Erteilt auf Grund des Ersten Überleitungsgesetzes vom 8. Juli 1949
(WiGBl. S. 175)

BUNDESREPUBLIK DEUTSCHLAND

AUSGEGEBEN AM
25. JULI 1955

DEUTSCHES PATENTAMT

PATENTSCHRIFT

Nr. 926 449

KLASSE 42m GRUPPE 15

Z 530 IX b / 42 m

Dipl.-Ing. Konrad Zuse, Neukirchen (Kr. Hünfeld)
ist als Erfinder genannt worden

Zuse K.-G., Neukirchen (Kr. Hünfeld)

Kombinierte numerische und nichtnumerische Rechenmaschine

Patentiert im Gebiet der Bundesrepublik Deutschland vom 13. Mai 1950 an
Patentanmeldung bekanntgemacht am 5. August 1954
Patenterteilung bekanntgemacht am 17. März 1955

Die Erfindung betrifft eine Rechenvorrichtung, die erfindungsgemäß in der Lage ist, Rechenoperationen sowohl mit numerischen Werten als auch mit aussagenlogischen Ausdrücken durchzuführen.

Es sind grundsätzlich Rechenvorrichtungen bekannt, die Aufgaben der theoretischen Logik, d. h. Rechenoperationen mit aussagenlogischen Ausdrücken, durchzuführen vermögen, auch solche, deren Rechenablauf durch ein Programmwerk gesteuert wird. Ebenso sind programmgesteuerte Rechenvorrichtungen zur Durchführung numerischer Rechnungen bekannt.

Vorliegende Erfindung stellt sich die Aufgabe, eine Rechenvorrichtung zu schaffen, die, gesteuert von einem an sich bekannten Programmwerk oder Schlüsselprogrammwerk, beliebig komplizierte Rechenoperationen mit Variablen beliebiger Art (wie z. B. mit aus Ja-Nein-Wert-Kombinationen bestehenden aussagenlogischen Ausdrücken, Zahlen od. ä.) nach den bekannten Regeln der Grundoperationen des Aussagenkalküls bzw. der Arithmetik durchführen kann, sofern nur das Rechenprogramm explizit in Form eines Rechenplanes gegeben ist. Hierbei kann der Rechenplan selbst beliebig kompliziert aufgebaut sein, wobei auch numerische und aussagenlogische Rechenoperationen miteinander verknüpft, ineinander verschachtelt oder gegenseitig voneinander abhängig gemacht werden können.

Eine solche Rechenvorrichtung kann neben der Durchführung numerischer Rechnungen alle Aufgaben der theoretischen Logik mit All- und Existenzoperatoren sowie mit den Operatoren »Diejenigen, welche . . .«, »Derjenige, welcher . . .« und dem μ-Operator »Das nächste Glied mit der Eigenschaft . . .« lösen, statistische Aufgaben beliebiger Art erledigen, die formale Richtigkeit komplizierter Rechenanweisungen prüfen u. a. m.

Der Erfindungszweck wird durch die Kombination folgender drei an sich bekannter Vorrichtungen erreicht:

1. Aussagenlogische Rechenvorrichtung, 2. numerische Rechenvorrichtung, 3. Programmwerk (mit oder ohne Schlüsselwerk).

Eine Verschachtelung bzw. Kopplung numerischer und aussagenlogischer Rechenoperationen wird durch besondere Übertragungsmittel zur Überleitung von Werten, z. B. des numerischen in den aussagenlogischen Wertekreislauf, ermöglicht, aber auch umgekehrt; das heißt, der Ablauf des Rechenprogramms kann durch ein erst im Verlauf der Rechnung anfallendes Ergebnis gesteuert werden, das sowohl numerischen als auch aussagenlogischen Charakter haben kann. Aussagenlogische Werte stellen beispielsweise Aussagen über Zahlenpaare dar, wie $a > b$. Eine solche Aussage (Angabe), die einen Ja-Nein-Wert darstellt, kann im Programmwerk zur Steuerung bedingter Kommandos verwendet werden, und zwar insofern, daß entweder ein Rechenprogramm A abläuft, falls $a > b$ (Bedingung erfüllt) ist, oder ein Rechenprogramm B, falls nicht $a > b$ (Bedingung nicht erfüllt) ist.

Rechenprogramme, an welche die Forderung großer Variabilität gestellt werden, können sehr kompliziert aufgebaut sein und kommen dann mit Bedingungen dieser einfachen Art nicht mehr aus. Zu ihrer Steuerung werden Angaben benötigt, die aus irgendwelchen gegebenen Angaben nach einer Vorschrift (Rechenanweisung) abgeleitet sind. Bekanntlich können aus mehreren Angaben durch Verknüpfungen (Operationen des Aussagenkalküls) wieder neue Angaben (Resultatangaben) gebildet werden, die ihrerseits wieder Ja-Nein-Werte darstellen, die richtig oder falsch (positiv bzw. negativ) sein können.

Grundoperationen des Aussagenkalküls sind

Konjunktion:	$a \wedge b$	gelesen:	a und b,
Disjunktion:	$a \vee b$	gelesen:	a oder b,
Negation:	$\bar{a}$	gelesen:	nicht a bzw. a quer.

Aussagenlogische Ansätze mit einer einfachen konjunktiven bzw. disjunktiven Verknüpfung lauten beispielsweise

$a \wedge b$ äq c gelesen: a und b äquivalent c,
$a \vee b$ äq d gelesen: a oder b äquivalent d.

Darin bedeutet

$\wedge$ = Operationszeichen für die aussagenlogische konjunktive Verknüpfung (a und b müssen beide positiv sein, wenn die durch die Verknüpfung gebildete neue Angabe c ebenfalls positiv sein soll),

$\vee$ = Operationszeichen für die aussagenlogische disjunktive Verknüpfung (a oder b muß positiv sein, wenn die neu gebildete Resultatangabe d positiv sein soll; hierbei können auch a und b zugleich positiv sein),

$\bar{a}$ = ein Querstrich über einer Angabe kennzeichnet die Negation, wodurch eine positive Angabe negativ wird und umgekehrt eine negative positiv,

äq = ist ein Substitutionszeichen, das besagt, daß der links von äq stehende Ausdruck und die rechts neben äq stehende Resultatangabe gleichwertig füreinander gesetzt werden können.

Durch mehrfache Anwendung der Grundoperationen lassen sich komplizierte Angabenverknüpfungen (logistische Formeln mit einer Vielzahl von Variablen) bilden, aus denen nach Einsetzen bestimmter Werte (Ja-Nein-Werte) an die Stelle der beteiligten Variablen nach den Vorschriften des Aussagenkalküls eine Resultatangabe abgeleitet werden kann, die ihrerseits wieder eine Angabe über den Wahrheitswert der in Frage stehenden logistischen Formel, also wiederum einen Ja-Nein-Wert darstellt.

Weitere aussagenlogische Operationszeichen für zusammengesetzte aussagenlogische Operationen sind in folgenden Ansätzen zu erkennen:

Äquivalenz: $a \sim b$ äq $(a \wedge b) \vee (\bar{a} \wedge \bar{b})$ äq c,
Disvalenz: $a \nsim b$ äq $(a \wedge \bar{b}) \vee (\bar{a} \wedge b)$ äq d.

Diese Ansätze besagen, daß c positiv wird, wenn a äquivalent b ist, d. h. wenn entweder sowohl a als auch b positiv oder sowohl a als auch b negativ sind, oder daß d positiv wird, wenn a disvalent b ist, d. h. wenn entweder a positiv und b negativ oder a negativ und b positiv ist.

Aussagenlogische Rechenvorrichtungen leiten beispielsweise über elektrische Verknüpfungsschaltungen mit Hilfe elektromagnetischer Relais oder mechanischer Schaltglieder in zeitlicher Folge nach den Regeln für die Grundoperationen des Aussagenkalküls aufbauend durch Verknüpfung jeweils zweier Angaben Resultat- bzw. Zwischenresultatangaben ab. Der Verlauf der einzelnen Operationen wird gegebenenfalls durch ein Programmwerk (mit oder ohne Schlüsselwerk) gesteuert, das die Einzelkommandos des Rechenplanes schrittweise auf die Rechenvorrichtung gibt und deren Ausführung in der vorgeschriebenen zeitlichen Reihenfolge bewirkt.

Ein einfaches Beispiel soll das Zusammenspiel von numerischem und aussagenlogischem Rechenwerk erläutern.

Es werde angenommen, daß im Verlauf eines numerischen Rechenprozesses drei Zwischenresultate Z_1, Z_2, Z_3 anfallen, von deren Größe bzw. Vorzeichen der weitere Programmablauf nach folgender Vorschrift abhängen soll:

Programm A läuft ab, falls $Z_1 \geqq 100$,
$Z_2 < 0$,
$Z_3 \geqq 0$ ist,

andernfalls Programm B.

Aus den im numerischen Rechenwerk gebildeten Zwischenresultaten $Z_1 \ldots Z_3$ werden zunächst durch entsprechende Kommandogabe folgende Angaben a, b, c (Ja-Nein-Werte) gebildet:

$$Z_1 \geqq 100 \Rightarrow a$$
$$Z_2 \geqq 0 \Rightarrow b$$
$$Z_3 \geqq 0 \Rightarrow c$$

$\Rightarrow$ bedeutet: ergibt.

Die Bedingung für den Ablauf des Programms A (die Resultatangabe d) kann nun in Form nachstehender logistischer Formel angeschrieben werden:

$$a \wedge \bar{b} \wedge c \Rrightarrow d.$$

Sind demnach die Angaben a und c positiv, die Angabe b jedoch negativ, so ist die Resultatangabe d positiv, d. h. die Bedingung ist erfüllt. Die Angabe b erscheint in der Formel negiert, weil »b positiv« bedeutet $Z_2 \geqslant 0$, verlangt ist aber $Z_2 < 0$, also die Negation der Angabe b. Man erkennt aus diesem einfachen Beispiel, in welcher Weise aussagenlogisches und numerisches Rechenwerk in gegenseitiger Wechselwirkung stehen.

Der Erfindungsgedanke werde an Hand des folgenden Ausführungsbeispiels erläutert, das eine Rechenanlage darstellt, die im wesentlichen beispielsweise aus folgenden Einzelvorrichtungen bestehe:

1. Rechenvorrichtung zur Durchführung von Rechenoperationen mit aussagenlogischen Ausdrücken
   a) mit einem Rechenwerk *RW* 1 und
   b) mit einem Komponentenspeicher *KSp*.
2. Rechenvorrichtung zur Durchführung numerischer Operationen
   a) mit einem Rechenwerk *RW* 2 und
   b) mit einem Hauptspeicher *HSp*.
3. Programmwerk zur Steuerung beider Rechenvorrichtungen
   a) mit einem Ablaufwerk *Ablw* und
   b) mit einem Schlüsselwerk *Schlw*.

In der das Wirkungsschema einer solchen Rechenanlage darstellenden Fig. 1 sind folgende Wirkungsbereiche (Kommandokreise bzw. Wertekreisläufe) erkennbar:

Rechnender Kreislauf		Kommandokreis	
I	II	III	IV
aussagenlogische Werte	numerische Werte	Adresse und gegebenenfalls Operationskommando bzw. Übertragungskommando	Grundkommando

Im Kreislauf I vollzieht sich der aussagenlogisch Werteverkehr bei Durchführung aussagenlogischer Rechenoperationen, in deren Verlauf die zur Rechnung benötigten Werte jeweils als Ja-Nein-Wert dem Komponentenspeicher *KSp* entnommen und dem Rechenwerk *RW* 1 zugeführt werden. Das Resultat (gegebenenfalls ein oder mehrere Zwischenresultate) wird nach abgeschlossener Rechenoperation, im allgemeinen als Ja-Nein-Wert, im Komponentenspeicher gespeichert.

Im Kreislauf II verkehren bei Durchführung numerischer Rechenoperationen numerische Werte. Ablesen und Eingeben von Werten in das numerische Rechenwerk *RW* 2, die Durchführung von Rechenoperationen und Speichern von Zwischenwerten oder des Resultats im Hauptspeicher *HSp* erfolgt in bekannter Weise.

Die mit I*ü* und II*ü* bezeichneten Wirkungslinien stellen Übertragungswege zur Verbindung des aussagenlogischen mit dem numerischen Wertekreislauf (bzw. analog umgekehrt) dar, über die eine Überleitung von Zahlen bzw. Wertekombinationen od. ä. von einem Kreislauf zum anderen erfolgen kann.

Der Übertragungsweg III*ü* ermöglicht die Überleitung von Werten des numerischen Wertekreislaufes zum Programmwerk, vorzugsweise zum Schlüsselwerk, und dient zur Steuerung des Schlüsselwerkes von den Rechenwerken aus, z. B. zur Übertragung von Schlüsseln, die auf Grund vorausgehender Rechenoperationen in der oder den Rechenvorrichtungen errechnet werden. Es können somit beispielsweise Adressen auf Grund von Ausgangswerten errechnet werden, die als Ergebnis vorausgehender Operationen im Verlaufe des Rechenablaufes beliebigen Variationen unterworfen sein können.

Beide Rechenvorrichtungen, deren Rechenoperationen und Werteverkehr, werden vom Programmwerk aus gesteuert, das im dargestellten Ausführungsbeispiel als Schlüsselprogrammwerk ausgebildet ist und im wesentlichen aus dem Ablaufwerk *Ablw* (dem eigentlichen Kommandogeber) und dem Schlüsselwerk *Schlw* besteht.

Rechenvorrichtungen und Programmwerk als Einzelvorrichtungen werden als grundsätzlich bekannt vorausgesetzt und sind nicht Gegenstand der Erfindung, daher auch in ihrer Wirkungsweise nur dann und so weit beschrieben, als zum Verständnis des Erfindungsgedankens erforderlich ist. Der Übersichtlichkeit halber sind im Wirkungsschema die bekannten technischen Mittel zum Eingeben bzw. Anzeigen von Ausgangs- bzw. Resultatwerten (z. B. Tastatur und Anzeigefeld), die einen integrierenden Bestandteil jeder Rechenvorrichtung darstellen, nicht gezeichnet, zumal sie mit dem zu erläuternden Erfindungsgedanken nur mittelbar in Beziehung stehen.

Alle Kommandos, die das Ablaufwerk des Schlüsselwerkes verlassen, umfassen je zwei Komponentengruppen, von denen die eine Gruppe *GK* zur Darstellung des Grundkommandos, die andere Gruppe *A* zur Darstellung der Adresse dient. Das Grundkommando umfaßt den wesentlichen Teil des Gesamtkommandos, z. B. »Ablesen«, »Speichern«, oder »Rechenoperation«, wie z. B. »Multiplikation«, während die Adresse gleichzeitig das Ziel angibt, z. B. das Ziel des Resultates der Multiplikation, etwa wie »Speichern auf Hauptspeicher Zelle Nr. x«. Auch Übertragungskommandos können also durch die Komponentengruppe *A* gegeben werden.

Das Schlüsselprogrammwerk arbeitet also mit Mehrfachkommandos, durch die jeweils mehrere Anweisungen gleichzeitig an verschiedene Teile der Rechenmaschine gegeben werden.

Das Grundkommando läuft in den Kommandokreis IV, die Adresse dagegen in den Kommandokreis III ein, wie der Zeichnung zu entnehmen ist.

Beide Komponentengruppen eines Gesamtkommandos können nun vor dem Verlassen des Programmwerkes über Verzweigungselemente W_1, W_2 (Umschaltkontakte eines oder mehrerer Relais), die durch eine Komponentengruppe T gesteuert werden (z. B. eine Untergruppe der Komponentengruppe A, die das Schlüsselprogrammwerk nicht verläßt), unabhängig voneinander so geführt werden, daß sie das Schlüsselprogrammwerk entweder unter Umgehung des Schlüsselwerkes unverändert verlassen oder im Schlüsselwerk nach einem bestimmten Schlüssel vollständig oder unter teilweiser Umgehung des Schlüsselwerkes nur teilweise umgeschlüsselt werden.

Es besteht also beispielsweise die Möglichkeit, das Kommando »Speichern auf Zelle x« so zu variieren, daß lediglich die Zellennummer x umgeschlüsselt wird (aus x mag $x + 1$ werden), wobei das Kommando »Speichern« erhalten bleibt (teilweise Umschlüsselung).

Es ist ferner eine Steuerungsmöglichkeit des Schlüsselwerkes durch das Rechenwerk vorgesehen, so daß in Abhängigkeit von einem während der Rechnung anfallenden Resultat der Schlüssel bestimmt werden kann, nach dem die Umschlüsselung vor sich gehen soll.

Insgesamt werden folgende Gruppen von Kommandos unterschieden: Normale Kommandos, Schlüsselkommandos, Schlüsselspeicherkommandos, Sonderschlüsselspeicherkommandos.

Normale Kommandos sind alle die Kommandos, die unverändert (z. B. als Operationskommandos) das Schlüsselwerk durchlaufen und an die zu steuernde Anlage gehen.

Schlüsselkommandos sind solche Kommandos, die ganz oder teilweise variiert werden können, wobei die Variation nach einem Schlüssel erfolgt, der vor dem Ablauf der KF in ein Schlüsselwerk gegeben wird.

Schlüsselspeicherkommandos dienen zur Einstellung der Schlüssel im Schlüsselwerk, so daß die Schlüsselkommandos entsprechende Variationen erfahren.

Sonderschlüsselspeicherkommandos dienen zur Übertragung von Schlüsseln, die auf Grund von vorausgehenden Normalkommandos in der zu steuernden Anlage zuvor ermittelt werden, auf das Schlüsselwerk und vermitteln die Steuerung des Schlüsselwerkes vom Rechenwerk aus.

Eine Variabilität der Gliedzahl beider Komponentengruppen, d. h. des Umfanges einer Teilumschlüsselung, wobei z. B. die Gesamtgliedzahl der Ja-Nein-Wert-Kombinationen konstant bleiben kann, wird dadurch erzielt, daß die Verzweigungselemente W_1 und W_2 nicht nur gruppenweise geschlossen, sondern in Abhängigkeit von einer in der Komponentengruppe T vorliegenden Kombination einzeln auf »Schlüsselwerk« bzw. »Umgehung« geschaltet werden können.

Die Aufteilung der vom Schlüsselwerk kommenden Kommandos in die beiden Kreise III und IV bedeutet also keine grundsätzliche und starre Zuordnung der betreffenden Komponenten eines Gesamtkommandos zu einem der beiden Kreise. Die Aufteilung kann vielmehr je nach Erfordernis jeweils entsprechend verschieden sein.

Adressen und gegebenenfalls Operationskommandos bzw. Übertragungskommandos (Kommandokreis III) und Grundkommandos (Kommandokreis IV) laufen zu den Rechenwerken bzw. den Speichervorrichtungen der Wertekreisläufe I und II. Die auslaufenden Wirkungspfeile bedeuten, daß dort ankommende Kommandos jeweils steuernd auf die betreffende Vorrichtung einwirken, also Rechenoperationen und den Werteverkehr in dem angesprochenen Wertekreislauf steuern.

Das Programmwerk der Rechenanlage des Ausführungsbeispiels arbeite beispielsweise mit nachstehenden Kommandos:

Operationskommandos

für das Rechenwerk RW 1	für das Rechenwerk RW 2
$\nearrow +$ (eingeben von $+$)	$a + b$
$a \vee \overline{b}$	$a - b$
$a \vee b$	$a \times b$
$a \wedge \overline{b}$	$a : b$
$a \wedge b$	$a \times (-1)$
$a \sim b$	$a \times 1/2$
$a \nsim b = \overline{a \sim b}$	$a \times 2$
$a \rightarrow b$	$a = b$
	$a > b$
	$a \geqslant b$

Adressenkommandos

III {
Rechenwerk RW 1 und Rechenoperationen,
Komponentenspeicher und gegebenenfalls Übertragungskommandos,
Rechenwerk RW 2 und Rechenoperationen,
Hauptspeicher, Zelle x und gegebenenfalls Übertragungskommandos,
Schlüsselspeicher und gegebenenfalls Übertragungskommandos,
Schlüsselspeicher, Zelle x.

Grundkommandos

IV {
»Eingeben«,
»Anzeigen«,
»Speichern«,
»Ablesen«,
»Operation«,
»Resultat halten«,
»Schlüsselspeicherkommando«,
»Sonderschlüsselspeicherkommando«.

Sonderkommandos

T {
Schlüsselspeicherkommando,
Sonderschlüsselspeicherkommando (Teilumschlüsselung).

Der Ablauf eines Rechenprogramms im Programmwerk kann über sogenannte Bedingungsleitungen Bn, Ba gesteuert werden, die Wirkungsverbindungen, beispielsweise in Form von elektrischen Leitungen oder von Schaltgliedern, darstellen, über die das Ergebnis von Operationen im Rechenwerk *RW* 2 (*Bn*), deren Ergebnis einen Ja-Nein-Wert darstellt (z. B. Aussagen über Zahlen oder Zahlenpaare, wie $a > b$), als Ja-Nein-Wert in den aussagenlogischen Wertekreislauf eingeführt werden kann. Ebenso kann über die Bedingungsleitung *Ba* vom Rechenwerk *RW* 1 zum Programmwerk das Ergebnis einer aussagenlogischen Rechenoperation als Ja-Nein-Wert dem Programmwerk zugeleitet werden, dessen Programmablauf dadurch beeinflußt wird. So wird z. B. dadurch die Ausführung eines bedingten Kommandos ermöglicht bzw. verhindert.

Über die Bedingungsleitung Bn_1 kann z. B. das Ergebnis der numerischen Rechenoperation $a > b$ dem Programmwerk direkt zur Steuerung des Ablaufwerkes mitgeteilt werden.

Bei nicht erfüllter Bedingung wird beispielsweise nicht das bedingte Kommando, sondern sofort das nächstfolgende Kommando des Rechenprogramms ausgeführt. Bedingte Kommandos spielen eine große Rolle bei iterativen Operationsfolgen, d. h. solchen Kommandofolgen, die mehrfach mit jeweils unterschiedlichen Ausgangswerten so oft durchgerechnet werden, bis z. B. eine bestimmte Genauigkeit bzw. eine gewisse Anzahl von Wiederholungen erreicht ist.

Im folgenden wird als Beispiel eine explizite Rechenanweisung und daraus der entsprechende Rechenplan für die Rechenanlage des Ausführungsbeispiels entwickelt. Es handele sich bei dieser Rechenanweisung um die Aufgabe, eine unübersichtliche, komplizierte mathematische Rechenanweisung vor der Entwicklung des Rechenplanes und der numerischen Durchführung auf ihre formale Richtigkeit hin zu untersuchen. Das Ergebnis einer solchen Untersuchung ist ein Ja-Nein-Wert, der die Frage beantwortet, ob für die eine Zeichenfolge darstellende mathematische Formel (Rechenanweisung) das Prädikat $Sa(x)$ zutrifft, das besagt »x ist eine sinnvolle Zeichenfolge«. Der Ausdruck $Sa(x)$ muß zunächst in Form einer Rechenanweisung so formuliert werden, daß eine Rechenmaschine danach arbeiten kann.

Die zu untersuchende mathematische Formel stellt eine Zeichenfolge mit folgenden Zeichen dar: Variablenzeichen, Negationszeichen, Operationszeichen und Klammerzeichen. Die einzelnen Zeichen werden durch Folgen von Ja-Nein-Werten verschlüsselt dargestellt. Es werden beispielsweise acht Ja-Nein-Werte (Komponenten $K_0 \ldots K_7$) zur Darstellung eines jeden Zeichens benutzt und folgender Schlüssel festgelegt:

Komponente 0	1	2	3	4	5	6	7	
o	o	o	o	o	o	o	+	$Va(x)$ Variablenzeichen,
o	o	o	o	—	—	+	—	$Neg(x)$ Negationszeichen,
o	o	o	o	+	—	+	—	$Kla(x)$ Klammer-auf-Zeichen,
o	o	o	o	+	+	—	—	$Klz(x)$ Klammer-zu-Zeichen,
o	o	o	o	+	—	—	—	$Op(x)$ Operationszeichen.

o bedeutet, daß die betreffende Stelle indifferent ist, also + oder — sein kann. Diese Stellen sind zur Kennzeichnung bzw. Unterscheidung z. B. der verschiedenen Variablen- oder Operationszeichen usw. verfügbar.

Für die einzelnen Zeichenarten werden Prädikatensymbole, wie z. B. $Va(x)$, d. h. x ist eine Variable, eingeführt, deren exakte Definition entsprechend dem festgelegten Schlüssel durch eine aussagenlogische Rechenanweisung gegeben ist, z. B.

$$K_4 \wedge \overline{K_5} \wedge \overline{K_6} \wedge \overline{K_7} \Rightarrow Op(x)$$

oder

$$K_7 \Rightarrow Va(x)$$

$\Rightarrow$ bedeutet: ergibt.

Zwecks schrittweiser Untersuchung einer Zeichenfolge ist die Einführung weiterer Hilfsprädikate erforderlich:

$Az(x)$ äq $Va(x) \vee Neg(x) \vee Kla(x)$

$Sz(x)$ äq $Va(x) \vee Klz(x)$

$Sq(x, y)$ äq $-(Sz(x) \sim Az(y))$

$Az(x)$: x ist ein Zeichen, das am Anfang eines sinnvollen Ausdrucks stehen kann;

$Sz(x)$: x ist ein Zeichen, das am Ende eines sinnvollen Ausdrucks stehen kann;

$Sq(x, y)$: innerhalb eines sinnvollen Ausdrucks kann das Zeichen y auf das Zeichen x folgen.

Es gelten folgende Ansätze:

$$K_7 \vee (\overline{K_5} \wedge K_6 \wedge \overline{K_7}) \Rightarrow Az(x)$$

$$K_7 \vee (K_4 \wedge K_5 \wedge \overline{K_6} \wedge \overline{K_7}) \Rightarrow Sz(x)$$

$$-\Big(\big(K_7 \vee (K_4 \wedge K_5 \wedge \overline{K_6} \wedge \overline{K_7})\big) \sim \big(K_7' \vee (\overline{K_5'} \wedge K_6' \wedge \overline{K_7'})\big)\Big) \Rightarrow Sq(x, x')$$

Die explizite Rechenanweisung für die Untersuchung des Prädikats $Sa(x)$, eine Ableitung und nähere Untersuchung (vgl. Arch. f. Math. Bd. I, H. 6) wird hier übergangen, kann dann im Formalismus des allgemeinen Plankalküls wie folgt angesetzt werden:

(1)

$$\begin{array}{c|ccc|}
 & R\,(V) & \Rightarrow & R \\
V & 0 & & 0 \\
S & m\,\sigma & & 0
\end{array}$$

Struktur $\sigma = 1 \cdot 8$, d. h. jedes Glied der Liste $\underset{0}{v}$ besteht aus einer Folge von acht Ja-Nein-Werten.

(2)

$$\begin{array}{c|ccc|}
 & Az\,(V) & \Rightarrow \wedge & R \\
V & 0 & & 0 \\
K & 0 & & \\
S & \sigma & & 0
\end{array}$$

(3)

$$\begin{array}{ccc|}
V & \Rightarrow & Z \\
0 & & 0 \\
0 & & \\
\sigma & & \sigma
\end{array}$$

(4)

$$\begin{array}{c}
0 \Rightarrow \varepsilon \\
\\
1 \cdot n
\end{array}$$

$$
W\left[\begin{array}{l}
(5)\quad \begin{array}{c|ccccccccc}
W & \mu x & \Big[& x \in V & \wedge & x \neq V & \Big] & \Rightarrow & Z \\
V & & & 0 & & 0 & & & 1 \\
K & & & & & 0 & & & \\
S & \sigma & & m\,\sigma & & \sigma & & & \sigma
\end{array}
\qquad
(6)\quad \begin{array}{ccccc}
Sq\,(Z, & Z) & \Rightarrow & \wedge & R \\
0 & 1 & & & 0 \\
\sigma & \sigma & & & 0
\end{array}
\\[2ex]
(7)\quad \begin{array}{c|ccc}
 & Kla\,(Z) & \longrightarrow & (\varepsilon + 1 \Rightarrow \varepsilon) \\
V & 1 & & \\
S & \sigma & &
\end{array}
\qquad
(8)\quad \begin{array}{ccc}
Klz\,(Z) & \longrightarrow & (\varepsilon - 1 \Rightarrow \varepsilon) \\
1 & & \\
\sigma & &
\end{array}
\\[2ex]
(9)\quad \begin{array}{c|cccc}
 & \varepsilon \geqslant 0 & \Rightarrow & \wedge & R \\
V & & & & 0 \\
S & & & & 0
\end{array}
\qquad
(10)\quad \begin{array}{ccc}
Z & \Rightarrow & Z \\
1 & & 0 \\
\sigma & & \sigma
\end{array}
\end{array}\right]
$$

$$
(11)\quad \begin{array}{c|cccc}
 & Sz\,(Z) & \Rightarrow & \wedge & R \\
V & 0 & & & 0 \\
S & \sigma & & & 0
\end{array}
\qquad
(12)\quad \begin{array}{cccc}
\varepsilon = 0 & \Rightarrow & \wedge & R \\
 & & & 0 \\
 & & & 0
\end{array}
$$

Um dieses Beispiel einer im allgemeinen Plankalkül aufgestellten Rechenvorschrift verständlich zu machen, sei einiges über die Darstellung im allgemeinen gesagt.

Die den einzelnen Variablen zugeordneten Indizes sind nach Zeilen geordnet. In der ersten Zeile, der Hauptzeile, stehen die einzelnen Variablen- und Formelzeichen. Variable werden mit V, Zwischenwerte mit Z, Resultatwerte mit R bezeichnet. Hierbei ist zu beachten, daß Variable, Zwischenwerte und Resultatwerte im Plankalkül allgemein von beliebiger Struktur sein können (z. B. Ja-Nein-Werte, Zahlen, Zeichen, Zeichenfolgen, Listen, Paarlisten usw.) und ein Rechenplan beliebig viele Variablen und Resultatwerte haben kann.

In der zweiten Zeile, der V-Zeile, stehen die den Variablen, Zwischenwerten und Resultatwerten zugeordneten Indizes, welche diese voneinander unterscheiden. Sie entsprechen etwa der fortlaufenden Numerierung der Werte in einer numerischen Formel.

In der dritten Zeile, der K-Zeile, steht der Komponentenindex, welcher angibt, welche Komponente der betreffenden Angabe gemeint ist.

In der vierten Zeile, der S-Zeile, steht die Kennzeichnung der Struktur der betreffenden Angabe (z. B. 0 bedeutet einen Ja-Nein-Wert, $1 \cdot n$ eine Folge von n Ja-Nein-Werten, σ ein sogenanntes variables Strukturzeichen, für welches eine geeignete Struktur eingesetzt werden kann).

Der gesamte Rechenplan zerfällt in verschiedene Rechenplangleichungen. Das Ergibtzeichen $\Rightarrow$ verbindet einen zu errechnenden Ausdruck (links) mit dem Resultat (rechts). Links steht also stets eine Rechenvorschrift, in welche Variable oder Zwischenwerte als bestimmende Größen eingehen, rechts ein Endresultat. Die Strukturen der einzelnen Werte können dabei sehr verschieden sein.

Einer Rechenvorschrift geht ein Randauszug voraus, aus welchem die eingehenden Variablen und die zu errechnenden Resultate erkenntlich sind.

Für das spezielle Beispiel gilt:

(1) der Randauszug besagt, daß als Variable eine Angabe V_0 von der Struktur $m\sigma$ in die Rechnung eingeht und ein Resultatwert R_0 von der Struktur 0 errechnet wird. ($m\sigma$ = Folge von m Zeichen der Struktur σ, 0 Struktur eines Ja-Nein-Wertes.) Das heißt, daß einer Folge von m Zeichen der Struktur σ ein Ja-Nein-Wert, also ein Prädikat, zugeordnet werden soll. Die in die Rechenvorschrift eingehende Variable ist also die gesamte zu untersuchende Formel (Zeichenfolge $m\sigma$), welche nicht mit dem einzelnen Variablenzeichen verwechselt werden darf. Das Resultat R_0 ist in diesem Fall ein Ja-Nein-Wert, der den Wahrheitswert der Aussage darstellt: V_0 ist eine sinnvolle Zeichenfolge.

(2) besagt, daß die Komponente 0 der gegebenen Zeichenfolge V_0 die Eigenschaft Az haben muß. Der Ausdruck $\Rightarrow \wedge R$ bedeutet: ergibt ein Konjunktionsglied zur Bestimmung von R, d. h. ist notwendige Bedingung für R.

Die Hauptrechnung besteht dann in einem Wiederholungsplan W, in welchem die Folgebedingung $Sq\ (Z_0, Z_1)$ zweier aufeinanderfolgender Zeichen geprüft und die Klammerbilanz ε gebildet wird. Dieser Vorgang muß durch zwei Ansätze (3) und (4) vorbereitet werden.

(3) besagt, daß das erste Glied von V_0 das erste Z_0 ergibt.

(4) besagt, daß der Hilfswert ε am Anfang 0 zu setzen ist.

Die Vorschriften (5) bis (10) liegen im zu wiederholenden Teil.

(5) ist ein Ansatz mit dem Operator μx, welcher bedeutet: das nächste Glied von folgender Eigenschaft. $\mu x A\ (x)$ ist der Befehl, stets das nächste noch nicht ermittelte Glied der Eigenschaft A herauszusuchen; die Untersuchung wird abgebrochen, sobald kein derartiges Glied vorhanden ist. In diesem Beispiel besagt die Forderung, daß x von der Struktur σ, also eines einzelnen Zeichens in der Zeichenfolge V_0 enthalten sein muß, aber nicht gleich dem ersten Zeichen sein darf. Das so ermittelte Zeichen ergibt den jeweiligen Zwischenwert Z_1.

(6) besagt, daß das Prädikat $Sq\ (Z_0, Z_1)$ notwendige Bedingung für das Resultat R_0 ist.

(7) und (8) sind die Ansätze zur Bildung des Zwischenwertes. Bei einem Klammer-auf-Zeichen wird ε

um 1 erhöht, bei einem Klammer-zu-Zeichen um 1 erniedrigt. Das Zeichen $\xrightarrow{\cdot}$ steht vor bedingten Planteilen. Die hinter diesem Zeichen stehende Rechenplangleichung ist nur durchzurechnen, falls die davorstehende Bedingung erfüllt ist. Die Gleichung $\varepsilon + 1 \Rightarrow \varepsilon$ ist wie folgt zu lesen: der alte ε-Wert, erhöht um 1, ergibt den neuen ε-Wert.

(9) ist die Bedingung, daß ε stets größer oder gleich 0 sein muß.

(10) besagt, daß der bisherige Zwischenwert Z_1 den neuen Zwischenwert Z_0 ergibt, wonach die Rechnung wieder bei (5) weiterläuft.

Sind alle in V_0 enthaltenen Werte erschöpft, so wird zu (11) übergegangen. Dieser Ansatz besagt, daß das letzte Glied der Zeichenfolge die Eigenschaft Sz haben muß. Dieses letzte Glied ist gleich dem bei der letzten Durchrechnung des Wiederholungsplanes ermittelten Z_0.

Schließlich stellt der Ansatz (12) die Bedingung dar, daß die Klammerbilanz ε am Schluß gleich 0 sein muß.

Zunächst werden die Rechenpläne für die einzelnen Prädikate $Va\,(x)$, $Neg\,(x)$ usw. aufgestellt:

$P\,22\ Va\,(x);\ K_7 \Rightarrow Va\,(x) \quad Sp\ 1 \cdot 10\ Va\,(x) \Rightarrow R\,22$

$P\,23\ Neg(x);\ \overline{K_4} \wedge \overline{K_5} \wedge K_6 \wedge \overline{K_7} \Rightarrow Neg(x)$

$Sp\ 1 \cdot 0 \ldots 7/\text{II}\,\ddot{u}$

$\nearrow +$

$\wedge\ \overline{K_4}$

$\wedge\ \overline{K_5}$

$\wedge\ K_6$

$\wedge\ \overline{K_7}$

$S\ 1 \cdot 11\ Neg(x) \Rightarrow R\,23$

$P\,24\ Kla(x);\ K_4 \wedge \overline{K_5} \wedge K_6 \wedge \overline{K_7} \Rightarrow Kla(x)$

$Sp\ 1 \cdot 0 \ldots 7/\text{II}\,\ddot{u}$

$\nearrow +$

$\wedge\ K_4$

$\wedge\ \overline{K_5}$

$\wedge\ K_6$

$\wedge\ \overline{K_7}$

$Sp\ 1 \cdot 12\ Kla(x) \Rightarrow R\,24$

$P\,25\ Klz(x);\ K_4 \wedge K_5 \wedge \overline{K_6} \wedge \overline{K_7} \Rightarrow Klz(x)$

$Sp\ 1 \cdot 0 \ldots 7/\text{II}\,\ddot{u}$

$\nearrow +$

$\wedge\ K_4$

$\wedge\ K_5$

$\wedge\ \overline{K_6}$

$\wedge\ \overline{K_7}$

$Sp\ 1 \cdot 13\ Klz(x) \Rightarrow R\,25$

$P\,26\ Op\,(x);\ K_4 \wedge \overline{K_5} \wedge \overline{K_6} \wedge \overline{K_7} \Rightarrow Op\,(x)$

$Sp\ 1 \cdot 0 \ldots 7/\text{II}\,\ddot{u}$

$\nearrow +$

$\wedge\ K_4$

$\wedge\ \overline{K_5}$

$\wedge\ \overline{K_6}$

$\wedge\ \overline{K_7}\ \ — \xrightarrow{\cdot} \left[— \Rightarrow \underset{0}{R} \,\middle|\, \text{Fin} \right]$

$Sp\ 1 \cdot 14\ Op\,(x) \Rightarrow R\,26$

$P\,27\ Az\,(x);\ K_7 \vee (\overline{K_5} \wedge K_6 \wedge \overline{K_7}) \Rightarrow Az\,(x)$

$Sp\ 1 \cdot 0 \ldots 7/\text{II}\,\ddot{u}$

$\nearrow +$

$\wedge\ \overline{K_5}$

$\wedge\ K_6$

$\wedge\ \overline{K_7}$

$\vee\ K_7$

$Sp\ 1 \cdot 15\ Az\,(x) \Rightarrow R\,27$

$P\,28\ Sz\,(x);\ K_7 \vee (K_4 \wedge K_5 \wedge \overline{K_6} \wedge \overline{K_7}) \Rightarrow Sz\,(x)$

$Sp\ 1 \cdot 0 \ldots 7/\text{II}\,\ddot{u}$

$\nearrow +$

$\wedge\ K_4$

$\wedge\ K_5$

$\wedge\ \overline{K_6}$

$\wedge\ \overline{K_7}$

$\vee\ K_7$

$Sp\ 1 \cdot 16\ Sz\,(x) \Rightarrow R\,28$

$P\,29\ Sq\,(x, y)$

$— (Sz\,(x) \sim Az\,(y)) \Rightarrow Sq\,(x, y)$

$A\,2 \cdot x$

$P\,28$

$A\,2 \cdot y$

$P\,27$

$\nearrow +$

$\wedge\ 1 \cdot 16\ (Sz\,(x) \Rightarrow R\,28)$

$\nrightarrow 1 \cdot 15\ (Az\,(y) \Rightarrow R\,27)$

$Sp\ 1 \cdot 17\ Sq\,(x, y) \Rightarrow R\,29$

Die Zellenbelegung der Speicherwerke werde wie folgt festgelegt:

Hauptspeicher 2

$2 \cdot 0$	V Liste der zu untersuchenden Zeichenfolge
·	0
·	$0 \ldots m - 1$
·	
·	
·	
·	Struktur $\sigma = 1 \cdot 8$, d. h. eine Folge von
$m - 1$	acht Ja-Nein-Werten
$2 \cdot 128$	Z_0
$2 \cdot 129$	ε Anzahl der Klammerzeichen, anfangs $= 0$
$2 \cdot 130$	ε_1 laufende Nummer des Listengliedes $V_0 \cdot \varepsilon_1$ der Liste V_0
$2 \cdot 131$	Z_1
$2 \cdot 132$	$m - 1$ Index des letzten Gliedes

Komponentenspeicher 1

1 · 0 ... 7	$K_0 \ldots 7$	Komponente 0 ... 7 eines Listengliedes $V_0 \cdot \varepsilon_1$
1 · 10	*Va* (*x*)	
1 · 11	*Neg* (*x*)	
1 · 12	*Kla* (*x*)	
1 · 13	*Klz* (*x*)	
1 · 14	*Op* (*x*)	
1 · 15	*Az* (*x*)	
1 · 16	*Sz* (*x*)	
1 · 17	*Sq* (*x*, *x'*)	
1 · 20 ...	Zwischenwerte der Rechnung	

Der Gesamtrechenplan zur Untersuchung des Prädikats *Sa* (*x*) für die Zeichenfolge der Rechenanweisung kann nun wie folgt aufgestellt werden:

Rechenplan

(1)	Start	
	↗ v_0	Liste v_0 der zu untersuchenden Zeichenfolge
	Sp 2 · 0 ... *m* — 1	Jedes Glied dieser Liste stellt ein Zeichen dar, verschlüsselt durch acht Ja-Nein-Werte (Struktur $\sigma = 1 \cdot 8$)
	↗ *m* — 1	*m* Anzahl der Listenglieder
	Sp 2 · 132	
(2a)	*A* 2 · 0 *Az* (*x*) *P* 27	
(2b)	$R\ 27 = \text{—} \longrightarrow \left[\begin{array}{c\|c} \text{—} \Rightarrow R & \mathrm{Fin}^2 \\ 0 & \\ 0 & \end{array}\right]$	
(3)	*A* 2 · 0 *Az* (*x*) *Sp* 2 · 128	
(4)	↗ $\varepsilon = 0$	Anzahl der Klammerzeichen
	Sp 2 · 129	
	↗ $\varepsilon_1 = 0$	ε_1 (laufende Nummer des Listengliedes $v_0 \cdot \varepsilon_1$ der Liste v_0)
	Sp 2 · 130	
	Rz 5a	*Rz* 5a Rufzeichen, das den Plan (5a) zum Ablauf anruft
(5a)	*A* 2 · 130/*RW* 2 + 1 *Sp* 2 · 130 *S* · *Schl* · *Sp* · *Kdo* O (*A* 2 · ε_1) *Rz* 5b	
(5b)	*Schl* · *Kdo* O (*A* 2 · ε_1) *Sp* 2 · 131 *Rz* 6	
(6a)	*A* 2 · 128 *P* 28 *A* 2 · 131 *P* 27 ↗ + ∧ 1 · 16 ↛ 1 · 15 *Rz* 7a	
(6b)	$R\ 29 = \text{—} \longrightarrow \left[\begin{array}{c\|c} \text{—} \Rightarrow R & \mathrm{Fin}^3 \\ 0 & \\ 0 & \end{array}\right]$	

(7a)	*A* 2 · 131 *P* 24 *Rz* 8a	(7b)	*Kla* $\longrightarrow$ [*A* 2 · 129 + 1 *Sp* 2 · 129 *Rz* 8a]
(8a)	*A* 2 · 131 *P* 25 *Rz* 9	(8b)	*Klz* $\longrightarrow$ [*A* 2 · 129 — 1 *Sp* 2 · 129 *Rz* 9]
(9)	*A* 2 · 129		$\varepsilon < 0 \longrightarrow$ | *P* 6b
(10)	*A* 2 · 131 *Sp* 2 · 128 *A* 2 · 130 *A* 2 · 132 *Rz* 5a		$\varepsilon_1 = m - 1 \longrightarrow$ *Rz* 11
(11)	*A* 2 · 128 *P* 28		*R* 28 = — $\longrightarrow$ *Rz* 2b
(12)	*A* 2 · 129 + ⇒ *R* 0 0 Fin		$\varepsilon \neq 0 \longrightarrow$ *Rz* 2b

Als zweites Beispiel werde eine Aufgabe aus dem Gebiet der theoretischen Logik behandelt, die unter Anwendung des Operators »Diejenigen, welche ...« gelöst wird. Beide Beispiele sollen die Vielseitigkeit einer erfindungsgemäß ausgebildeten Rechenvorrichtung veranschaulichen, die in entsprechender Weise auch alle Aufgaben der theoretischen Logik mit All- und Existenzoperatoren, mit dem Operator »Derjenige, welcher ...« und dem μ-Operator »Das nächste Glied mit der Eigenschaft ...« zu lösen vermag.

Das Rechenbeispiel laute: Es existiere eine Liste v_0 mit *m* Gliedern, von denen jedes als Komponente von v_0 eine durch acht Ja-Nein-Werte verschlüsselte allgemeine Angabe darstellt. Die Struktur der Komponenten der Liste v_0 ist also von der Art $\sigma = 1 \cdot 8$. Aus dieser Liste v_0 sollen diejenigen Glieder ausgewählt werden, welche eine bestimmte, durch das Prädikatensymbol *R* □ gekennzeichnete Eigenschaft besitzen, wobei das Prädikat *R* □ ebenfalls durch eine Ja-Nein-Werte-Kombination definiert ist. □ ist ein Leerstellenzeichen, für das eine beliebige Prädikatbezeichnung eingesetzt werden kann.

Zur Durchführung der Aufgabe wird zunächst die Rechenanweisung und aus dieser der entsprechende Rechenplan, nach dem die Rechenmaschine arbeiten soll, entwickelt.

Die Rechenmaschine arbeitet in der Weise, daß sämtliche Glieder der Liste, angefangen mit dem ersten Glied (Komponente 0), systematisch untersucht werden, ob auf sie das Prädikat *R* □ zutrifft.

Die richtigen Glieder der Liste v_0 werden schrittweise zur Bildung einer Aufbauliste herangezogen, die dadurch gebildet wird, daß diese Glieder der Reihe nach bei fortlaufender Zellenbelegung, gesteuert durch die Rechenmaschine, im Hauptspeicher gespeichert werden.

Die Rechenanweisung kann allgemein formal wie folgt angesetzt werden:

$$\hat{\hat{x}} \left(x \in \underset{0}{v} \wedge R \,\square\, (x)\right) \Rightarrow \underset{0}{R}$$

$\hat{\hat{x}}$ bedeutet: diejenigen x, welche der durch den Klammerausdruck gegebenen Bedingung genügen (unter Beibehaltung mehrfach auftretender Glieder).

$x \in \underset{0}{v}$ wird gelesen: x ist Glied der Liste $\underset{0}{v}$

$R \,\square\, (x)$ wird gelesen: x besitzt die Eigenschaft $\square$

Der entsprechende allgemeine Rechenplan lautet

$0 \Rightarrow \varepsilon$

$$\begin{array}{l|l} & W\,1\left(N\left(\underset{0}{v}\right)\right)\left[\begin{array}{c} R\,\square\,(V) \dashrightarrow \\ 0 \\ i \end{array}\left[\begin{array}{cc|c} v \Rightarrow & R & \varepsilon + 1 \Rightarrow \varepsilon \\ 0 & 0 & \\ i & \varepsilon & \end{array}\right]\right] \\ V & \\ K & \end{array}$$

Es bedeutet

$N\left(\underset{0}{v}\right)$	Anzahl der Listenglieder,
$W\,1$	Wiederholungsplan (Anzahl der Wiederholungen gleich der der Listenglieder),
$R\,\square\left(\underset{\substack{0\\ i}}{v}\right) \dashrightarrow$	ergibt die Untersuchung, daß das Listenglied $v_{0 \cdot i}$ (Komponente i der Liste $\underset{0}{v}$) die Eigenschaft $\square$ hat, dann läuft der rechts neben dem $\dashrightarrow$ Zeichen in [ ] stehende Rechenplan ab,
$\underset{\substack{0\\ \varepsilon}}{R}$	Listenglied ε der neu zu bildenden Aufbauliste $\underset{0}{R}$ (Komponente ε der Aufbauliste $\underset{0}{R}$)
ε	Hilfswert zur Numerierung der Aufbaulistenglieder,
V	bezeichnet die Zeile der Variablenindizes,
K	bezeichnet die Zeile der Komponentenindizes,
$\varepsilon + 1 \Rightarrow \varepsilon$	der alte ε-Wert plus 1 ergibt den neuen ε-Wert.

Der Definitionsansatz für das Prädikat $R\,\square\,(x)$ laute beispielsweise

$$K_4(x) \vee \left(\overline{K_5}(x) \wedge K_6(x)\right) \Rightarrow R\,\square\,(x)$$

und der entsprechende Rechenplan

$P0$	$Sp\ 1 \cdot 0 \ldots 7/\mathrm{II}\,\ddot{u}$	
	$\nearrow +$	(Eingeben von $+$)
	$\wedge\ \overline{K_5}$	
	$\wedge\ K_6$	$\dashrightarrow (Rz\ P\,2)$
	$\vee\ K_4$	
	$A\ 1 \cdot 0 \ldots 7/\mathrm{II}\,\ddot{u}$	

Der Gesamtrechenplan kann wie folgt aufgestellt werden:

Gesamtrechenplan

$P\,1$	Start	
	v $0 \cdot 0 \ldots m - 1$	Eingeben der Listenglieder $\underset{0 \cdot 0 \ldots m-1}{v}$
		Anzahl der Glieder $= N\left(\underset{0}{v}\right) = m$
	$Sp\ 2 \cdot 0 \ldots m - 1$	Speichern der Liste $\underset{0}{v}$ im Hauptspeicher, Zelle $0 \ldots m - 1$
	$\nearrow \varepsilon = m - 1$ $Sp\ 2 \cdot 2\,m$	ε Speicherzellen-Nr. des Gliedes $\underset{0}{R}$ der Aufbauliste (Laufwert ε)
	$\nearrow i = -1$ $Sp\ 2 \cdot 2 \cdot m + 1$	i Hilfswert zur Kennzeichnung des zu untersuchenden Listengliedes (Laufwert i)
	$\nearrow m - 1$ $Sp\ 2 \cdot 2 \cdot m + 2$	$m - 1$ Komponentenindex des letzten Gliedes der Liste v_0
	$Rz\ P\,2$	Rufzeichen des Planes $P\,2$

$P\,2$	$A\ 2 \cdot 2\,m + 1/RW\,2$ $+\ 1$ $R \cdot h$	Bildung des Laufwertes i
	$Sp\ 2 \cdot 2\,m + 1$ S. Schl. Sp. Kdo O (III $\ddot{u}$/Ablesen $2 \cdot i$)	
	$A\ 2 \cdot 2\,m/RW\,2$ $+\ 1$ $R \cdot h$	Bildung des Laufwertes ε
	$Sp\ 2 \cdot 2\,m$ S. Schl. Sp. Kdo 1 (III $\ddot{u}$/Speichern $2 \cdot \varepsilon$) $Rz \cdot P\,3$	

$P\,3$	Schlüsselkommando O (Ablesen $2 \cdot i$)
	P O
	Schlüsselkommando 1 (Speichern $2 \cdot \varepsilon$)
	$A\ 2 \cdot 2\,m + 2$
	$A\ 2 \cdot 2\,m + 1$
	$i < m - 1 \dashrightarrow Rz\ P\,2$
	Fin 3

PATENTANSPRÜCHE:

1. Rechenanlage zur Durchführung aussagenlogischer und numerischer Rechenoperationen, gekennzeichnet durch die Kombination folgender Einzelvorrichtungen: a) Rechenvorrichtung für aussagenlogische Rechenoperationen nach den den Grundoperationen des Aussagenkalküls entsprechenden Rechenvorschriften, b) Rechenvorrichtung für numerische Rechenoperationen nach den Rechenvorschriften der Arithmetik, c) Programmwerk (mit oder ohne Schlüsselwerk) zur Steuerung beider Rechenvorrichtungen nach Maßgabe eines Rechenprogramms.

2. Rechenanlage nach Anspruch 1, dadurch gekennzeichnet, daß Übertragungsmittel, wie z. B. Verbindungsleitungen oder Schaltglieder und vom Programmwerk gesteuerte Schaltelemente, von der numerischen Rechenvorrichtung zur aussagenlogischen Rechenvorrichtung vorgesehen sind, mittels deren Angaben (z. B. Ja-Nein-Wert-Kombinationen, Zahlen od. ä.) von dem Wertekreislauf der numerischen Rechenvorrichtung in den Wertekreislauf der aussagenlogischen Rechenvorrichtung bzw. analog umgekehrt übergeführt werden können.

3. Rechenanlage nach Anspruch 1 und 2, dadurch gekennzeichnet, daß zur Herstellung der Wirkungsverbindung zur Werteüberführung von dem Wertekreislauf der aussagenlogischen Rechenvorrichtung in den der numerischen Rechenvorrichtung ein oder mehrere Komponentenspeicher eingeschaltet sind, in deren einzelnen Speicherzellen jeweils nur eine Komponente (ein Ja-Nein-Wert) einer aus mehreren Komponenten bestehenden Angabe steht, so daß zwecks Durchführung aussagenlogischer Rechenoperationen jede Komponente einzeln, bei Überführung der vollständigen Angabe in den Wertekreislauf der numerischen Rechenvorrichtung jedoch alle zu einer Angabe gehörenden Komponenten zusammen abgelesen und analog umgekehrt bei Überführung einer vollständigen, aus Ja-Nein-Werten bestehenden Angabe vom Wertekreislauf der numerischen Rechenvorrichtung in den des aussagenlogischen Wertekreislaufes gleichzeitig zusammen, aber jede Komponente jeweils in nur einer Zelle des Komponentenspeichers gespeichert werden können.

4. Rechenanlage nach Anspruch 1 bis 3, dadurch gekennzeichnet, daß Übertragungsmittel, wie z. B. Verbindungsleitungen oder Schaltglieder und vom Programmwerk gesteuerte Schaltelemente, von dem numerischen Wertekreislauf zum Programmwerk vorgesehen sind (z. B. zu dem oder den Schlüsselspeichern des Schlüsselwerkes), mit deren Hilfe in der numerischen oder bzw. und aussagenlogischen Rechenvorrichtung errechnete Angaben bzw. Werte in das Programmwerk übergeführt werden (z. B. durch Sonder- oder Teilschlüsselkommandos) und den Ablauf des Rechenprogramms entsprechend beeinflussen können.

5. Rechenanlage nach Anspruch 1 bis 4, dadurch gekennzeichnet, daß eine Bedingungswirkungsverbindung vom numerischen Rechenwerk zum Komponentenspeicher bzw. aussagenlogischen Rechenwerk vorgesehen ist, über die das Ergebnis von Operationen, deren Resultat ein Ja-Nein-Wert ist (z. B. Aussagen über Zahlen oder Zahlenpaare, wie $a > b$), in den aussagenlogischen Wertekreislauf eingeführt werden kann.

6. Rechenanlage nach Anspruch 1 bis 5, dadurch gekennzeichnet, daß eine Bedingungswirkungsverbindung vom aussagenlogischen Rechenwerk zum Programmwerk vorgesehen ist, über die das Ergebnis einer aussagenlogischen Rechenoperation als Ja-Nein-Wert in das Programmwerk eingeleitet werden und den Programmablauf beeinflussen kann, beispielsweise durch Sperrung bzw. Freigabe eines bedingten Kommandos, z. B. über ein von diesem Ja-Nein-Wert gesteuertes Relais oder analoges Schaltelement.

7. Rechenanlage nach Anspruch 1 bis 6, dadurch gekennzeichnet, daß eine Bedingungswirkungsverbindung vom numerischen Rechenwerk zum Programmwerk vorgesehen ist, über die das Ergebnis von Operationen, deren Ergebnis ein Ja-Nein-Wert ist (z. B. Aussagen über Zahlen oder Zahlenpaare, wie $a > b$), in das Programmwerk, vorzugsweise in dessen Ablaufwerk, übertragen werden und den Ablauf des Programms beeinflussen kann, beispielsweise durch Sperrung bzw. Freigabe eines bedingten Kommandos, z. B. mittels eines von dem Übertragungswert gesteuerten Relais oder analogen Schaltelementes.

Hierzu 1 Blatt Zeichnungen

509 529 7. 55

Zu der Patentschrift **926 449**
Kl. **42m** Gr. 15