

**Zeitschrift:** Mitteilungen / Vereinigung Schweizerischer Versicherungsmathematiker  
= Bulletin / Association des Actuaires Suisses = Bulletin / Association of  
Swiss Actuaries

**Herausgeber:** Vereinigung Schweizerischer Versicherungsmathematiker

**Band:** 59 (1959)

**Artikel:** Über die Symbolik der Programmierung programmgesteuerter  
elektronischer Rechenanlagen

**Autor:** Beneš, R.

**DOI:** <https://doi.org/10.5169/seals-966811>

### **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:** 18.04.2026

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

# Über die Symbolik der Programmierung programmgesteuerter elektronischer Rechenanlagen

Von R. Beneš, Basel

## Zusammenfassung

Die auf programmgesteuerten Rechengerten zu realisierenden Arbeitsabläufe hängen weitgehend von der Anzahl der Ein- und Ausgabegeräte und von der Grösse des Speichers ab. Auf dieser Grundlage wird der Versuch unternommen, eine allen Rechengerten gemeinsame Symbolik der Ablaufdiagramme zu entwerfen.

## Einleitung

Unter *Programmierung* versteht man die Planung geordneter Arbeitsabläufe und deren Übertragung in eine Folge von Maschineninstruktionen. Der eigentlichen *Verschlüsselung* hat also stets eine *Formalisierung* des operativen Prozesses voranzugehen. Bei der Bearbeitung mathematischer Probleme findet der Programmierer im allgemeinen die zur Lösung geeigneten Rechenvorschriften vor. Im kommerziellen und verwaltungstechnischen Bereich dagegen hat eine solche Formalisierung der Arbeitsabläufe kaum stattgefunden. Es war Bruno Thuring, der als erster die Forderung formulierte, dass das sogenannte *Ablauf-* oder *Strukturdiagramm* unabhängig von einer bestimmten Rechenanlage zum definierenden Mittel eines Arbeitsablaufs werden sollte<sup>1)</sup>. Aus dieser weitgehenden Abstraktion vom Gerät entwickelte sich eine *Logik der Programmierung*, welche etwa an Fragestellungen folgender Art heranging: Gegeben sei eine Folge von  $n$  Zahlen. Enthält diese Folge die Zahl  $a$ ? Formulieren wir den Weg, der von den gegebenen Grössen zu den gewünschten Resultaten führen soll, so erhalten wir einen genau umschreibbaren *Problemkern*, der unabhängig von den technischen Eigenschaften und der Befehlsgebung einer bestimmten

<sup>1)</sup> B. Thuring: «Die Logik der Programmierung.» Robert Göller-Verlag, Baden-Baden 1957.

Maschine dargestellt werden kann. Betrachten wir hingegen den Arbeitsablauf als Ganzes, die Eingabe, den Fluss der Informationen, ihre Bearbeitung und die Ausgabe der Ergebnisse, so werden wir leicht dessen Abhängigkeit von der Anzahl der Ein- und Ausgabegeräte sowie von der Grösse des Speichers feststellen.

Diese Abhängigkeit vom Aufbau der Rechenmaschine berechtigt aber keineswegs zur Annahme, dass es erst möglich sei, festzustellen, wie eine bestimmte Aufgabe auf einer Datenverarbeitungsanlage zu lösen sei, wenn man die Eigenschaften der Anlage sehr gut kenne <sup>1)</sup>. Diese weitverbreitete Meinung betrifft nur die letzte Phase der Programmierung, die Befehlsgebung; sie hat aber doch die Entstehung einer einheitlichen Symbolik behindert. Ein flüchtiger Blick in die Arbeiten über den «Einsatz elektronischer Rechenmaschinen im Versicherungsbetrieb» <sup>2)</sup> bestätigt diese Beobachtung. Als Mangel kommt dieser Zustand auch in den «Verhandlungen des XV. internationalen Kongresses der Versicherungsmathematiker» <sup>3)</sup> in folgenden Worten zum Ausdruck: «Das Diagramm (Ablaufdiagramm, Flowchart) kann so voller Symbole und persönlicher Abkürzungen sein, dass es praktisch für jede andere Person ausser der, die es erstellt hat, unverständlich ist.»

Eine allen Rechengernäten gemeinsame Symbolik könnte nicht nur ein Mittel sein, vollständige Abläufe einheitlich darzustellen, sondern darüber hinaus, vorher unbekannte Beziehungen zwischen den verschiedenen Maschinentypen freizulegen. Im folgenden soll versucht werden, eine einheitliche Symbolik zu entwerfen und die Abhängigkeit der Abläufe vom Aufbau der Maschinen aufzuzeigen.

## I. Aufbau und Arbeitsweise elektronischer Rechenanlagen

Jedes programmgesteuerte Rechengernät besitzt mindestens eine Eingabestation, einen Speicher, ein Rechenwerk und eine Ausgabevorrichtung. In den Eingabestationen wird das zu verarbeitende Zahlenmaterial paketweise – in sogenannten Informationseinheiten zusammengefasst – eingelesen und in den Speicher der Maschine übertragen. Den Begriff der Informationseinheit definiert Thüring <sup>4)</sup> als

<sup>1)</sup> MVSV, Bd. 57, Heft 2, S. 316.

<sup>2)</sup> MVSV, Bd. 57, Heft 2, S. 221 ff.

<sup>3)</sup> Bd. III, deutsche Ausgabe, S. 61.

<sup>4)</sup> B.Thüring: «Die Logik der Programmierung», S. 71.

eine Folge von Angaben, welche auf Grund irgendwelcher zwischen ihnen bestehenden Beziehungen innerhalb eines Problems als eine zusammengehörige Einheit behandelt werden. Jeder Eingabestation ist im Speicher eine ganz bestimmte Zellengruppe zugeordnet, welche nunmehr zum Träger der Informationseinheit geworden ist. Ihr werden die zur Lösung eines Problems notwendigen Informationen entnommen, in andere Speichergruppen verschoben oder ins Rechenwerk abberufen und den Operationen zugeführt. Ist diese Bearbeitung beendet, wird die nächste Informationseinheit nachgeschoben und ihr Inhalt ebenfalls dem Rechenprogramm unterzogen. Während dieses Ablaufes entstehen Resultate, welche wiederum in besonderen Zellengruppen zu Ausgabeinformationseinheiten zusammengestellt und über die ihnen zugeordneten Ausgabevorrichtungen ausgegeben werden.

## II. Problemanalyse

Nachdem das zum Einsatz gelangende Gerät feststeht, handelt es sich zunächst darum, die im Rahmen einer Aufgabe auftretenden Ein- und Ausgabeinformationseinheiten zu bestimmen, ihre Reihenfolge festzulegen und sie den Ein- bzw. Ausgabestationen zuzuordnen. Wie eingangs skizziert wurde, überführt jede Eingabevorrichtung den Inhalt eines Trägers <sup>1)</sup> in eine ganz bestimmte Zellengruppe des Speichers.

Wir ordnen einer Eingabevorrichtung einen Buchstaben, etwa  $L$  zu und bezeichnen mit

$$L_i \quad (i = 1, 2, \dots, n)$$

die Folge der darin einzulesenden Träger. Unter  $(L_i)$  soll dann der Inhalt des  $i$ -ten, im Gerät  $L$  eingegebenen Trägers verstanden werden. Den Vorgang der Eingabe stellen wir durch den Befehlsschritt

$$(L_i) \Rightarrow \{I\} \quad (\text{lies: Inhalt von } L \text{ ergibt die Informationseinheit } I)$$

dar. Im Speicher liegt nunmehr die Informationseinheit  $\{I\}$ , gemäss Definition eine Folge von Angaben

$$\{I\} \equiv I_1, I_2, \dots, I_m.$$

Diese letzten Symbole gestalten sich etwas anschaulicher, wenn wir bei der Aufgliederung der Informationseinheit  $\{I\}$  anstelle der Numerierung

<sup>1)</sup> Es kann sich hierbei um eine Lochkarte, ein Feld auf einem Magnetband oder einem Lochstreifen handeln.

einen die einzelnen Informationen kennzeichnenden Index verwenden, wie beispielsweise

$$\begin{aligned} I_n &= \text{Policennummer,} \\ I_N &= \text{Name des Versicherten,} \\ I_A &= \text{Abschlussalter,} \\ I_S &= \text{Summe,} \\ I_L &= \text{Landnummer,} \\ &\text{usw.} \end{aligned}$$

Wird gleichzeitig auf zwei Eingabegeräten gearbeitet, dann soll ganz analog der «Elementarschritt»

$$(U_j) \Rightarrow \{J\}$$

den Übergang der im Gerät  $U$  bearbeiteten  $j$ -ten Informationseinheit in den Speicher charakterisieren.  $\{J\}$  könnte sich wie folgt zusammensetzen:

$$\begin{aligned} J_D &= \text{Datum,} \\ J_L &= \text{Landnummer,} \\ J_K &= \text{Umrechnungskurs.} \end{aligned}$$

Umgekehrt werden Resultate zu einer Ausgabeeinheit, etwa

$$\{P\} \equiv P_n, P_N, P_S, \dots$$

zusammengestellt und über ein Ausgabegerät, z. B.  $D$ , auf einen Träger ausgegeben:

$$\{P\} \Rightarrow (D_p).$$

Ob es sich hier um den Druck einer Zeile, um die Stanzung einer Lochkarte oder eines Lochstreifens oder ob es sich um die Magnetisierung eines bestimmten Feldes auf einem Band handelt, ist völlig belanglos. Die vorliegende Instruktion besagt einfach, dass die dem Gerät  $D$  zugeordnete Informationseinheit  $\{P\}$  auf den  $p$ -ten Träger abzusetzen sei. Jedem weiteren zum Einsatz gelangenen Ausgabegerät werden wir einen andern Buchstaben zuordnen und die darin verarbeiteten Informationseinheiten besonders bezeichnen. Bei der Analyse einer Aufgabe sind zunächst die verschiedenen Typen von Informationseinheiten und deren Aufgliederung in die Folge ihrer Informationen zu ermitteln. Nachdem dann die Beziehungen zwischen den Eingabe- und Ausgabeinformationseinheiten formuliert sind, ist bei gegebenem Gerät der Arbeitsablauf weitgehend festgelegt.

### III. Symbolik der Operationen

Ganz allgemein soll vorerst mit  $(Z)$  der Inhalt der Speicherzelle  $Z$  bezeichnet werden. Die *Übertragung* von  $(Z_1)$  nach  $Z_2$  stellen wir mit dem Symbol

$$\triangleright \boxed{(Z_1) \Rightarrow (Z_2)} \triangleright \quad (1)$$

dar. Die Elementarschritte

$$\triangleright \boxed{(Z_1) + (Z_2) \Rightarrow (Z_3)} \triangleright \quad (2)$$

$$\triangleright \boxed{(Z_1) - (Z_2) \Rightarrow (Z_3)} \triangleright \quad (3)$$

$$\triangleright \boxed{(Z_1) \cdot (Z_2) \Rightarrow (Z_3)} \triangleright \quad (4)$$

$$\triangleright \boxed{\frac{(Z_1)}{(Z_2)} \triangleright (Z_3)} \triangleright \quad (5)$$

befehlen die Ausführung der vier *Grundrechenoperationen*. Ausserdem sind stets die beiden *Vergleiche*

$$\triangleright \text{⌢} \boxed{(Z_1) : (Z_2)} \text{⌢} \quad (6)$$

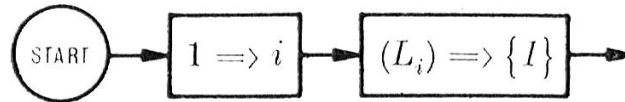
$$\triangleright \text{⌢} \boxed{(Z_1) : (Z_2)} \text{⌢} \quad (7)$$

möglich, wobei der Doppelpunkt nicht eine Division, sondern einen Vergleich charakterisiert.

Diese sieben Operationen sind in *jedem* programmgesteuerten Rechengertät in irgendeiner Weise realisierbar.

### IV. Symbolik der Abläufe

Beschränken wir uns vorerst auf ein Gerät mit *einer* Eingabestation, so steht nach dem Vorgang des Einlesens



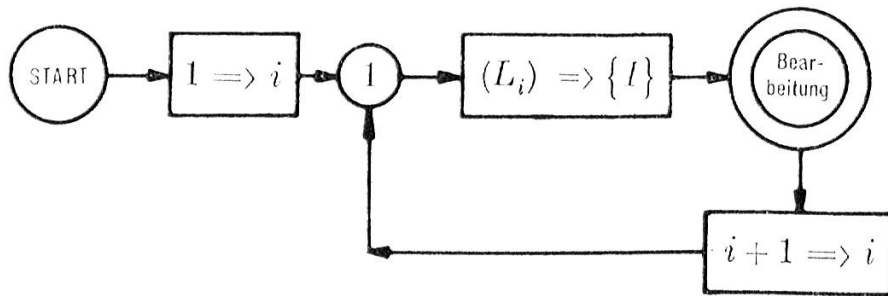
der ersten Informationseinheit <sup>1)</sup> eine Folge von Informationen

$$I_a, I_b, I_c, \dots$$

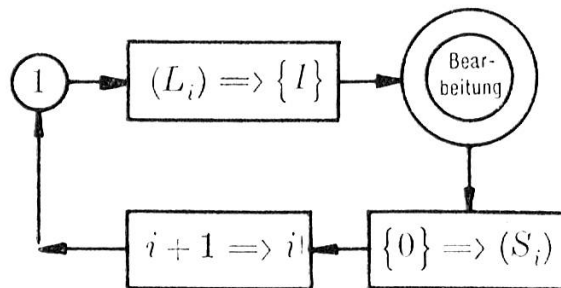
im Speicher zur Verfügung. Ihr werden einzelne Grössen entnommen und den Operationen zugeführt. Diesen Teil des Programms wollen wir vorläufig *Bearbeitung* nennen und mit



symbolisch darstellen. Nachdem die erste Informationseinheit bearbeitet ist, wird der nächste Träger ( $L_2$ ) eingelesen. Rein formal erreichen wir diesen Übergang durch die Erhöhung des Index  $i$  um eins  $i + 1 \Rightarrow i$  mit anschliessendem Sprung zur Eingabeinstruktion:



Liefert jede Eingabeinformationseinheit eine und nur eine Ausgabeinheit, welche etwa  $\{0\}$  genannt und auf einen Träger  $S_i$  abgesetzt werden soll, so ändert sich der Ablauf in folgender Weise:

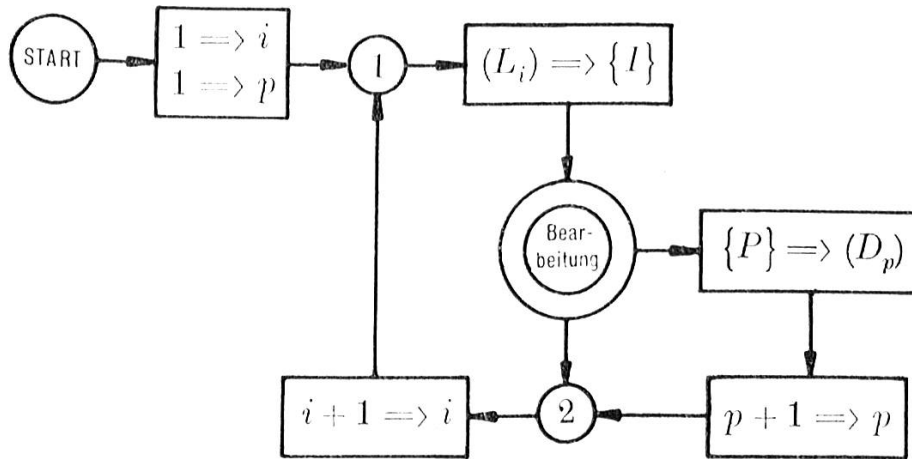


Wird der Träger der Eingabeinheiten durch die Resultate ergänzt, so tritt in



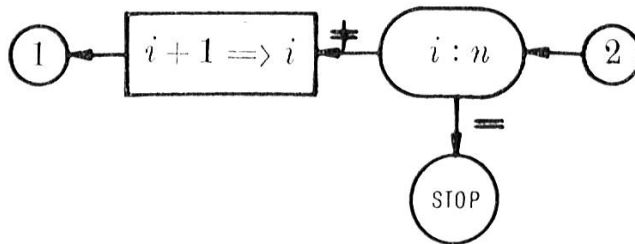
<sup>1)</sup> Sie ist im Ablaufdiagramm durch den Anfangswert  $1 \Rightarrow i$  festgelegt.

Im Ablaufdiagramm



erkennt man leicht einen etwas allgemeiner gehaltenen Fall: Eine Ausgabeinformationseinheit  $\{P\}$  entsteht erst, nachdem eine gewisse Anzahl von Eingabeeinheiten die «Bearbeitung» durchlaufen haben.

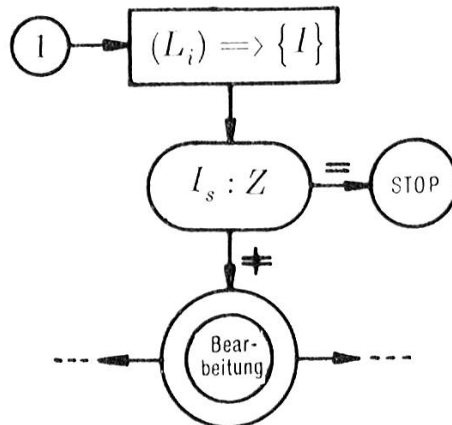
Ist die Menge der einlaufenden Informationseinheiten gegeben und gleich  $n$ , so gelangt der Ablauf mit Hilfe folgender Ergänzung zu seinem Abschluss:



Die Zahl  $n$  wird aber in vielen Fällen nicht bekannt sein. Der letzte Träger der Eingabe wird dann mit einem besonderen Merkmal (Sentinel)

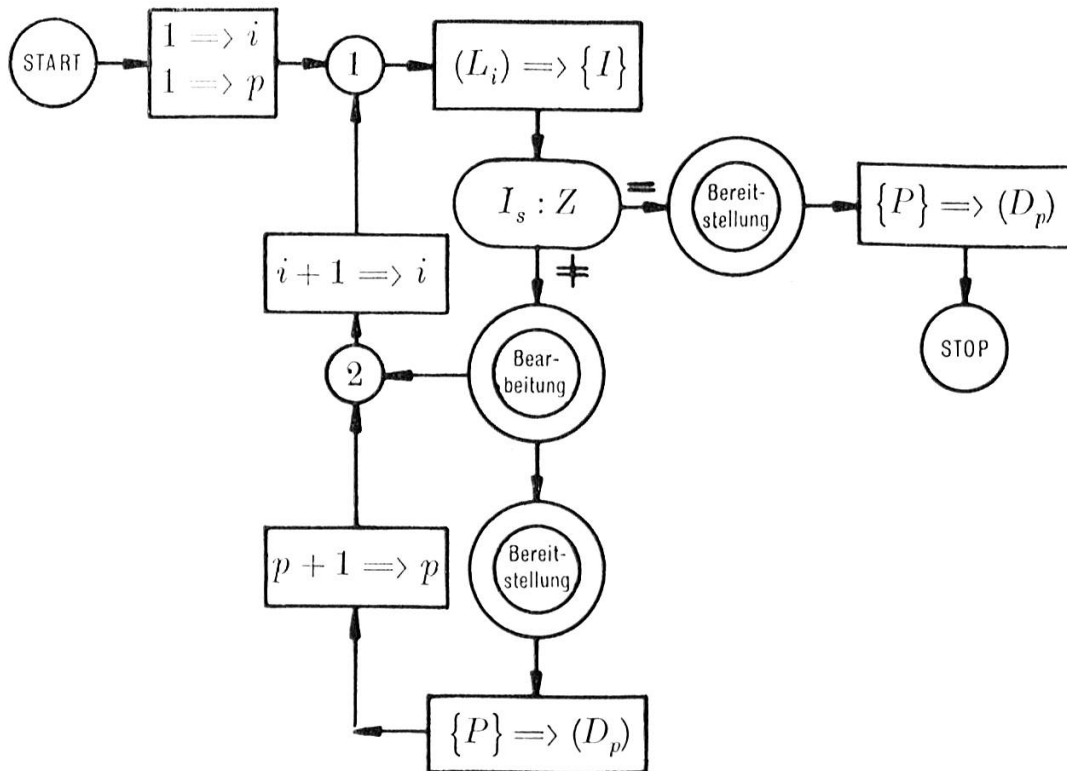
$$I_s = Z$$

versehen, und jedem einzelnen Eingabeschritt ist der sogenannte Sentineltest

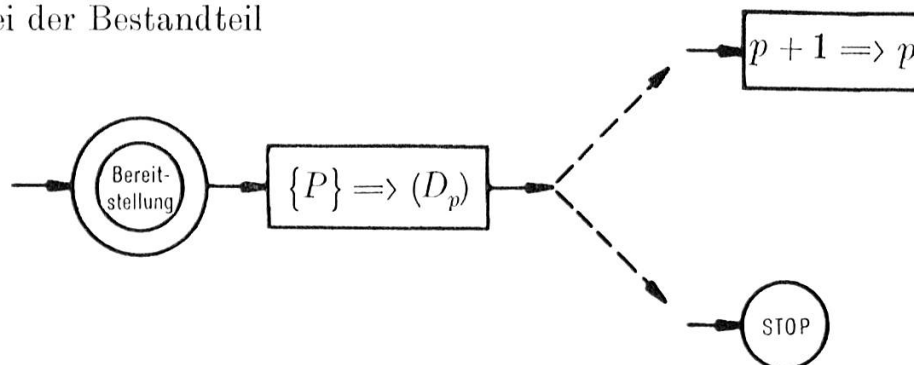


anzufügen.

Der vorliegende *Stop*-Befehl ist im allgemeinen an dieser Stelle verfrüht, weil die letzte Ausgabeinformationseinheit die Maschine in diesem Zeitpunkt noch nicht verlassen hat. Daher ist vorerst noch ein Teil der «Bearbeitung» <sup>1)</sup> zu durchlaufen und die letzten Resultate auf den letzten Träger auszugeben. Um die Darstellung dieses Ablaufes zu ermöglichen, soll die «Bearbeitung» in zwei Teilpläne *Bearbeitung* und *Bereitstellung* aufgespalten werden, wobei die *Bereitstellung* denjenigen Teil der ehemaligen «Bearbeitung» umfassen soll, der sich auf die Zusammenstellung der Ausgabeinheiten beschränkt. Dann ergibt sich als Gesamtplan



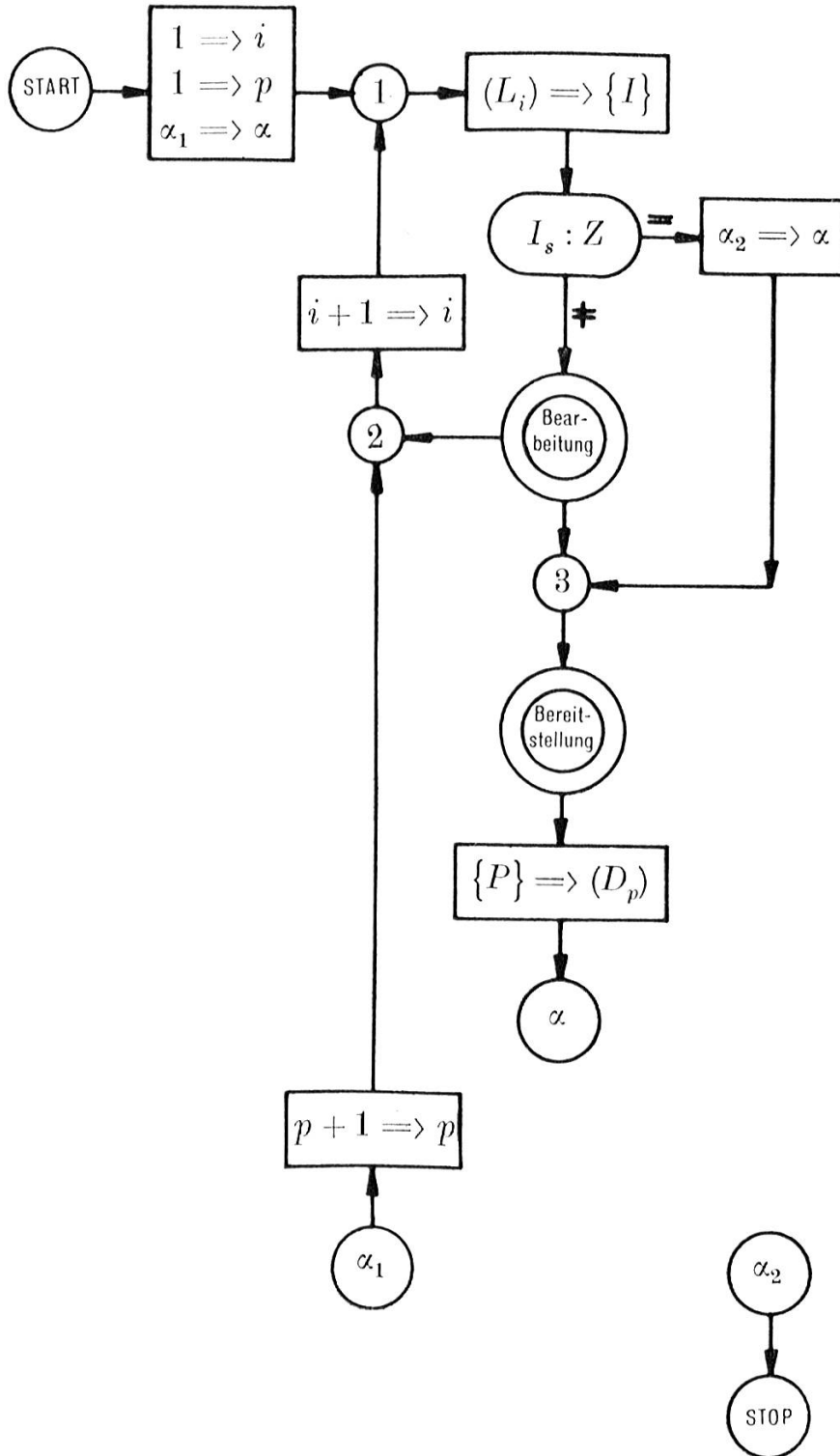
wobei der Bestandteil



zweimal und mit zwei verschiedenen Ausgängen in Erscheinung tritt.

<sup>1)</sup> Die Zusammenstellung der letzten Ausgabeinformationseinheit.

Dieser Teilplan könnte derart umfangreich sein, dass der Versuch unternommen werden müsste, ihn nur einmal auftreten zu lassen. Dies geschieht in der Weise, dass sein Ausgang mit einer «Weiche» (variabler Konnektor) versehen wird:



Darin bestimmen die Übertragungsinstruktionen  $\alpha_1 \Rightarrow \alpha$  und  $\alpha_2 \Rightarrow \alpha$  die an der Stelle  $\textcircled{\alpha}$  vorzunehmenden Sprünge.

## V. Über die Abhängigkeit der Abläufe vom Aufbau der Maschinen

Gegeben sei zunächst die Folge  $\{I\}^{(i)}$  ( $i = 1, 2, \dots, v$ ), wobei sich ein  $\{I\}^{(i)}$  aus einem Merkmal  $I_s^{(i)} = a$  und einer Nummer  $I_n^{(i)} > 0$  zusammensetze. Die Folge sei bestimmt durch die Bedingung

$$I_n^{(i+1)} > I_n^{(i)} \quad (i = 1, 2, \dots, v-1).$$

Ausserdem sei eine weitere Folge  $\{J\}^{(j)}$  ( $j = 1, 2, \dots, u$ ) gegeben. Jede Informationseinheit bestehe aus dem Merkmal  $J_s^{(j)} = b$  und der Nummer  $J_n^{(j)} > 0$ .

Die Ungleichung 
$$J_n^{(j+1)} \geq J_n^{(j)} \quad (j = 1, 2, \dots, u-1)$$

bestimmt diese zweite Folge.

Alle Elemente von  $\{J\}^{(j)}$  ( $j = 1, 2, \dots, u$ ) sollen nunmehr zur Bildung einer dritten Folge  $\{K\}^{(k)}$  ( $k = 1, 2, \dots, w$ ) herangezogen werden. Der Folge  $\{I\}^{(i)}$  ( $i = 1, 2, \dots, v$ ) werden diejenigen Glieder entnommen, welche für  $\{K\}^{(k)}$  folgende Bedingungen ergeben:

1.  $K_n^{(k+1)} \geq K_n^{(k)} \quad (k = 1, 2, \dots, w-1).$  (1)

2. Wenn unter der Annahme  $K_n^{(0)} = 0$ ,  $K_n^{(k)} > K_n^{(k-1)}$  (2)  
( $k = 1, 2, \dots, w-1$ ) wird, dann und nur dann soll  $K_s^{(k)} = a$  sein.

3. Wenn  $K_s^{(k)} = a$  wird, dann soll  $K_s^{(k+1)} = b$  sein, (3)  
( $k = 1, 2, \dots, w-1$ )<sup>1)</sup>.

In dieser festgelegten Reihenfolge sollen die zwei verschiedenen Typen von Informationseinheiten (Merkmal  $a$  und  $b$ ) der  $K$ -Folge in ein Bearbeitungsprogramm einlaufen.

Nehmen wir ein Gerät mit einer einzigen Eingabestation  $S$  an, dann müssen wir die  $K$ -Folge als gegeben voraussetzen.

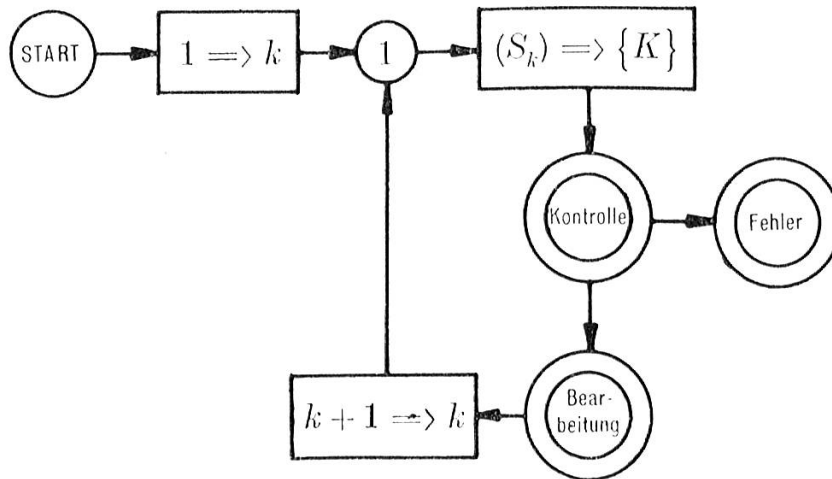
---

<sup>1)</sup> In der Praxis könnte es sich hier um einen nach aufsteigenden Nummern sortierten Kartenstapel handeln, wobei bei jedem Nummernwechsel eine Meisterkarte eingeschoben ist.

Der Eingabe

$$(S_k) \Rightarrow \{K\} \quad (k = 1, 2, \dots, w)$$

hat somit ein separater Mischvorgang <sup>1)</sup> voranzugehen, der die  $K$ -Folge gemäss Bildungsgesetz aus den Elementen der gegebenen Folgen  $\{I\}^{(i)}$  und  $\{J\}^{(j)}$  herstellt. Wir können uns dann darauf beschränken, die einlaufende Folge daraufhin zu prüfen, ob sie die gewünschten Bedingungen erfüllt. Zwischen Eingabe und Bearbeitung wird daher ein *Kontrollprogramm* eingeschoben:



Der Teilplan *Kontrolle* nimmt im vorliegenden Fall folgende Gestalt an (siehe Abbildung folgende Seite).

Darin bedeuten

- Fehler 1:* Die Voraussetzung  $K_n^{(k)} > 0$  trifft nicht zu,
- Fehler 2:* Die Bedingung (1) ist nicht erfüllt,
- Fehler 3:* Die Bedingung (2) ist nicht erfüllt,
- Fehler 4:* Die Bedingung (3) ist nicht erfüllt.

Man beachte, dass am Anfang ein Arbeitsspeicher  $W_n$  mit Null belegt werden muss, um für  $K_n^{(1)}$  die Bedingung (2) zu erzwingen.

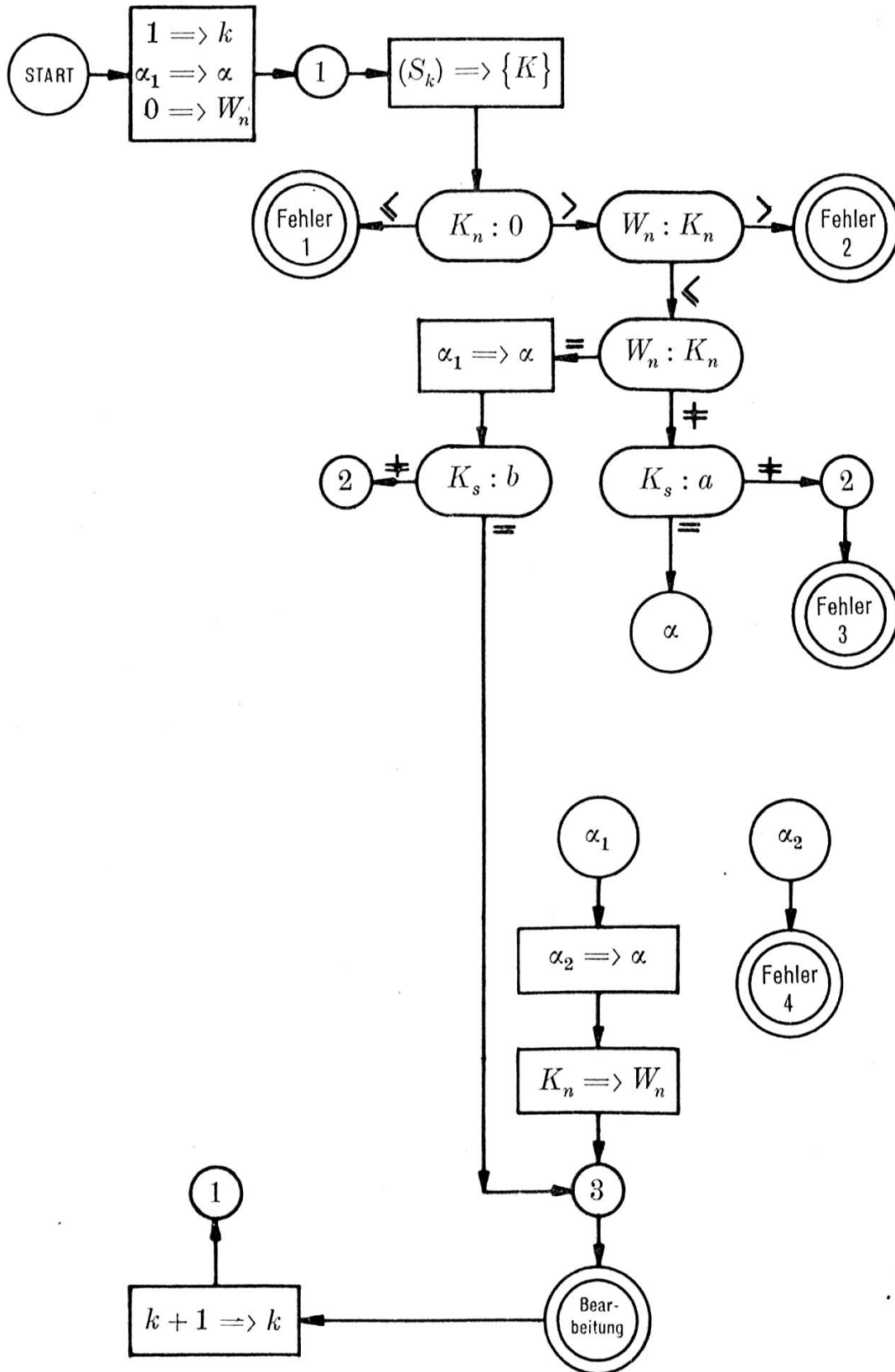
Nehmen wir nunmehr ein Gerät mit zwei Eingabestationen  $L$  und  $U$  an und bezeichnen wir die Eingabevorgänge mit

$$(L_i) \Rightarrow \{I\} \quad (i = 1, 2, \dots, v)$$

und

$$(U_j) \Rightarrow \{J\} \quad (j = 1, 2, \dots, u),$$

<sup>1)</sup> Im Falle von Lochkarten kann dieser Mischvorgang von Hand oder mit Hilfe einer Sortiermaschine oder eines Mischers erfolgen.



dann werden wir vorerst in einem *Mischprogramm* die Informationseinheiten der beiden gegebenen Folgen derart ineinanderschieben müssen, dass die *K-Folge* entsteht. Anschliessend durchlaufen die Elemente dieser Folge den *Bearbeitungsplan* in gleicher Weise wie im vorangehenden Programm.

Streng genommen sind die vorausgesetzten Eigenschaften der beiden getrennt einlaufenden Folgen in zwei Unterplänen *Kontrolle 1* und *Kontrolle 2* zu prüfen, um dann erst auf Grund der Bildungsvorschrift zur *K-Folge* verarbeitet zu werden.

Das Ablaufdiagramm (auf folgender Seite) zeigt den ganzen Mischvorgang.

Zwischen den Konnektoren ① und ② wir die *I-Folge* eingelesen und geprüft. Dasselbe geschieht für die *J-Folge* von ② bis ③. Konnektor ③ führt in die Abfrage, ob die beiden eingelesenen Nummern gleich sind. Trifft dies zu, dann ist  $\{I\}^{(1)}$  das erste Element der *K-Folge* und läuft als  $\{K\}^{(1)}$  in die *Bearbeitung*. Über ② wird dann das Element  $\{J\}^{(1)}$  erfasst und geht als  $\{K\}^{(2)}$  in die *Bearbeitung*.

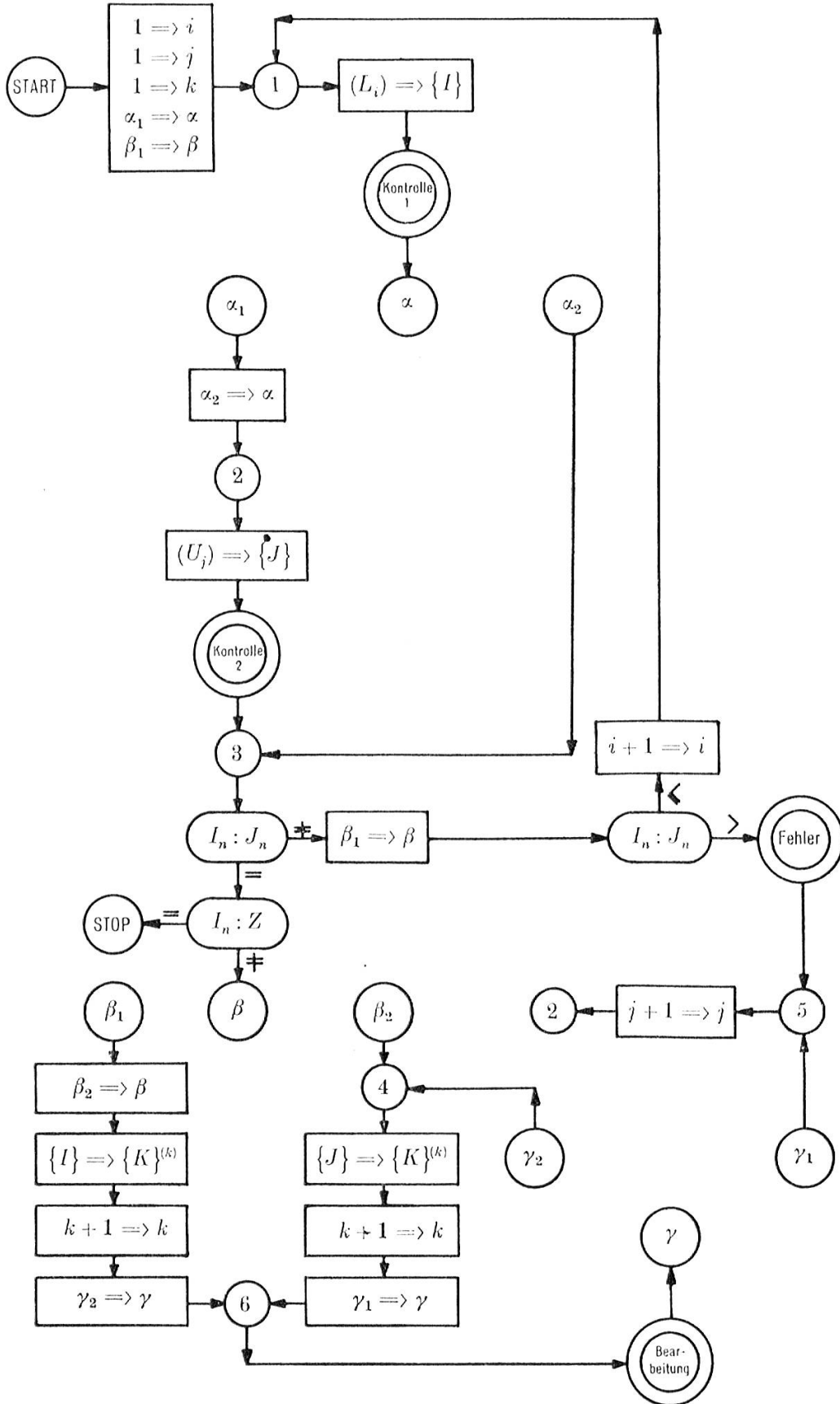
Sind die Nummern  $I_n$  und  $J_n$  nicht gleich, dann ist zunächst festzustellen, welche von beiden die grössere ist, um im entsprechenden Eingabegerät für Nachschub zu sorgen.

Die «Weiche» ② musste gesetzt werden, um im allerersten Durchgang aus beiden Folgen je das erste Glied einzulesen. In jedem weiteren Fall wird immer nur entweder ein Element der *J-* oder der *I-Folge* eingegeben. der beigefügte *Fehler-Plan* zeigt einen Widerspruch zur Bedingung (2) <sup>1)</sup> auf.

Wie bereits angedeutet, mündet in beiden Abläufen die *K-Folge* in den *Bearbeitungs-Plan* ein. Für die Maschine mit *einer* Eingabestation war es notwendig, die beiden gegebenen Folgen in einem separaten Mischvorgang zur *K-Folge* zu verarbeiten. In der Praxis bedeutet das aber einen zusätzlichen Durchgang des gesamten zu verarbeitenden Zahlenmaterials durch ein anderes Gerät. Steht hingegen eine Anlage mit *zwei* Eingabevorrichtungen zur Verfügung, dann haben die beiden getrennt einlaufenden Folgen zunächst ein *Misch-Programm* zu durchschreiten. Dieser *Misch-Plan* ist somit im ersten Fall das Strukturdiagramm des Arbeitsablaufs einer unabhängig arbeitenden Misch-

---

<sup>1)</sup> Vgl. Seite 68.



maschine und im zweiten Fall ein Bestandteil des Gesamtplans. Als gemeinsamer Kern bleibt der *Bearbeitungs*-Plan.

Weist der Speicher einen Umfang auf, der es erlaubt, eine der beiden gegebenen Folgen gesamthaft zu speichern, dann verlagern sich weitere Arbeitsabläufe. Die Elemente dieser Folge können völlig ungeordnet in den Speicher der Maschine gelegt werden, wobei sich dann das Hauptprogramm um einen *Speicher*-Plan und um eine *innere Sortierung* erweitert.

Ähnlich liegen die Verhältnisse bei der Ausgabe mehrerer verschiedener Ausgabe-Informationseinheiten. Von der Anzahl der Ausgabegeräte hängt es dann ab, ob die Trennung der Ausgabeeinheiten innerhalb oder ausserhalb des Gesamtablaufs vorzunehmen ist. Das bedeutet ganz analog eine Verminderung oder Vermehrung von Maschinenpassagen. Völlig unabhängig vom Gerät bleibt somit einzig der *Problemerkern*.

### Zusammenfassung

In ihrem grundsätzlichen Aufbau und in der Fähigkeit, eine Folge von bestimmten Grundoperationen zu realisieren, sind sich alle programmgesteuerten Rechenanlagen gleich. Auf dieser Grundlage wurde der Versuch unternommen, eine allen Rechengeräten gemeinsame Symbolik der Ablaufdiagramme aufzubauen. Erst eine einheitliche Schreibweise erlaubte es, die eingangs gestellte Frage anzugehen, wie weit das logische Gerüst eines operativen Ablaufes unabhängig von einem bestimmten Maschinentyp in Erscheinung tritt. An einem Beispiel erläutert, zeigte sich, dass die Abläufe als Ganzes betrachtet, weitgehend von der Anzahl der Ein- und Ausgabestationen sowie von der Grösse des Speichers abhängen. Stets lässt sich aber ein ganz bestimmter Problemerkern unabhängig von einer bestimmten Anlage darstellen. Die Beziehungen zwischen verschiedenen Maschinentypen kommen in den Verlagerungen gewisser Programmteile zum Ausdruck.

## Résumé

La réalisation d'un travail sur un ordinateur à programme préétabli dépend essentiellement du nombre des mécanismes de lecture (input units) et de tabulation (output units) comme aussi de la capacité de la mémoire du cerveau électronique. En partant de ces prémisses l'auteur propose un système de symboles pour les diagrammes de travaux (flowcharts) communs à tous les types d'ordinateurs.

## Riassunto

La realizzazione d'un calcolo con una macchina elettronica dipende dal numero di «input units» et di «output units» come pure dalla capacità della memoria del cervello elettronico. Partendo da queste premesse l'autore propone un sistema di simboli per i «flowcharts» che possono essere usati per tutti i tipi di calcolatrici elettroniche.

## Summary

The feasibility of any numerical computation on a programmed electronic data processing machine depends on the number of input and output units and the storage capacity. Bearing these fundamentals in mind the author attempts to trace out a universal flowchart symbolism for all types of computers.