikel:	Technische Grundlagen der Rechenmaschinen
Autor:	Läuchli, Peter
DOI:	https://doi.org/10.5169/seals-90610

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 27.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Das Interessante an den Rechenmaschinen ist nun dies, *dass sie zwar mit den Zahlen und den Operationen der Mathematiker operieren*, aber dass sie sich keineswegs in einem Jenseits befinden. In ihnen kommen Zähne von Zahnrädern vor, bzw. verschiebbare Massstäbe, geschlossene oder offene Stromkreise, Stromimpulse, geschlossene oder unterbrochene Kontakte.

Es ist nun von höchstem erkenntnistheoretischem Interesse, zu untersuchen, *durch welche Mittel* die Rechenmaschinen mathematische und logische Sachverhalte darstellen; durch was sie z. B. Zahlen darstellen; durch Einsatz welcher Mittel sie mathematische und logische Operationen ausführen.

Der Philosoph hofft also, aus dem Faktum der Existenz und des Funktionierens von Rechen- und Denkmaschinen eine Menge lernen zu können. Welches ist *die sachliche Natur* logischer und mathematischer Erkenntnisse? Von was hangen die herausspringenden Resultate ab, d.h. durch was sind sie gesetzt (und damit allgemeingültig und notwendig)? In welcher Weise können die Probleme formuliert und angesetzt werden? Welches ist also das *Minimum* von *Sachvoraussetzungen*, die ich machen muss, um logische und mathematische Sachverhalte darstellen und berechnen zu können? *Calculare* heisst ursprünglich: mit Steinen spielen (*calculus*, der Stein). Über was für Rechensteine verfügen wir heute? *In was besteht die Struktur der rationalen Erkenntnis?* Das ist die Frage, die wir heute hier stellen.

2. PETER LÄUCHLI (Zürich) – *Technische Grundlagen der Rechenmaschinen.*

Anhand eines einfachen Beispiels sollen einige wesentliche Züge herausgearbeitet werden:

Berechnung der durchschnittlichen Regenmenge pro Tag über die 30 Tage eines Monats, also Bildung eines arithmetischen Mittels. Für die «Handrechnung» würde ein Arbeitsblatt vorbereitet, auf welchem Messwerte, Summe und Resultat einzutragen wären. Durchführung der Addition und Division durch 30 auf einer Tischrechenmaschine.

Der Prozess soll nun automatisiert werden, und wir stellen uns die Aufgabe, eine Maschine zu konzipieren, welche imstande ist, derartige Rechengänge automatisch durchzuführen, welche also im vorliegenden Falle die Messwerte entgegennimmt und den Mittelwert als Resultat herausgibt. Dazu braucht die Maschine folgende «Organe»:

ein *Rechenwerk*, welches Addition und Division (allgemein: die Grundoperationen) ausführt;

anstelle des Papierblattes einen *Speicher*, der in jeder Zelle eine mehrstellige Zahl aufnehmen kann;

Ein- und Ausgabemedien für die Verbindung mit dem Benutzer, z.B. Tastatur und Druckwerk.

Dem Ganzen übergeordnet muss nun noch eine *Steuerung* vorhanden sein, welche für die richtige zeitliche Folge der einzelnen Abläufe sorgt. Wenn es sich um einen universellen Automaten handeln soll, muss diese Steuerung von einem *Programm* abhängig gemacht werden können, welches für jedes Problem individuell zusammengestellt wird. Das Programm wiederum besteht aus Befehlen, von denen jeder einen elementaren Schritt, wie z. B. Rechenoperation, Transfer zum Speicher usw. auslöst. Damit dasselbe Programm jederzeit wieder verwendet werden kann (Mittelbildung am Ende jedes Monats!) muss es in einer Form festgehalten sein; derart, dass es vom Automaten mechanisch gelesen werden kann.

Eine praktisch realisierbare Möglichkeit hierzu bietet z. B. der gelochte Papierstreifen: Die einzelnen Befehle werden in einem geeigneten 0-1-Code (an jeder Stelle: Loch oder kein Loch) verschlüsselt auf den Streifen gebracht. Ein mechanischer Abtaster liest dann die Befehle und führt sie sofort aus. Mit dieser Anordnung können allerdings nur sehr primitive, nämlich sogenannte «gestreckte» Programme verarbeitet werden. (Beispiel: Mittelung nur über zwei Messwerte.)

Wenn dagegen wie im ursprünglichen Beispiel eine grosse Anzahl von Malen dieselbe Folge von Befehlen durchgespielt werden soll, dann lässt sich der Automat nur sinnvoll einsetzen, wenn dieser Befehlszyklus trotzdem nur einmal gelocht werden muss. Dies lässt sich nun offenbar so realisieren, dass man einen zweiten Streifenabtaster einbaut und den zweiten Lochstreifen zu einem endlosen Band zusammenfügt. Es braucht dann Sprungbefehle, welche die Steuerung veranlassen, nach den auf dem ersten Streifen gelochten Vorbereitungsarbeiten auf den zweiten Abtaster hinüberzuwechseln und nach Ausführung des zyklischen Teils des Programmes wieder zurück auf den ersten Streifen zu springen, wo der Abschluss der Rechnung festgehalten ist. Die Ausführung des Rücksprungbefehles muss ferner von gewissen Bedingungen abhängig gemacht werden können, da er ja z. B. in unserem Falle erst beim dreissigsten Durchlauf zur Wirkung kommen darf.

Im folgenden wird stichwortartig noch auf einzelne Tatsachen hingewiesen.

Zur *Geschichte*: Charles Babbage, Mathematikprofessor in Cambridge, England, besass erstaunlicherweise schon in der ersten Hälfte des 19. Jahrhunderts die vollständige Konzeption des Rechenautomaten. Seine Ideen eilten der Technik um 100 Jahre voraus und konnten leider damals nicht richtig verwirklicht werden.

Lochkartenmaschinen, vor allem für Sortier- und Tabellierzwecke um die Jahrhundertwende (H. Hollerith, USA).

Erste Rechenautomaten im heutigen Sinne gleichzeitig durch H. Aiken in den USA und K. Zuse in Deutschland um 1945 verwirklicht. Relais-technik, Programm auf Lochstreifen (s. obiges Beispiel). Operationszeiten ~ Sekunden.

Bald darauf ENIAC, der erste elektronische Automat, ein Monstrum mit 18000 Elektronenröhren. Operationszeiten ~ Millisekunden.

Die heutigen Maschinen sind dank der Transistorentechnik sehr kompakt und nochmals schneller geworden:

Operationszeiten ~ Mikrosekunden. Grosses Gewicht wird auf leistungsfähigen Ein- und Ausgang gelegt. Auch die Anforderungen an den Speicher hinsichtlich Kapazität und Zugriffszeit sind gestiegen: Besassen die ersten Automaten vor 20 Jahren noch mechanische Speicher von weniger als 100 Zellen, so gehören zu einem heutigen grösseren Automaten Magnetkernspeicher mit einigen 10000 Zellen, welche ohne Zeitverlust sofort erreichbar sind.

Das Prinzip, wonach intern nur mit zwei verschiedenen Werten gearbeitet wird, hat sich vor allem aus Gründen der Betriebssicherheit und der Übersichtlichkeit durchwegs erhalten. Die kompliziertesten Rechen- und Steuerschaltungen sind aus Elementen aufgebaut, welche aufgrund der an ihren Eingängen vorhandenen 0-1-Werte einen 0-1-Wert als Resultat liefern. Es hat sich dabei als zweckmäßig herausgestellt, für die Beschreibung derartiger Funktionen den Formalismus der Aussagenlogik heranzuziehen.

Ein sehr wesentlicher Zug der heutigen Konzeption des Rechenautomaten liegt schliesslich darin, dass nach einer in erster Linie von J. v. Neumann stammenden Idee das Programm im Speicher aufbewahrt wird, wodurch erst beliebig komplizierte Programmstrukturen mit Sprüngen, ineinander geschachtelten Zyklen und Verzweigungen ermöglicht werden. Das Programm kann dann sogar so angelegt werden, dass es sich selbst aufbaut und verändert.

Abschliessend einige Bemerkungen allgemeinerer Art. Wenn der Vortragende schon zu Beginn gewisse Bedenken gegen den Ausdruck «Denkmashine» angemeldet hat, so sind diese Bedenken vor allem als Ressentiments gegenüber häufig anzutreffendem Missbrauch von Wörtern wie «Elektronengehirn» und dgl. zu verstehen. In erster Linie hat man sich Klarheit darüber zu schaffen, von welcher Klasse von Maschinen man überhaupt spricht: Auf der gegenwärtigen Entwicklungsstufe ist der digitale Rechenautomat doch wohl als «endlicher Automat» anzusehen, d.h. alles, was er produziert, ist in völlig deterministischer Weise durch Anfangszustand (des Speichers) und Eingabedaten bestimmt. Ob man dann bei einer seriösen Betrachtung wirklich dem Automaten so etwas wie Denkfähigkeit zuschreiben will, ist nur eine Frage der entsprechenden Definitionen. Nur darf man nicht in den Fehler verfallen, Leistungen des Menschen, der die Maschine programmiert, in diese hineinzuprojizieren.

3. P. KIRSCHENMANN (Freiburg) – *Rationale Erkenntnis, Logik und Automaten.*

Der Ausgangspunkt für die folgende Erörterung ist die Frage: Was haben die drei Begriffe Erkenntnis, Logik und Automat miteinander zu tun? Den Grund abzuschreiten, auf dem diese Begriffe miteinander in Verbindung treten können, wäre deshalb das Vorhaben. Etwas einge-