Title:	Über den Allgemeinen Plankalkül als Mittel zur Formulierung schematisch-kombinativer Aufgaben
Author(s):	Konrad Zuse
Date:	1948
Published by:	Konrad Zuse Internet Archive
Source:	Document - ZIA ID: 0242

Über den Allgemeinen Plankalkül als Mittel zur Formulierung schematisch-kombinativer Aufgaben[*] [†]

Von Konrad Zuse in Hopferau bei Füssen

Vorwort

Die Entwicklung mathematischer Rechengeräte ist in den letzten zehn Jahren erheblich vorangeschritten. Vor allem in USA und Deutschland sind die Entwicklungen unabhängig voneinander ähnliche Wege gegangen. Unter diesen Entwicklungen sind zu nennen:

1. Die mathematischen Instrumente vom Typ der Differentialanalysatoren
   - In USA: Bush-Geräte (MIT).
   - In Deutschland: Institut für Praktische Mathematik, Darmstadt, in Zusammenarbeit mit Fa. Ott, ferner Askania, jetzt fortgesetzt durch Fa. Rechenautomaten-GmbH, Göttingen.
2. Programmgesteuerte numerische Rechengeräte
   - USA: Verschiedene Geräte, deren bekanntestes ENIAC ist.
   - Deutschland: Zuse-Aparatebau Berlin, jetzt Zuse-Ingenieurbüro Hopferau.

   In den amerikanischen Geräten wurden erstmals in großem Stile Elektronenröhren zur Durchführung der Rechnungen verwand. Eine analoge Entwicklung wurde in Deutschland unabhängig davon durch Dr.-Ing. Schreyer in Zusammenarbeit mit dem Verfasser betrieben, kam jedoch wegen kriegsbedingter Schwierigkeiten nicht über das Versuchsstadium hinaus.

[*] ZIA 0242. ZuP 010/029. Version 1. Durchgesehen von R. Rojas, G. Wagner, L. Scharf.

[†] Es sei darauf hingewiesen, daß „kombinative“ Aufgaben nicht auf das Gebiet der „kombinatorischen“ Aufgaben der Mathematik beschränkt sind.

Während die unter 1. Und 2. genannten Geräte instrumentell oder ziffernmäßig der Zahlenrechnung dienen, hat der Verfasser am 24.9.1948 auf der Tagung der GAMM, Göttingen, einen neuen Gerätetyp skizziert, welcher *der automatischen Lösung schematisch-kombinativer Probleme dient, die über das Zahlenrechnen hinausgehen.*

Aufgabenstellung

Es handelt sich zunächst darum, die mathematischen Voraussetzungen für die Formulierung beliebiger schematisch-kombinativer Aufgaben in Form einer Rechenanweisung zu schaffen, nach welcher entsprechende Rechengeräte diese Aufgaben selbsttätig lösen können. Der Verfasser hat in den letzten Jahren eine Theorie des allgemeinen Rechens entwickelt, welche auf diese praktische Aufgabenstellung zugeschnitten ist. Ausgangspunkt der Theorie ist folgender Satz:

> „Rechnen heißt: Aus gegebenen Angaben nach einer Vorschrift neue Angaben bilden.“

Dementsprechend ist zunächst der Begriff der „Angabe“ zu analysieren, und danach die Gesetzt, nach denen diese Angaben zu kombinieren sind.

„*Angaben*“ können sehr verschiedenartig sein, z.B. Zahlen, Aussagen, Namen, Adressen, Koordinaten usw. Sie unterscheiden sich nach ihrer Struktur. Die einfachste Struktur hat der Ja-Nein-Wert (z.B. Vorzeichen einer Zahl, einzelne Dualziffer). Es zeigt sich, daß alle komplizierteren Angaben aus Ja-Nein-Werten aufgebaut werden können. Dementsprechend ist ein sorgfältig aufgebauter „*Strukturenkalkül*“ erforderlich.

Als „*Rechenvorschriften*“ können alle schematischen Operationen, Formeln, Ableitungen, Algorithmen, Anweisungen usw. dienen, bei denen für alle in Frage kommenden Fälle nach einer bestimmten Vorschrift aus gegebenen Ausgangsangaben bestimmte Resultatangaben abgeleitet werden können. Zu ihrer exakten Formulierung dient der „*Allgemeine Plankalkül*“. Dieser baut auf dem Aussagen- und Prädikatenkalkül der Logistik auf.

Der Unterschied zwischen dem Allgemeinen Plankalkül und den genannten logistischen Kalkülen ist folgender: Die Kalküle der Logistik dienen der impliziten Darstellung mathematischer Sätze mit dem Zweck des axiomatischen Aufbaues der Mathematik. Der Allgemeine Plankalkül dient der expliziten Formulierung von Rechenvorschriften. Eine eingehende Darstellung dieses Kalküls kann hier aus Platzgründen nicht gegeben werden. Es soll jedoch versucht werden, an Hand eines Beispieles einen Begriff dieses Kalküls zu geben.

Darstellung der Lösungsmittel an Hand eines Beispiels

Die formale Behandlung z.B. mathematischer Aufgaben besteht zum großen Teil in der nach einem Schema verlaufenden Anwendung gegebener Gesetze.

So ist die Bildung der Ableitung einer gegebenen Funktion in allen Fällen durch Gesetze festgelegt. Dadurch ist der Prozeß des Bildens der Ableitung der Mechanisierung zugänglich, wobei in diesem Zusammenhang nicht die Errechnung der Ableitung als Kurve nach Art der Differentialanalysatoren gemeint ist, sondern das Auftreten der Ableitung als Zeichenfolge an Hand der ebenfalls als Zeichenfolge gegebenen Funktion.

Ähnliche Operationen sind: Die Ermittlung von Eigenschaften von Funktionen, das Ordnen der Glieder einer Funktion nach gegebenen Gesichtspunkten, Umformen von Ausdrücken, Einsetzen von Ausdrücken an Stelle von Variablen usw.

Bei allen derartigen Operationen werden die mathematischen Ausdrücke vorteilhafterweise durch Zeichenfolgen dargestellt. Es sein nun als Beispiel die Aufgabe betrachtet, einen gegebenen aussagenlogischen Ausdruck daraufhin zu untersuchen, ob das Prädikat „Es ist eine sinnvolle Zeichenfolge“ auf ihn zutrifft. Die Ausführungen sollen hier auf den wesentlichsten Teil dieser Aufgabe beschränkt werden, der darin besteht, das Prädikat „Sinnvolle Zeichenfolge“ in Form einer Rechenanweisung zu formulieren, nach der eine Maschine die Untersuchung durchführen könnte. Der Gang der Untersuchung wird hier aus Gründen der einfachen Darstellung am Aussagenkalkül gezeigt. Selbstverständlich kann die gleiche Untersuchung auch bei den bekannten arithmetischen Formeln oder anderen Kalkülen in ähnlicher Weise durchgeführt werden.

Zuvor noch einige allgemeine Bemerkungen über die Darstellung dieses Kalküls durch Zeichenfolgen. Im wesentlichen wird mit dem Hilbertschen Formalismus gearbeitet, jedoch mit folgenden Unterschieden:

1. Operationszeichen werden nie fortgelassen,
2. die Negation wird durch Vorsetzen eines Negationszeichens vor den zu negierenden Ausdruck dargestellt,
3. es wird nur eine Art von Klammerzeichen verwendet.

Diese Art der Darstellung hat dann die Form reiner Zeichenfolgen, zum Beispiel:

$$-(a \vee b) \sim -a\& - b. \tag{1}$$

Derartig aufgebaute Ausdrücke enthalten folgende Zeichen: Variablenzeichen, Negationszeichen, Operationszeichen, Klammerzeichen und Zwischenraumzeichen, welches bei der Mechanisierung zur Abgrenzung einzelner Ausdrücke erforderlich ist. Die einzelnen Zeichen werden durch Folgen von Ja-Nein-Werten verschlüsselt dargestellt.

Es werden 8 Ja-Nein-Werte zur Darstellung jedes einzelnen Zeichens benutzt (Komponente K_0, K_1 ... K_7). Es wird festgelegt, daß ein Variablenzeichen dadurch gekennzeichnet ist, daß der letzte der 8 Ja-Nein-Werte, die Komponente K_7 positiv ist. Die Komponenten K_0 bis K_6 dienen dann als Dualzahl der Unterscheidung der einzelnen Variablen. Für Operationszeichen wird das Kriterium $K_6\& - K_7$ festgelegt. Die Komponenten K_0 bis K_5 dienen dann der Unterscheidung der einzelnen Operationszeichen. Entsprechende Festlegung für andere Zeichen. Es lassen sich dann Eigenschaften von Zeichen als aussagenlogische Formeln ansetzen

$$Op(x) = K_6(x)\& - K_7(x) \qquad \text{„x ist ein Operationszeichen“.} \tag{2}$$

Nicht jede Folge von Zeichen stellt einen sinnvollen aussagenlogischen Ausdruck dar.

Zur Durchführung der obigen Aufgabe muß zunächst das Prädikat $Sa(x)$ „x ist ein sinnvoller Ausdruck“ streng definiert werden. Dies erfolgt durch eine rekursive Definition:

1. Eine Variable ist ein sinnvoller Ausdruck.
2. Durch Vorsetzen eines Negationszeichens vor einen sinnvollen Ausdruck entsteht wieder ein sinnvoller Ausdruck.
3. Durch Zwischensetzen eines Operationszeichens zwischen zwei sinnvolle Ausdrücke entsteht wieder ein sinnvoller Ausdruck.
4. Durch Einklammern eines sinnvollen Ausdrucks entsteht wieder ein sinnvoller Ausdruck.

Diese Regeln lassen sich mit Hilfe des Prädikatenkalküls formulieren. Es werden zunächst folgende Prädikate definiert:

$Va(x)$ „x ist ein Variablenzeichen“,

$Op(x)$ „x ist ein Operationszeichen“,

$Neg(x)$ „x ist ein Negationszeichen“,

$Kla(x)$ „x ist ein Klammerauf-Zeichen“,

$Klz(x)$ „x ist ein Klammerzu-Zeichen“,

$Sa(x)$ „x ist ein sinnvoller Ausdruck“,

$Va'(x)$ „der Ausdruck x besteht aus einer einzelnen Variablen“.

Die Einführung eines Prädikates $Va'(x)$ ist nötig, um ein isoliertes Zeichen von einem durch ein einzelnes Zeichen dargestellten Ausdruck zu unterscheiden.

Eine aus den Zeichenfolgen x und y durch einfaches Aneinandersetzen gebildete Zeichenfolge sei mit $Lz(x,y)$ bezeichnet (Längszusammensetzung).

Nunmehr können wir die obigen Regeln für „Sinnvolle Zeichenfolge“ wie folgt im Prädikatenkalkül ansetzen[1]

$$x\left[Sa(x) \sim \begin{pmatrix} Va'(x) \\ \vee(x = Lz(y_0,y_1)\&Neg(y_0)\&Sa(y_1)) \\ \vee(x = Lz(y_0,y_1,y_2)\&Sa(y_0)\&Op(y_1)\&Sa(y_2)) \\ \vee(x = Lz(y_0,y_1,y_2)\&Kla(y_0)\&Sa(y_1)\&Klz(y_2)) \end{pmatrix}\right] \tag{3}$$

Durch diesen Ausdruck ist da Kriterium „sinnvolle Zeichenfolge“ in impliziter Form gegeben. Die praktische Aufgabe für eine Rechenmaschine besteht jedoch darin, in jedem vorgelegten Fall festzustellen, ob das Kriterium erfüllt ist oder nicht. Hierzu muß obige Formel (3) mit Hilfe des vom Verfasser aufgestellten „Allgemeinen Plankalküls“ in eine explizite Rechenanweisung umgewandelt werden. Diese Umformung kann hier aus Raummangel nur in groben Zügen angedeutet werden.

Die mechanische Ermittlung des Kriteriums Sa(x) erfolgt am besten schrittweise Zeichen für Zeichen, wobei bei jedem Zeichen von der Maschine festzustellen ist, ob die bis dahin gegebene Zeichenfolge sinnvoll ergänzbar ist. Erst bei dem letzten Zeichen kann das Kriterium Sa(x) gebildet werden.

Diese Ermittlung erfolgt mit Hilfe dreier Hilfsprädikate:

$Az(x)$	„x ist ein Zeichen, welches am Anfang eines sinnvollen Ausdrucks stehen kann“,
$Sz(x)$	„x ist ein Zeichen, welches am Schluß eines sinnvollen Ausdrucks stehen kann“,
$Sq(x,y)$	„innerhalb eines sinnvollen Ausdrucks kann das Zeichen y auf das Zeichen x folgen“.

Eine nähere Untersuchung, die hier umgangen wird, ergibt folgende Ansätze für diese Prädikate:

$$\begin{array}{lcl} Az(x) & = & Va(x) \vee Neg(x) \vee Kla(x), \\ Sz(x) & = & Va(x) \vee Klz(x), \\ Sq(x,y) & = & -(Sz(x) \sim Az(y)). \end{array} \tag{4}$$

[1]Siehe Anmerkung am Schluß.

Hiernach lassen sich zunächst folgende Bedingungen für sinnvolle Ausdrücke aufstellen:

1. Das erste Zeichen muß der Bedingung (Az(x) genügen;
2. Zwei aufeinanderfolgende Zeichen müssen der Bedingung Sq(x, y) genügen;
3. Das letzte Zeichen muß der Bedingung Sz(x) genügen.

Hierzu kommen noch weitere Bedingungen, welche die Zahl der Klammern betreffen. Für den gesamten Ausdruck muß zunächst gelten:

4. Die Zahl der Klammerauf-Zeichen muß gleich der Zahl der Klammerzu-Zeichen sein.

Für jeden Abschnitt der schrittweise untersuchten Zeichenfolgen muß ferner gelten:

5. Die Zahl der Klammerauf-Zeichen muß größer oder gleich der Zahl der Klammerzu-Zeichen sein.

Diese fünf Sätze lassen sich wiederum entsprechend Ansatz (3) im Prädikatenkalkül ansetzen. Es soll jedoch gleich die hiernach abgeleitete explizite Rechenanweisung gegeben werden. Diese sieht im Allgemeinen Plankalkül wie folgt aus:

$$
\begin{array}{c|l}
 & 1\; R(\underset{m\sigma}{\overset{0}{V}}\quad) \Rightarrow \underset{0}{\overset{0}{R}} \\
V & \\
S &
\end{array}
$$

$$
\begin{array}{c|cccc|ccc|l}
 & 2\; Az(& V &) \Rightarrow \& & R & 3\; & V \Rightarrow & Z & 4\; 0 \Rightarrow \varepsilon \\
V & & 0 & & 0 & & 0 & 0 & \\
K & & 0 & & & & 0 & & \\
S & & \sigma & & 0 & & \sigma & \sigma & \quad 1.n
\end{array}
$$

$$
\begin{array}{c|l}
 & W \left[\begin{array}{l}
\begin{array}{ccccccc|ccccc}
5\; \mu x & \left[x \in\right. & V & \& x \neq & \left. V \right] & \Rightarrow & Z & 6\; Sq(& Z, & Z \Rightarrow \& & R \\
 & & 0 & & 0 & & 1 & & 0 & 1 & 0 \\
 & & & & 0 & & & & & & \\
 & \sigma & m\sigma & & \sigma & & \sigma & & \sigma & \sigma & 0
\end{array} \\
\begin{array}{l|l}
7\; Kla(\underset{\sigma}{\overset{1}{Z}} \underset{\cdot}{\rightarrow} (\varepsilon + 1 \Rightarrow \varepsilon) & 8\; Klz(\underset{\sigma}{\overset{1}{Z}}\;) \underset{\cdot}{\rightarrow} (\varepsilon - 1 = \varepsilon)
\end{array} \\
\begin{array}{l|l}
9\; \varepsilon \geq 0 \Rightarrow \& \underset{0}{\overset{0}{R}} & 10\; \underset{\sigma}{\overset{1}{Z}} \Rightarrow \underset{\sigma}{\overset{0}{Z}}
\end{array}
\end{array} \right] \\
V & \\
K & \\
S &
\end{array}
\qquad (5)
$$

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

Um dieses spezielle Beispiel (5) 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 F, 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 „Ergibt-Zeichen“ $\Rightarrow$ 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 Resultat erkenntlich sind.

Für das spezielle Beispiel gilt:

1, der Randauszug besagt, daß als Variable eine Angabe V_0 von der Struktur m 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 $\Rightarrow \& R$ bedeutet: „ergibt ein Konkunktionsglied 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 Z1.

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 Klammerauf-Zeichen wird ε um 1 erhöht, bei einem Klammerzu-Zeichen um 1 erniedrigt. Das Zeichen $\dot{\rightarrow}$ 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.

Rechenvorschriften dieser Art bieten die notwendigen mathematischen Voraussetzungen, um die Frage der mechanischen Lösung solcher Aufgaben mit Hilfe „logistischer“ Rechenmaschinen praktisch in Angriff zu nehmen. Es ist leicht einzusehen, daß in der skizzierten Art auch kompliziertere mathematische Operationen explizit als Rechenvorschrift formulierbar sind. Dasselbe gilt für schematische Aufgaben aus Gebieten außerhalb der Mathematik.

Das praktische Arbeiten mit einer solchen Rechenmaschine würde bei Lösung mathematischer Aufgaben dann wie folgt vor sich gehen: Die für ein bestimmtes Gebiet, z.B. Aussagenkalkül, entwickelten Rechenvorschriften, die alle auf diesem Gebiet vorkommenden schematischen Prozesse explizit enthalten müssen, werden in einen Programmspeicher der Maschine hineingegeben. Der Mathematiker gibt die zu untersuchende Aufgabe über eine Tastatur in das Gerät. Es sind eine Reihe von Operationstasten verfügbar, welche die verschiedenen Umformprozesse auslösen. Zwischen- und Endresultate können gedruckt werden. So könnte z.B. das Gerät umständliche Beweisverfahren fehlerfrei nach den Regeln des Kalküls durchführen, wobei der Mathematiker lediglich die grundsätzliche Richtung des Prozesses angibt. Jeder Mathematiker weiß, welche ungeheure rein schematische Kleinarbeit bei schwierigen mathematischen Ableitungen zu leisten ist. Diese gilt es der Mechanisierung zugänglich zu machen.

Die Ausführbarkeit einer solchen Maschine erscheint nach dem heutigen Stand der konstruktiven Entwicklung gesichert.

Die Schaffung der mathematischen Voraussetzung, welche einen wesentlichen Teil der Gesamtaufgabe bildet, könnte bereits jetzt unabhängig von der praktischen Entwicklung in Angriff genommen werden.

Zusammenfassung

Der Verfasser stellte sich zur Aufgabe, einen Rechengerätetyp zu schaffen, mit Hilfe dessen beliebige schematisch-kombinative Aufgaben automatisch zu lösen sind (Logistische Rechenmaschine).

Als wesentliche theoretische Voraussetzung für diese Geräteentwicklung ist die Aufstellung eines Formalismus anzusehen, der es gestattet, beliebige schematisch-kombinative Aufgaben in expliziter Form als Rechenanweisungen zu formulieren, nach welchen entsprechende Rechenmaschinen arbeiten können.

Während die bekannten, für andere Zwecke entwickelten logistischen Kalküle zur Formulierung von derartigen Rechenanweisungen nicht ausreichend sind, stellt der vom Verfasser aufgestellte „Allgemeine Plankalkül“ einen Weg zur Lösung der gestellten Aufgabe dar.

An einem aus dem Gebiete der Mathematik gegriffenen Beispiel einer schematisch-kombinativen Aufgabe wird versucht, den grundsätzlichen Gedankengang bei der Aufstellung einer solchen allgemeinen Rechenanweisung zu erläutern.

(Eingegangen am 6.12.1948)

Zusatzbemerkung bei der Korrektur: Die Formel (3) stellt eine leichter verständliche abgekürzte Form der strengen Formulierung im Prädikatenkalkül dar, welche im folgenden gegeben wird:

$$(x)\left[Sa(x) \sim \left[\begin{array}{l} Va'(x) \\ \vee (Ey_0)(Ey_1)[(x = Lz(y_0, y_1))\&Neg(y_0)\&Sa(y_1)] \\ \vee (Ey_0)(Ey_1)(Ey_2)\left[\begin{array}{l}(x = Lz(y_0, z))\&(z = Lz(y_1, y_2)) \\ \& \left[\begin{array}{l}[Sa(y_0)\&Op(y_1)\&Sa(y_2)] \\ \left[Kla(y_0)\&Sa(y_1)\&Klz(y_2)\right]\end{array}\right]\end{array}\right]\end{array}\right]\right]$$