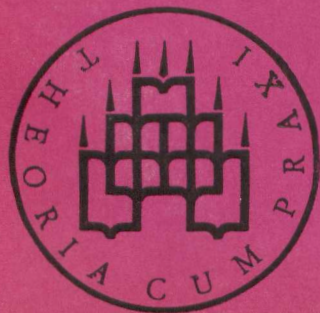


Rostocker

Mathematisches Kolloquium

Heft 4



**WILHELM-PIECK-UNIVERSITÄT
ROSTOCK**

ROSTOCKER MATHEMATISCHES KOLLOQUIUM

Heft 4

1977

Wilhelm-Pieck-Universität Rostock

Sektion Mathematik

**Redaktion: Abt. Wissenschaftspublizistik der Wilhelm-Pieck-
Universität Rostock, 25 Rostock, Vogelsang 13/14
Fernruf 369 577**

**Verantwortlicher Redakteur: Dipl.-Ges.-Wiss. Bruno Schrage
Fachredakteur: Doz. Dr. rer. nat. Gerhard Maeß,
Sektion Mathematik**

**Herausgegeben von der Wilhelm-Pieck-Universität Rostock unter
Genehmigungs-Nr. C 758/77**

Druck: Ostsee-Druck Rostock, Werk II

InhaltSeite

Neßelmann, Dieter	Über irreduzible Ideale in 1-dimensionalen lokalen Ringen	5
Berg, Lothár	Anwendungen innerer und äußerer Inversen	9
Sändig, Anna-Margarete	Korrekt gestellte Aufgaben für lineare elliptische partielle Differentialoperatoren mit konstanten Koeffizienten	19
Keller, Eckart	Über das Randverhalten biharmonischer Funktionen in der Hyperkugel	33
Maeß, Gerhard	Ein stochastischer Projektionsalgorithmus zur iterativen Lösung linearer Gleichungssysteme	47
Thielcke, Helmut	Numerische Ermittlung des Trägheitsellipsoids eines starren Körpers (Massepunktsystem)	55
Fehlauer, Klaus-Uwe	Eine Methode zur Abschätzung des Quadraturfehlers im 2-Dimensionalen	63
Dassow, Jürgen	Über Abschlußeigenschaften von biologisch motivierten Sprachen	69
Goos, Gerhard	Einige Eigenschaften der Programmiersprache BALG	85
Hohberg, Bodo	Programmierung von Dialogcompilern	99
Schier, Dietmar	PL/1-Spracherweiterung durch Benutzung der Möglichkeiten des Makrointerpreters	113
Sietmann, Günter	Zur Auswahl wesentlichen mathematischen Wissens hinsichtlich der inhaltlichen Abstimmung mit den Schulfächern Physik, Chemie und Biologie	117
Drews, Klaus-Dieter	Zur Definition der reellen Zahlen durch Dezimalbrüche	123

Über irreduzible Ideale in 1-dimensionalen lokalen Ringen

(R, M) sei stets ein lokaler Ring der Krull-Dimension 1 mit dem maximalen Ideal M und $o.R = Q \cap A_0$ eine Darstellung des Nullideals, bei der Q die isolierte 1-dimensionale und A_0 eine mögliche M -primäre Komponente ist. R ist ein Cohen-Macaulay Ring genau dann, wenn $o.R = Q$, und nach /3/ ein Buchsbaum-Ring genau dann, wenn $M \cdot Q = o.R$. Ein Element $z \in M$ ist genau dann ein Parameter in R , wenn $Q : z.R = Q$. Ist N ein endlicher R -Modul, bezeichnen wir mit $L(N)$ seine Länge als R -Modul. In /1/ wird gezeigt, daß für ein M -primäres Ideal B die Länge $L(B; M/B)$ die Anzahl der irreduziblen Komponenten von B ist. Nach /2/ ist diese Anzahl für Parameterideale $z.R \subseteq M$ in Cohen-Macaulay Ringen eine Ringinvariante. Wir zeigen hier für beliebige 1-dimensionale lokale Ringe:

Satz 1: Sind a und x Parameter, dann gilt

$$L(ax.R; M/ax.R) = L(x.R; M/x.R) + L((x, Q).R; M/(Q, x.R; M)) + L((a, Q)x.R; M/ax.R) - L((a, Q)x.R; M/ax.R; M).$$

Satz 2: Sind $y = a \cdot x$ und $z = b \cdot y$ Parameter und $Q \cdot x.R = o.R$, dann gilt $L(y.R; M/y.R) = L(z.R; M/z.R)$, wenn $a, b \in M$.

Für Buchsbaum-Ringe ergibt sich mit obigen Voraussetzungen:

Folgerung 1: $L(y.R; M/y.R) = L(ax.R; M/ax.R) = L(x.R; M/x.R) + L((x, Q).R; M/x.R; M) \geq L(x.R; M/x.R)$.

Folgerung 2: Für jeden Parameter $z \in ax.R$ ist $L(z.R; M/z.R) = L(ax.R; M/ax.R)$.

Hieraus erhält man

Satz 3: Sind $y, z \in M$ zwei Parameter in einem 1-dimensionalen lokalen Buchsbaum-Ring, so daß $y = a \cdot b$ und $z = c \cdot d$ mit $a, b, c, d \in M$, dann gilt $L(y.R; M/y.R) = L(z.R; M/z.R)$.

Wir wollen nun obige Aussagen beweisen.

Beweis zu Satz 1: Wir zeigen zunächst: $(x) \cap (ax, Q) = (a, Q)x$.

Offenbar ist $(a, Q)x = (ax, Qx) \subseteq (x) \cap (ax, Q)$. Sei $z = c \cdot ax + d \in (x)$ mit $c \in R$ und $d \in Q$. Dann ist notwendig $d \in (x) \cap Q = (x)Q$, also $d = d' \cdot x$ mit $d' \in Q$ und daher $z = (c \cdot a + d')x \in (a, Q)x$.

Insbesondere haben wir

$$(a, Q)x.R : M = (x.R : M) \cap ((ax, Q).R : M)$$

und die Isomorphismen

$$(ax, Q)/(a, Q)x = (ax, Q)/(x) \cap (ax, Q) \cong (x, Q)/(x)$$

sowie

$$\begin{aligned} (ax, Q):M/(a, Q)x:M &= (ax, Q):M/((x):M) \cap ((ax, Q):M) \cong \\ &\cong ((ax, Q):M, x.R:M)/x.R:M \subseteq (x, Q):M/x.R:M. \end{aligned}$$

Wir zeigen jetzt: $((ax, Q):M, x.R:M) = (Q, x.R:M)$.

Offenbar ist $(Q, x.R:M) \subseteq ((ax, Q):M, x.R:M)$. Sei $z \in (ax, Q):M$, also $z.M \subseteq (ax, Q)$. Dann ist $z \cdot a = r \cdot ax + q$ mit $r \in R$ und $q \in Q$, also $(z - r \cdot x)a \in Q$, was $z - r \cdot x \in Q$ zur Folge hat. Daher ist $z = r \cdot x + q' \in (x, Q) \subseteq (Q, x.R:M)$ und somit

$$((ax, Q):M, x.R:M) \subseteq (Q, x.R:M).$$

Zusammenfassend ergibt sich

$$(ax, Q):M/(a, Q)x:M \cong (Q, x.R:M)/x.R:M \subseteq (x, Q):M/x.R:M.$$

Für die Modulnängen erhält man hieraus

$$\begin{aligned} L(ax.R:M/ax.R) &= L((ax, Q):M/(ax, Q)) - L((ax, Q):M/(a, Q)x:M) \\ &\quad - L((a, Q)x:M/ax.R:M) + L((ax, Q)/(a, Q)x) + \\ &\quad + L((a, Q)x.R/ax.R). \end{aligned}$$

Da R/Q ein Cohen-Macaulay Ring ist, ergibt sich nach /2/,

Theorem 3, $L((ax, Q):M/(ax, Q)) = L((x, Q):M/(x, Q))$, und daher

$$\begin{aligned} L(ax.R:M/ax.R) &= L((x, Q):M/(x, Q)) - L((Q, x.R:M)/x.R:M) + \\ &\quad + L((x, Q)/x.R) + L((a, Q)x.R/ax.R) - \\ &\quad - L((a, Q)x.R:M/ax.R:M). \end{aligned}$$

Eine weitere Zusammenfassung liefert die Aussage, qed.

Lemma: x und y seien zwei Parameter. Dann gilt

$$(x, y, Q):M = (x((y, Q):M), Q) = (y((x, Q):M), Q).$$

Beweis: Wir brauchen nur eine Gleichheit zu zeigen. Offenbar ist $(x((y, Q):M), Q) \subseteq (x, y, Q):M$. Sei $z.M \subseteq (x, y, Q)$. Dann ist insbesondere $z \cdot y - r \cdot x \cdot y \in Q$ mit $r \in R$ und daher $z - r \cdot x \in Q$, etwa $z = r \cdot x + q$. Ist $m \in M$ beliebig, so gilt $z \cdot m = r \cdot x \cdot m + q \cdot m$

$\in (x.y, Q)$, etwa $z.m = s.x.y + q'$ und somit $r.x.m - s.x.y \in Q$ sowie $r.m - s.y \in Q$. Daher ist $r.M \subseteq (y, Q)$, also $r \in (y, Q):M$, was $z = r.x + q \in (x((y, Q):M), Q)$ zur Folge hat, qed.

Beweis zu Satz 2: Aus Satz 1 ergibt sich

$$L(z.R:M/z.R) = L(y.R:M/y.R) + L((y, Q):M/(Q, y.R:M)).$$

Nach dem Lemma gilt

$(y, Q):M = (x((a, Q):M), Q) \subseteq (x(a, Q):M, Q) = (ax.R:M), Q)$, also $L((y, Q):M/(Q, y.R:M)) = 0$, qed.

Folgerung 1 ergibt sich unmittelbar aus Satz 1, da wegen $M.Q = 0.R$ auch $x.Q = 0.R$ ist. Hieraus ergibt sich gleichzeitig Folgerung 2. Satz 3 erhält man aus Folgerung 2 unter Beachtung von

$$L(y.R:M/y.R) = L(y.z.R:M/y.z.R) = L(z.R:M/z.R).$$

Literatur:

- /1/ Gröbner, W. Über irreduzible Ideale in kommutativen Ringen; Math. Ann 110 197 - 222 (1935)
- /2/ Northcott, D.G. On irreducible ideals in local rings; J. London Math. Soc. 32 82 - 88 (1957)
- /3/ Stückrad, J. und Vogel, W.
Eine Verallgemeinerung der Cohen-Macaulay Ringe und eine Anwendung auf ein Problem der Multiplizitätstheorie
J. Math. Kyoto Univ. 13 513 - 528 (1973)

eingegangen am: 15. 1. 1977

Anschrift des Verfassers:

Dr.rer.nat. Dieter Neßelmann
Wilhelm-Pieck-Universität Rostock
Sektion Mathematik
DDR 25 Rostock
Universitätsplatz 1

Anwendungen innerer und äußerer Inversen

Es seien V, V' zwei Mengen, A ein Operator von V in V' und R ein Operator von V' in V . Gilt eine der Gleichungen

$$A = ARA, R = RAR, \quad (1)$$

so heißt R nach /8/ im ersten Fall eine innere und im zweiten Fall eine äußere Inverse von A . Solche verallgemeinerten Inversen kann man benutzen, um Gleichungen

$$Ax = f \quad (2)$$

bei gegebenem $f \in V'$ nach $x \in V$ aufzulösen. Hierüber gibt es eine Reihe von Ergebnissen, von denen einige jetzt verallgemeinert und in möglichst einheitlicher Form zusammengestellt werden sollen.

Der Einfachheit wegen nehmen wir an, daß V, V' lineare Vektorräume mit demselben Koeffizientenring und A, R lineare Operatoren sind. Aus A, R bilden wir die Operatoren

$$P = I - RA, Q = I - AR, \quad (3)$$

wobei I sowohl den Einheitsoperator über V als auch den Einheitsoperator über V' bezeichnet, beide Operatoren sind durch die Beziehungen

$$AP = QA, PR = RQ \quad (4)$$

miteinander verknüpft. Ist eine der Beziehungen (1) erfüllt, so verschwinden die entsprechenden Produkte in (4). Außerdem sind dann P, Q Projektoren, d.h., es gilt

$$P^2 = P, Q^2 = Q. \quad (5)$$

Ist umgekehrt P Projektor, so ist auch RA Projektor, d.h., es gilt $RA = RARA$. Hieraus folgt bei rechtsinvertierbarem A die zweite und bei linksinvertierbarem R die erste der Beziehungen (1). Analog folgt, falls Q Projektor ist, aus $AR = ARAR$ bei linksinvertierbarem A die zweite und bei rechtsinvertierbarem

R die erste der Beziehungen (1).

Satz 1: Die Gleichung (2) ist zu dem System der Gleichungen

$$x = Px + Rf, APx = Qf \quad (6)$$

äquivalent.

Beweis: Aus (2) folgt durch Multiplikation mit R die erste und durch Multiplikation mit Q unter Beachtung von (4) die zweite der Gleichungen (6). Umgekehrt folgt aus (6) wegen (3)

$$Ax = APx + ARf = (Q + AR)f = f.$$

Dieser Satz ist eine Verallgemeinerung von Satz 3 aus /1/, wenn man davon absieht, daß letzterer sogar für gewisse nicht-lineare Operatoren gilt. In /6/ wird er bei der Auflösung linearer Gleichungssysteme angewendet. Dabei ist es nicht erforderlich, daß R innere oder äußere Inverse von A ist. Die Nützlichkeit von Satz 1 erkennt man aus folgendem

Zusatz: In (6) genügt es, die zweite Gleichung nach Px aufzulösen und das Ergebnis in die erste Gleichung einzusetzen, um die allgemeine Lösung von (2) zu erhalten. Ist $APx = Qf$ nicht lösbar, so ist auch (2) nicht lösbar. Verschwinden AP und Qf, so kann Px (im Bildbereich von P) beliebig gewählt werden.

Beweis: Es sei P_y Lösung von $AP_y = Qf$ und $x = P_y + Rf$.

Dann gilt wie beim vorhergehenden Beweis $Ax = f$ sowie

$$Px = (I - RA)P_y + PRf = P_y + (PR - RQ)f = P_y.$$

Die letzten beiden Aussagen sind nach Satz 1 evident.

Es sei jetzt B ein weiterer im allgemeinen nichtlinearer Operator von V in V', den wir in die Gleichung (2) als Störungsoperator einbauen.

Satz 2: Ist R innere Inverse von A, so ist die Gleichung

$$(A + B)x = f \quad (7)$$

zu dem System

$$x = Px - RBx + Rf, \quad QBx = Qf \quad (8)$$

Äquivalent.

Beweis: Die Richtigkeit dieses Satzes folgt sofort aus Satz 1, wenn man dort f durch $f - Bx$ ersetzt und $AP = 0$ beachtet.

Im Unterschied zum vorhergehenden Fall läßt sich das System (8) im allgemeinen nicht entkoppeln. Doch haben wir auch diesmal wieder einen praktischen Vorteil, denn es gilt der folgende

Zusatz: Ermittelt man Py und Bz aus dem System

$$Bz = B(Py - RBz + Rf), \quad QBz = Qf,$$

so ist $x = Py - RBz + Rf$ die allgemeine Lösung von (7).

Beweis: Man verifiziert leicht die Gleichungen $Px = Py$, $Bx = Bz$ und findet dann

$$Ax = AR(f - Bz) = f - Bz = f - Bx.$$

Die Vorteile dieses Lösungsverfahrens zur Auflösung linearer Gleichungssysteme wurden (in leichter Abänderung) in /2/ durchgeführt, dabei wurde A dort speziell als zirkulante Matrix gewählt. Auch die Arbeit /4/ zur Auflösung entarteter Gleichungen kann als praktische Illustration zu diesem Verfahren angesehen werden. Allgemein wird man es verwenden, wenn zwar für den in (7) auftretenden Operator $A + B$ keine innere Inverse bekannt ist, wohl aber für einen möglichst guten Näherungsoperator A . Die Operatoren $A + B$ und A brauchen dabei keineswegs dieselben Auflösungsseigenschaften zu besitzen.

Bei der Auflösung komplizierterer Systeme als (6) bzw. (8) ist es notwendig, aus gegebenen Projektoren neue zu bilden.

Hilfssatz: Sind P_1, P_2 Projektoren und ist P_0 ein weiterer linearer Operator über V , so ist $P = P_1 + P_2 - P_0$ genau dann Projektor, wenn

$$P_0^2 = P_0 + (P_1 - P_0)(P_2 - P_0) + (P_2 - P_0)(P_1 - P_0) \quad (9)$$

gilt. Im Fall $P_0 = P_1 P_2$ lautet diese Bedingung

$$(I - P_1)P_2P_1(I - P_2) = 0. \quad (10)$$

Ist außerdem P_1P_2 Projektor, so ist auch P_2P_1 Projektor.

Beweis: Aus

$$P^2 = (P_1 - P_0)P_1 + P_2P_1 + (P_1 - P_0)(P_2 - P_0) + P_2(P_2 - P_0)$$

und

$$- P_0P_1 + P_2P_1 + P_2P_0 = (P_2 - P_0)(P_1 - P_0) - P_0^2$$

ist unmittelbar ersichtlich, daß (9) zu $P^2 = P$ äquivalent ist.

Im Fall $P_0 = P_1P_2$ folgt aus

$$(P_1 - P_0)(P_2 - P_0) = P_1(I - P_2)(I - P_1)P_2$$

$$= -P_1P_2(I - P_1P_2) = P_0^2 - P_0$$

und

$$(P_2 - P_0)(P_1 - P_0) = (I - P_1)P_2P_1(I - P_2)$$

die zweite Behauptung.

Durch Auflösung der Klammern erhalten wir aus (10), falls P_1P_2 Projektor ist,

$$P_1P_2 + P_2P_1 = P_1P_2P_1 + P_2P_1P_2,$$

und hieraus ergibt sich durch Multiplikation von links mit P_2 , daß dann auch P_2P_1 Projektor ist.

Zwei Spezialfälle dieses Hilfssatzes wurden in /3/ bewiesen und dort zur Herleitung asymptotischer Aussagen verwendet (vgl. auch /7/).

Im folgenden ersetzen wir das Operatorenpaar A, R durch n Paare $A_i, R_i, i = 1, \dots, n$, und führen analog zu (3) die Operatoren $P_i = I - R_iA_i, Q_i = I - A_iR_i$ ein. Weiterhin bilden wir die Operatormatrizen

$$A = (A_1 \ A_2 \ \dots \ A_n)^T, \quad R = (R_1 \ P_1R_2 \ \dots \ P_1 \dots P_{n-1}R_n). \quad (11)$$

Satz 3: Zu den Operatoren (11) gehört nach (3) der Operator

$$P = P_1P_2 \ \dots \ P_n. \quad (12)$$

Der Operator R ist genau dann innere Inverse von A, falls

$$A_i P = 0 \quad (13)$$

ist und genau dann äußere Inverse von A, falls

$$P R_i = 0 \quad (14)$$

ist für $i = 1, \dots, n$.

Beweis: Durch Multiplikation der Operatoren (11) finden wir

$$RA = R_1 A_1 + P_1 R_2 A_2 + \dots + P_1 \dots P_{n-1} R_n A_n.$$

Substituieren wir hier $R_i A_i = I - P_i$, so heben sich in der Summe alle inneren Glieder heraus, und es bleibt

$$RA = I - P_1 \dots P_n$$

übrig, d.h. (12). Der Operator R ist genau dann innere Inverse von A, falls $AP = 0$ gilt, d.h. (13), und genau dann äußere Inverse von A, falls $PR = 0$ gilt, d.h.

$$P P_1 \dots P_{i-1} R_i = 0$$

für $i = 1, \dots, n$. Für $i = 1$ ist diese Bedingung mit (14) identisch. Für $i = 2$ besagt sie wegen

$$P P_1 R_2 = P(I - R_1 A_1) R_2 = P R_2 = 0$$

ebenfalls nichts anderes als (14). Entsprechend läßt sich die Äquivalenz dieser Bedingungen auch für die weiteren Werte von i rekursiv nachweisen.

Die Matrixen AR und Q (vgl. (3)) lassen sich ebenfalls geschlossen angeben. Um die Ergebnisse zu vereinfachen, wollen wir aber die Voraussetzungen verschärfen.

Zusatz: Sind die R_j innere Inversen der Operatoren A_j mit

$$A_i R_j = 0 \quad (15)$$

für $1 \leq j < i \leq n$, so ist (13) erfüllt, und es gilt

$$AR = \text{diag} (A_1 R_1 \ A_2 R_2 \ \dots \ A_n R_n),$$

$$Q = \text{diag} (Q_1 \ Q_2 \ \dots \ Q_n).$$

Beweis: Aus

$$A_i P_1 \dots P_{i-1} = A_i (I - R_1 A_1) \dots (I - R_{i-1} A_{i-1}) = A_i$$

und $A_i P_i = 0$ folgt

$$A_i P_1 \dots P_i = 0 \quad (16)$$

und damit (13). Wegen (11) hat die Matrix AR an der Stelle (i,j) das Element

$$A_i P_1 \dots P_{j-1} R_j.$$

Für $i < j$ ist es wegen (16) gleich 0. Für $i \geq j$ ist es gleich $A_i R_j$, so daß die Behauptungen wegen (15) bewiesen sind.

Ein Spezialfall von Satz 3 wurde in /5/ bewiesen. Setzen wir nur (16) voraus, so ist ebenfalls (13) erfüllt, und die Matrizen AR und Q sind untere Dreiecksmatrizen. Für $i = 1$ besagt (16), daß wenigstens R_1 innere Inverse von A_1 ist. Für $i = 2$ lautet (16) $A_2 P_1 P_2 = 0$, und diese Bedingung ist für (10) hinreichend.

Eine Anwendung von Satz 3 ergibt sich, wenn dieser Satz bei erfüllter Voraussetzung (13) mit Satz 1 gekoppelt wird. Dann findet man für das System simultaner Gleichungen

$$A_1 x = f_1, \dots, A_n x = f_n \quad (17)$$

unter der notwendigen und hinreichenden Lösbarkeitsbedingung $Q(f_1 \dots f_n)^T = 0$ die allgemeine Lösung

$$x = Py + R_1 f_1 + P_1 R_2 f_2 + \dots + P_1 \dots P_{n-1} R_n f_n, \quad (18)$$

wobei y in V beliebig vorgegeben werden kann. Zwei Beispiele hierzu wurden in /5/ behandelt.

Gleichungssysteme: Ein weiteres Beispiel erhalten wir, wenn wir x in (17) als gewöhnlichen m -dimensionalen Spaltenvektor und die A_i als gewöhnliche m -dimensionale Zeilenvektoren

$$A_i = (a_{i1} \dots a_{im})$$

wählen. Nehmen wir der Einfachheit wegen an, daß die Matrix

$A = (a_{ij})$ regulär und damit $m = n$ ist, so können wir die R_j als zu den A_j rechtsinverse Spaltenvektoren wählen, die außerdem den Bedingungen (15) genügen. Die Gleichungssysteme

$$A_i R_j = \delta_{ij},$$

$i = j, \dots, n$, wobei δ_{ij} das Kroneckersymbol bezeichnet, sind dann nämlich bei festem j stets nach R_j auflösbar. Mit diesen R_j liefert (18) die Lösung von (17), wobei wegen der Regularität von A natürlich $P = 0$ ist. Für kleine n kann die Lösungsdarstellung (18) auch zur numerischen Auflösung des Gleichungssystems (17) verwendet werden, für große n sind aber andere numerische Verfahren vorzuziehen.

Faktorisierungen: Nach Einführung eines weiteren Vektorraumes V'' möge A die Faktorisierungen

$$A = A_i B_i$$

für $i = 1, \dots, n$ besitzen, wobei die B_i lineare Operatoren von V in V'' und die A_i lineare Operatoren von V'' in V' seien. Bilden wir jetzt aus einer beliebigen Lösung x von (2) die Elemente

$$B_i x = y_i, \quad (19)$$

so erfüllen sie die Gleichungen

$$A_i y_i = f \quad (20)$$

und besitzen mit den früheren Bezeichnungen nach Satz 1 die Darstellungen

$$y_i = P_i y_i + R_i f. \quad (21)$$

Im folgenden sollen n lineare Operatoren C_i von V'' in V mit

$$I = \sum_{i=1}^n C_i B_i$$

existieren. Dann besitzt jede Lösung von (2) wegen (19) die Darstellung

$$x = \sum_{i=1}^n C_i y_i, \quad (22)$$

und hieraus folgt durch Einsetzen von (21) wie in (6)

$x = Px + Rf$ mit

$$P = \sum_{i=1}^n C_i P_i B_i, \quad R = \sum_{i=1}^n C_i R_i,$$

wobei sich der Zusammenhang (3) zwischen diesen Operatoren unmittelbar nachprüfen läßt.

Abschließend wollen wir uns der umgekehrten Fragestellung zuwenden, unter welchen Bedingungen das Element (22) mit gewissen oder sogar beliebigen Lösungen (21) von (20) eine Lösung der Gleichung (2) bzw. der simultanen Gleichungen (19) ist. Die letzte Aussage ist eine Verschärfung der vorhergehenden, da aus (19) und (20) stets (2), aber aus (2) und (20) zunächst nur $A_i(B_i x - y_i) = 0$ folgt.

Satz 4: Ist das System simultaner Gleichungen (19) lösbar, so ist die Lösung eindeutig bestimmt und wird durch (22) gegeben. Ist

$$AC_i P_i = 0 \quad (23)$$

für alle i , so ist R innere Inverse von A und (22) ist bei erfüllter Lösbarkeitsbedingung $Qf = 0$ eine Lösung von (2). Sind alle R_i innere Inversen der zugehörigen Operatoren A_i , so ist für (23) die Bedingung

$$B_i C_j P_j = \delta_{ij} P_j \quad (24)$$

für alle i, j hinreichend, und (22) ist bei erfüllter Lösbarkeitsbedingung $Qf = 0$ auch eine Lösung von (19).

Beweis: Es sei $B_i z = y_i$ für alle i . Dann folgt aus (22)

$$x = \sum_{i=1}^n C_i B_i z = z.$$

Aus (23) erhalten wir

$$AP = \sum_{i=1}^n AC_i P_i B_i = 0,$$

so daß R innere Inverse von A ist. Weiterhin folgt aus (21), (22) und (23)

Über das Randverhalten biharmonischer Funktionen in der
Hyperkugel

1. f sei eine stetige Funktion auf der Kugeloberfläche ∂B der Kugel B mit dem Radius r und

$$u(x) := \text{PI}(f, \partial B)(x)$$

das Poissonsche Integral, so ist

$$\lim_{\substack{x \rightarrow z \\ z \in \partial B}} u(x) = f(z) .$$

Wird in der Poissonschen Integralformel der Ausdruck

$$f(y) dO_n(y) \quad (dO_n \text{ ist das Oberflächenmaß auf } \partial B)$$

durch ein beliebiges signiertes Maß μ von beschränkter Variation auf den Borelschen Teilmengen von ∂B ersetzt, so erhalten wir aus dem Satz von Fatou eine Aussage über das Konvergenzverhalten von $\text{PI}(\mu, \partial B)(x)$, wenn x gegen den Rand der Kugel B strebt. Bei Annäherung von x innerhalb eines bestimmten Kegels mit der Spitze in $z \in \partial B$ an den Punkt z existiert bis auf eine Menge vom Oberflächenmaß Null fast überall ein Grenzwert, der gleich der Ableitung des Maßes μ in diesem Punkt ist. Vgl. /3/ S. 68.

Für biharmonische Funktionen in der Kugel existiert eine dem Poissonschen Integral entsprechende Integraldarstellung

$$b(x) := P(g_0, g_1, \partial B)(x) .$$

Erfüllen die Randfunktionen g_0 und g_1 gewisse Stetigkeits- und Differenzierbarkeitsbedingungen, so konvergiert $b(x)$ gegen die vorgegebenen Randwerte. Vgl. /2/ S. 160

$g_0(y) dO_n(y)$ und $g_1(y) dO_n(y)$ können ebenfalls durch beliebige signierte Maße μ und η ersetzt werden. In der vorliegenden Arbeit wird das Randverhalten von $b(x)$ und $\frac{\partial}{\partial |x|^2} b(x)$ mit beliebigen signierten Maßen von beschränkter Variation unter-

sucht. Wenn x gegen $z \in \partial B$ innerhalb eines Kegels mit der Spitze z konvergiert, strebt $b(x)$ bis auf eine Menge vom Oberflächenmaß Null gegen die Ableitung von μ im Punkt z . Unter bestimmten Voraussetzungen an das Maß μ kann auch die Konvergenz von $\frac{\partial}{\partial |x|} b(x)$ gegen die Ableitung des Maßes η im Punkt z erreicht werden.

2. Punkte des n -dimensionalen Euklidischen Raumes R^n ($n \geq 2$) werden mit kleinen Buchstaben ohne Index bezeichnet $x = (x_1, \dots, x_n)$. Der Abstand zwischen zwei Punkten x und y ist wie folgt definiert

$$|x - y| := \left(\sum_{i=1}^n (x_i - y_i)^2 \right)^{1/2},$$

$$(x, y) := \sum_{i=1}^n x_i y_i.$$

$\mathcal{C}(X)$ sei das System der Borelschen Teilmengen einer kompakten Menge $X \subset R^n$.

Ein signiertes Maß μ ist eine σ -additive Mengenfunktion auf $\mathcal{C}(X)$ mit $\mu(\emptyset) = 0$, die auf kompakten Mengen endlich ist. μ kann in zwei positive Maße μ^+ und μ^- zerlegt werden.

$$\mu = \mu^+ - \mu^-.$$

Die totale Variation von μ ist $|\mu| := \mu^+ + \mu^-$ und die Norm

$$\|\mu\| := \sup_{F \in \mathcal{C}(X)} |\mu|(F).$$

Ist $\|\mu\| < +\infty$ so sagen wir, daß das signierte Maß μ eine beschränkte Variation hat.

Gilt eine Aussage außerhalb einer Menge vom μ -Maß Null, so sagen wir, daß sie für μ -fast alle Punkte (μ -f.a.) gilt, oder sie gilt fast überall bezüglich μ (μ -f.ü.).

B bezeichne eine Hyperkugel im R^n mit dem Radius r und ∂B ihre Oberfläche. Ohne Beschränkung der Allgemeinheit sei der Mittel-

punkt von B mit dem Nullpunkt O des Koordinatensystems identisch.

$$B := \{x : x \in \mathbb{R}^n, |x| < r, r > 0\}$$

$$\partial B := \{x : x \in \mathbb{R}^n, |x| = r, r > 0\}$$

dO_n sei das Oberflächenmaß auf ∂B und ω_n die Oberfläche der n-dimensionalen Einheitskugel. f sei eine integrierbare Funktion auf ∂B , dann ist

$$PI(f, \partial B)(x) := \frac{1}{\omega_n r} \int_{\partial B} \frac{r^2 - p^2}{|x - y|^n} f(y) dO_n(y), \quad p = |x| < r,$$

das Poissonsche Integral. μ sei ein signiertes Maß auf $\mathcal{L}(\partial B)$, dann bezeichne

$$PI(\mu, \partial B)(x) := \frac{1}{\omega_n r} \int_{\partial B} \frac{r^2 - p^2}{|x - y|^n} d\mu(y), \quad p = |x| < r.$$

Durch Berechnung des Ausdruckes $\frac{\partial}{\partial r^2} \frac{r^{n-2}}{|x - y|^n}$ mit Hilfe des

Kosinussatzes können wir aus der Formel von Edenhofer (vgl. /2/ S. 150 (3)) eine Lösungsformel des Dirichletschen Problems für die Bipotentialgleichung in einer n-dimensionalen Kugel gewinnen.

$$\begin{aligned} P(\xi_0, \xi_1, \partial B)(x) &:= \frac{(r^2 - p^2)^2}{4 \omega_n r^2} \int_{\partial B} \frac{n(r^2 - p^2) - (n-4)|x-y|^2}{|x-y|^{n+2}} \xi_0(y) dO_n(y) \\ &\quad - \frac{(r^2 - p^2)^2}{\omega_n r} \int_{\partial B} \frac{1}{|x-y|^n} \xi_1(y) dO_n(y), \end{aligned} \quad (1)$$

z sei ein fester Punkt auf ∂B . Die Randfunktionen ξ_0 und ξ_1 seien in einer Umgebung von z stetig. In Polarkoordinaten habe ξ_0 die Darstellung $\xi_0 = \xi_0(\varphi_1, \dots, \varphi_{n-1})$. ξ_0 sei nach $\varphi_1, \dots, \varphi_{n-1}$ einmal stetig differenzierbar, dann ist nach

Edenhofer (vgl. /2/ S. 160 Satz 2)

$$\lim_{\substack{x \rightarrow z \\ |x| = p < r}} \frac{\partial^i}{\partial (p^2)^i} P(\xi_0, \xi_1, \partial B)(x) = \xi_i(z) \quad i = 0; 1$$

wobei die Konvergenz von x gegen z auf einer beliebigen Kurve innerhalb B erfolgen kann.

$$\text{Sei } Pr(\mu, \partial B)(x) := \frac{n(r^2 - p^2)^3}{4\omega_n r^5} \int_{\partial B} \frac{1}{|x-y|^{n+2}} d\mu(y),$$

so erhalten wir aus (1)

$$P(\mu, \eta, \partial B)(x) := Pr(\mu, \partial B)(x) - \frac{(n-4)(r^2 - p^2)}{4r^2} PI(\mu, \partial B)(x) - (r^2 - p^2) PI(\eta, \partial B)(x), \quad (2)$$

wobei μ und η signierte Maße von beschränkter Variation auf $\mathcal{C}(\partial B)$ sind.

Bezeichnen wir mit

$$Ps(\mu, \partial B)(x) := \int_{\partial B} \frac{(r^2 - p^2)^2}{|x-y|^{n+2}} d\mu(y) \quad \text{und}$$

$$Pt(\mu, \partial B)(x) := \int_{\partial B} \frac{(r^2 - p^2)^4}{|x-y|^{n+4}} d\mu(y), \quad \text{so ergibt sich}$$

$$\begin{aligned} \frac{\partial}{\partial p^2} P(\mu, \eta, \partial B)(x) &:= -\frac{n+2}{4p^2} Pr(\mu, \partial B)(x) - \frac{3n}{4\omega_n r^3} Ps(\mu, \partial B)(x) \\ &\quad + \frac{(n+2)n}{16\omega_n r^3 p^2} Pt(\mu, \partial B)(x) \\ &\quad + \frac{n-4}{4r^2} \left(\left(2 + \frac{n(r^2 - p^2)}{4p^2} \right) PI(\mu, \partial B)(x) - \frac{r^2}{p^2} Pr(\mu, \partial B)(x) \right) \end{aligned}$$

$$+ \left(2 + \frac{n(r^2 - p^2)}{4p^2} \right) \text{PI}(\eta, \partial B)(x) - \frac{r^2}{p^2} \text{Pr}(\eta, \partial B)(x). \quad (3)$$

Aus Gründen der Übersichtlichkeit wurde die Ableitung nach p^2 benutzt. Das bedeutet keine Einschränkung, da sich die Ableitungen $\frac{\partial}{\partial p}$ eindeutig aus den Ableitungen $\frac{\partial}{\partial p^2}$ nach der Formel

$$\frac{\partial}{\partial p^2} u(p) = \frac{1}{2p} \frac{\partial}{\partial p} u(p)$$

berechnen lassen und umgekehrt.

Um das Randverhalten der Ausdrücke (2) und (3) zu untersuchen, müssen zunächst die Ableitungen der signierten Maße μ und ν definiert werden.

$C_{z, \alpha}$ bezeichne einen Kegel in B mit der Spitze im Nullpunkt O , dessen Achse mit der Geraden durch O und $z \in \partial B$ zusammenfällt und dessen halber Öffnungswinkel α ist ($0 \leq \alpha \leq \frac{\pi}{2}$).

$$C_{z, \alpha} := \left\{ x : \cos \alpha \leq \frac{(x, z)}{|x||z|} \leq 1 \right\}.$$

Der Kegel $C_{z, \alpha}$ schneidet aus der Kugeloberfläche die abgeschlossene Polkappe $K_{z, \alpha}^m$ mit dem Mittelpunkt $z \in \partial B$ heraus.

$$K_{z, \alpha}^m := C_{z, \alpha} \cap \partial B$$

\mathcal{A}_z^m bezeichne die Mengen aller Polkappen $K_{z, \alpha}^m$ mit dem Mittelpunkt z . \mathcal{A}_z bezeichne die Menge aller Polkappen $K_{z, \alpha}$, die den Punkt z enthalten, z muß aber nicht Mittelpunkt dieser Polkappen sein. Existiert

$$\lim_{\substack{K \in \mathcal{A}_z \\ O_n(K) \rightarrow 0}} \frac{\mu(K)}{O_n(K)} \quad \text{bzw.} \quad \lim_{\substack{K \in \mathcal{A}_z^m \\ O_n(K) \rightarrow 0}} \frac{\mu(K)}{O_n(K)},$$

so heißt dieser Limes die Ableitung bzw. symmetrische Ableitung des signierten Maßes μ im Punkt z und wird mit $D\mu(z)$ bzw.

$D_g \mu(z)$ bezeichnet. Wenn $D\mu(z)$ existiert, existiert auch $D_g \mu(z)$, und es ist $D\mu(z) = D_g \mu(z)$. Als zweite symmetrische Ableitung bezeichnen wir den Grenzwert, falls er existiert, wie folgt

$$D_g^2 \mu(z) = \lim_{\substack{K \in \mathcal{K}_z^m \\ O_n(K) \rightarrow 0}} \frac{\mu(K)}{\int_K |z-y| dO_n(y)} .$$

In dieser Arbeit sollen nur nicht-tangentiale Limiten der Ausdrücke (2) und (3) untersucht werden. Aussagen über tangentielle Grenzwerte des Poissonschen Integrals an einen Randpunkt eines Kreises werden in /1/ und /4/ gemacht. $u(x)$ sei eine in B definierte Funktion. Der radiale Limes von $u(x)$ im Punkt $z \in \partial B$ ist der Grenzwert

$$R\text{-}\lim_{x \rightarrow z} u(x) := \lim_{\lambda \rightarrow 1^-} u(\lambda z) .$$

$E_z(\Theta)$ sei ein Kegel in B mit der Spitze im Punkt $z \in \partial B$ und dem halben Öffnungswinkel Θ ($0 \leq \Theta < \frac{\pi}{2}$), dessen Achse durch 0 und z verläuft.

$$E_z(\Theta) := \left\{ x : 1 \geq \frac{(-z, x-z)}{|z| |x-z|} \geq \cos \Theta \right\} .$$

Der nicht-tangentiale Limes, falls er existiert, der Funktion $u(x)$ im Punkt $z \in \partial B$ ist dann

$$\Theta\text{-}\lim_{x \rightarrow z} u(x) := \lim_{\substack{x \rightarrow z \\ x \in E_z(\Theta) \cap B}} u(x) .$$

3. Im Hilfssatz 1 werden wir die radialen Limiten der Ausdrücke (2) und (3) untersuchen.

Hilfssatz 1 μ und η seien signierte Maße von beschränkter Variation auf $\mathcal{K}(\partial B)$. Die symmetrischen Ableitungen von μ und η im Punkt $z \in \partial B$ $D_g \mu(z)$ und $D_g \eta(z)$ seien gleich Null.

Aus diesen Voraussetzungen folgt:

$$1. \lim_{x \rightarrow z} P(\mu, \eta, \partial B)(x) = 0$$

2. Gilt zusätzlich noch $D_S^2 \mu(z) = 0$, dann ist auch

$$\lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta, \partial B)(x) = 0.$$

Beweis: Wir beginnen mit dem Beweis des zweiten Teiles des Hilfssatzes. Am Beispiel des Integrals $Pt(\mu, \partial B)(x)$ werden wir zeigen, daß der radiale Limes sämtlicher Integrale aus (3) unter den obigen Voraussetzungen gleich Null ist.

Gegeben sei ein beliebiges hinreichend kleines $\varepsilon > 0$.

Da $D_S^2 \mu(z) = 0$ ist, existiert ein Winkel ϕ mit $0 < \phi < \frac{\pi}{2}$,

so daß für alle Polkappen $K_{z, \gamma}^m$

$$|\mu(K_{z, \gamma}^m)| < \varepsilon \int_{K_{z, \gamma}^m} |z-y| dO_n(y) \quad (4)$$

gilt, wenn $0 \leq \gamma \leq \phi$ ist. Es ist

$$\lim_{x \rightarrow z} Pt(\mu, \partial B / K_{z, \phi}^m)(x) = 0,$$

d. h., es muß noch

$$\lim_{x \rightarrow z} Pt(\mu, K_{z, \phi}^m)(x) = 0 \text{ gezeigt werden.}$$

Es sei

$$F(\gamma) := \int_{K_{z, \gamma}^m} d\mu(y).$$

Unter Beachtung von $|x-y|^2 = r^2 + p^2 - 2pr \cos \gamma$, wobei γ der Winkel $\sphericalangle xOy$ ist, ergibt sich

$$I := Pt(\mu, K_{z, \phi}^m)(x) = \int_0^\phi \frac{(r^2 - p^2)^4}{(r^2 + p^2 - 2pr \cos \gamma)^{(n+4)/2}} dF(\gamma).$$

Durch partielle Integration erhalten wir

$$I = \left[\frac{(r^2 - p^2)^4 F(\varphi)}{(r^2 + p^2 - 2pr \cos \varphi)^{(n+4)/2}} \right]_0^{\varphi} + \\ + \frac{n+4}{2} \int_0^{\varphi} \frac{(r^2 - p^2)^4 2pr \sin \varphi F(\varphi)}{(r^2 + p^2 - 2pr \cos \varphi)^{(n+6)/2}} d\varphi, \quad (5)$$

$$I = I_{\varphi} - I_0 + I_1.$$

I_{φ} bezeichnet den an der Stelle φ ausgerechneten Ausdruck in den eckigen Klammern. Eine analoge Bedeutung hat I_0 . Das Integral in (5) wird mit I_1 bezeichnet. Strebt x radial gegen z , so strebt der Zähler von I_{φ} gegen Null, und der Nenner bleibt größer als Null, d. h., der radiale Limes von I_{φ} ist gleich Null. Da $D_{\mathfrak{B}} \mu(z) = 0$ ist, ist $\mu(\{z\}) = F(0) = 0$, und damit ist auch der radiale Limes von I_0 gleich Null. Es bezeichne

$$G(\varphi) := \int_{K_{z,\varphi}^m} dO_n(y),$$

$$\int_{K_{z,\varphi}^m} |z-y| dO_n(y) = \int_0^{\varphi} 2r \sin \frac{\varphi}{2} dG(\varphi) = H(\varphi). \quad (6)$$

Auf Grund von (4) ist für $0 \leq \varphi \leq \varphi$

$$|F(\varphi)| \leq \varepsilon H(\varphi). \quad (7)$$

Das Integral I_1 kann dann mit (7) wie folgt abgeschätzt werden

$$|I_1| \leq \varepsilon \int_0^{\varphi} \frac{n+4}{2} \cdot \frac{(r^2 - p^2)^4 2pr \sin \varphi H(\varphi)}{(r^2 + p^2 - 2pr \cos \varphi)^{(n+6)/2}} d\varphi$$

Durch partielle Integration erhalten wir

$$|I_1| \leq \varepsilon \left\{ \left[- \frac{(r^2 - p^2)^4 H(\varphi)}{(r^2 + p^2 - 2pr \cos \varphi)^{(n+4)/2}} \right]_0^{\varphi} + \right.$$

$$+ \int_0^{\phi} \frac{(r^2 - p^2)^4}{(r^2 + p^2 - 2pr \cos \gamma)^{(n+4)/2}} dy H(\gamma) \Big\},$$

$$I_1 = \ell \left\{ J_{\frac{1}{2}} - J_0 + J_1 \right\}.$$

Analog wie oben kann gezeigt werden, daß der radiale Limes von $J_{\frac{1}{2}}$ und J_0 ebenfalls Null ist.

Es muß jetzt noch bewiesen werden, daß J_1 beschränkt ist, wenn x radial gegen z konvergiert.

$G(\gamma)$ ist nach γ stetig differenzierbar und besitze die Ableitung $g(\gamma)$. Dann folgt aus (6)

$$H(\gamma) = \int_0^{\gamma} 2r \sin \frac{\gamma}{2} g(\gamma) d\gamma,$$

und das Integral J_1 läßt sich wie folgt umformen

$$\begin{aligned} J_1 &= \int_0^{\phi} \frac{(r^2 - p^2)^4}{(r^2 + p^2 - 2pr \cos \gamma)^{(n+4)/2}} 2r \sin \frac{\gamma}{2} g(\gamma) d\gamma \\ &= \int_0^{\phi} \frac{(r^2 - p^2)^4}{(r^2 + p^2 - 2pr \cos \gamma)^{(n+4)/2}} 2r \sin \frac{\gamma}{2} dG(\gamma) \\ &= \int_{K_{z, \phi}^m} \frac{(r^2 - p^2)^4}{|x - y|^{n+4}} |z - y| dO_n(y). \end{aligned}$$

Offensichtlich ist $\frac{r^2 - p^2}{|x - y|} \leq 2r$ und weil $0 \leq \phi \leq \frac{\pi}{3}$

$$\frac{|z - y|}{|x - y|} \leq \frac{2}{3} \sqrt{3}. \text{ Damit ist}$$

$$J_1 \leq (2r)^3 \frac{2}{3} \sqrt{3} \int_{K_{z, \phi}^m} \frac{r^2 - p^2}{|x - y|^n} dO_n(y)$$

$$\leq (2r)^3 \frac{2}{3} \sqrt{3} \omega_n r \text{PI}(1, 2B)(x) = \text{const.}$$

Für den Beweis der Konvergenz der Integrale PI und Pr ist die Voraussetzung, daß die erste symmetrische Ableitung gleich Null ist, ausreichend.

Der Beweis des ersten Teiles dieses Hilfssatzes erfolgt analog zum obigen Beweis unter Verwendung der Zerlegung (2).

q. e. d.

Jetzt sind wir in der Lage, eine Aussage über nicht-tangentiale Limiten zu beweisen.

Hilfssatz 2 μ und η seien positive Maße auf $\mathcal{A}(\partial B)$ und Θ ein Winkel mit $0 \leq \Theta < \frac{\pi}{2}$.

$$D_B \mu(z) = D_B \eta(z) = 0.$$

Aus diesen Voraussetzungen folgt:

$$1. \quad \Theta - \lim_{x \rightarrow z} P(\mu, \eta, \partial B)(x) = 0.$$

2. Unter der zusätzlichen Voraussetzung $D_B^2 \mu(z) = 0$ ist auch

$$\Theta - \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta, \partial B)(x) = 0.$$

Beweis: Wir zeigen, daß der Θ -Limes jedes Integrals aus (2) bzw. (3) gleich Null ist.

Durch elementare Überlegungen läßt sich die folgende Aussage beweisen.

ν sei ein positives Maß auf $\mathcal{A}(\partial B)$. Ist E eine beliebige Kugel innerhalb des Kegels $E_z(\Theta)$, wobei $0 \leq \Theta < \frac{\pi}{2}$,

die hinreichend dicht bei z liegt, und $x; w \in E$, so existieren positive Konstanten K_1, \dots, K_4 , die nur von Θ abhängig sind, so daß die folgenden Ungleichungen gelten

$$\begin{aligned} PI(\nu, \partial B)(x) &\leq K_1 PI(\nu, \partial B)(w), \\ Pr(\nu, \partial B)(x) &\leq K_2 Pr(\nu, \partial B)(w), \\ Ps(\nu, \partial B)(x) &\leq K_3 Ps(\nu, \partial B)(w), \\ Pt(\nu, \partial B)(x) &\leq K_4 Pt(\nu, \partial B)(w). \end{aligned} \tag{8}$$

x sei jetzt hinreichend dicht innerhalb des Kegels $E_z(\Theta)$ bei z , und w sei die Projektion von x auf die Achse Oz des Kegels. Zu x und w existiert eine Kugel E , so daß $x; w \in E$.

Dann folgt aus der Ungleichung (8)

$$\frac{PI(\mu, \partial B)(x)}{PI(\mu, \partial B)(w)} \leq K_1$$

$$\begin{aligned} & |PI(\mu, \partial B)(w) - PI(\mu, \partial B)(x)| \\ &= PI(\mu, \partial B)(w) \left| 1 - \frac{PI(\mu, \partial B)(x)}{PI(\mu, \partial B)(w)} \right| \leq PI(\mu, \partial B)(w) |1 + K_1|. \end{aligned}$$

Da nach Hilfssatz 1 $R\text{-}\lim_{w \rightarrow z} PI(\mu, \partial B)(w) = 0$ ist, folgt

$$\text{daraus } \textcircled{C} \text{-}\lim_{x \rightarrow z} PI(\mu, \partial B)(x) = 0.$$

Ganz analog verläuft der Beweis für die übrigen Integrale.

q. e. d.

Definition μ sei ein signiertes Maß von beschränkter Variation auf $\mathcal{A}(\partial B)$ und besitze im Punkt $z \in \partial B$ die symmetrische Ableitung $D_g \mu(z) = a$ mit $|a| < \infty$.

Wir sagen, μ erfüllt die Voraussetzung D im Punkt $z \in \partial B$, wenn die folgenden Aussagen wahr sind.

Es existiert eine integrierbare Funktion f auf ∂B mit den Eigenschaften

1. f ist in einer gewissen Umgebung von z stetig.
2. f habe in Polarkoordinaten die Darstellung $f = f(\varphi_1, \dots, \varphi_{n-1})$, die in einer Umgebung von z einmal stetig differenzierbar nach $\varphi_1, \dots, \varphi_{n-1}$ ist.
3. Für das signierte Maß ν , das durch

$$\nu(D) = \mu(D) - \int_D f(y) dQ_n(y), \quad D \in \mathcal{A}(\partial B),$$

auf $\mathcal{A}(\partial B)$ erzeugt wird, sei $D_g^2 |\nu|(z) = 0$.

Hilfssatz 3 μ sei ein signiertes Maß von beschränkter Variation auf $\mathcal{A}(\partial B)$, das im Punkt $z \in \partial B$ die Voraussetzung D erfüllt. ν sei ein signiertes Maß von beschränkter Variation auf $\mathcal{A}(\partial B)$ mit $D_g \nu(z) = b$, $|b| < \infty$.

Für das signierte Maß ξ , das durch

$$\xi(D) = \eta(D) - b \int_D dO_n(y), \quad D \in \mathcal{A}(\partial B),$$

auf $\mathcal{A}(\partial B)$ erzeugt wird, gelte $D_B \left| \xi \right| (z) = 0$. Dann ist

$$\ominus \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta, \partial B)(x) = b.$$

Beweis: Da μ die Voraussetzung D im Punkt z erfüllt, existiert die Funktion f , und für das signierte Maß

$$\nu(D) = \mu(D) - \int_D f(y) dO_n(y), \quad D \in \mathcal{A}(\partial B),$$

gilt $D_B^2 |\nu|(z) = 0$. Aus Hilfssatz 2 folgt dann, daß der \ominus -Limes der einzelnen Integrale in (3) mit den positiven Maßen $|\nu|$ und $|\xi|$ gleich Null ist.

$$+ \ominus \lim_{x \rightarrow z} \frac{(n+2)}{4p^2} \Pr(|\nu|, \partial B)(x) + \dots + \dots + \ominus \lim_{x \rightarrow z} \frac{r^2}{p^2} \Pr(|\xi|, \partial B)(x)$$

$$\geq \left| - \ominus \lim_{x \rightarrow z} \frac{(n+2)}{4p^2} \Pr(\nu, \partial B)(x) - \dots - \dots - \ominus \lim_{x \rightarrow z} \frac{r^2}{p^2} \Pr(\xi, \partial B)(x) \right|$$

$$= \left| \ominus \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\nu, \xi, \partial B)(x) \right|$$

$$= \left| \ominus \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta, \partial B)(x) - \ominus \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(f, b, \partial B)(x) \right|$$

$$= \left| \ominus \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta, \partial B)(x) - b \right| = 0$$

q. e. d.

Satz μ und η seien signierte Maße von beschränkter Variation auf den Borelschen Teilmengen der Kugeloberfläche ∂B .

1. Es existiert der Limes $\ominus \lim_{x \rightarrow z} P(\mu, \eta, \partial B)(x)$

und ist gleich $D\mu(z)$ für O_n -f.a. $z \in \partial B$.

2. μ erfülle in O_n -f.a. Punkten der Kugeloberfläche die Voraussetzung D, dann existiert der Limes

$$\textcircled{+} \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta, \partial B)(x)$$

und ist gleich $D\eta(z)$ für O_n -f.a. $z \in \partial B$.

Beweis: Wir beginnen mit dem Beweis des zweiten Teiles des Satzes. Dazu betrachten wir die Zerlegung $\eta = \eta_a + \eta_s$ des signierten Maßes η bezüglich O_n . Dabei bezeichne η_a den absolutstetigen und η_s den singulären Anteil.

Weiter ist $\eta_s = \eta_s^+ - \eta_s^-$.

Nach /3/ S. 65 Satz 3.7 (ii) ist $D\eta_s^+(z) = 0$ und

$D\eta_s^- = 0$ O_n -f.ü.. Aus Hilfssatz 3 folgt dann

$$\textcircled{+} \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta_s^+, \partial B)(x) = 0 \text{ und}$$

$$\textcircled{-} \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta_s^-, \partial B)(x) = 0 \quad O_n\text{-f.ü.}$$

Es existiert eine integrierbare Funktion f , daß für jede Borelsche Teilmenge $D \in \mathcal{L}(\partial B)$

$$\eta_a(D) = \int_D f(y) dO_n(y) \text{ gilt.}$$

Nach /3/ S. 65 Satz 3.7 (i) ist $D\eta_a(z) = f(z)$ O_n -f.ü.,

$$(\eta_a - f(z)O_n)(D) = \int_D (f(y) - f(z)) dO_n(y)$$

$$\left| \eta_a - f(z)O_n \right| (D) = \int_D |f(y) - f(z)| dO_n(y) \quad D \in \mathcal{L}(\partial B)$$

Aus /3/ S. 66 Satz 3.8 ergibt sich

$$D \left| \eta_a - f(z)O_n \right| (z) = 0 \text{ für } O_n\text{-f.a. } z \in \partial B.$$

$$N_a = \left\{ z : z \in \partial B, D \left| \eta_a - f(z)O_n \right| (z) \neq 0 \right\}$$

$$\cup \left\{ z : z \in \partial B, \mu \text{ erfüllt in } z \text{ nicht die Voraussetzung D} \right\}$$

$$\cup \left\{ z : z \in \partial B, D \eta_a(z) \neq f(z) \right\}.$$

N_a hat das Oberflächenmaß Null. Dann ist für $z \notin N_a$

nach Hilfssatz 3

$$\textcircled{+} \lim_{x \rightarrow z} \frac{\partial}{\partial p^2} P(\mu, \eta_a, \partial B) = f(z) = D\eta_a(z)$$

Daraus folgt die Behauptung.

Der Beweis des ersten Teiles des Satzes verläuft analog zum Beweis des zweiten Teiles unter Verwendung der Zerlegung (2) und $\mu = \mu_a + \mu_s$.

q. e. d.

Literatur

- /1/ Boehme, T. K. Extensions of Fatou's theorem to tangential
Weiss, M. L. asymptotic values
Proc. Amer. math. Soc. 27, 289 - 298 (1971).
- /2/ Edenhofer, J. Eine Integraldarstellung der Lösung der
Dirichletschen Aufgabe bei der Polypotential-
gleichung im Falle einer Hyperkugel
Math. Nachr. 69, 149 - 162 (1975).
- /3/ Helms, L. L. Einführung in die Potentialtheorie
Walter de Gruyter, Berlin - New York 1973.
- /4/ Tsuji, M. On Fatou's theorem on Poisson integrals
Japan J. Math. 15, 13 - 18 (1938).

eingegangen: 9. 2. 1977

Anschrift des Verfassers

Dipl.-Math. Eckart Keller
Wilhelm-Pieck-Universität Rostock
Sektion Mathematik
DDR-25 Rostock
Universitätsplatz 1

Ein stochastischer Projektionsalgorithmus zur iterativen Lösung linearer Gleichungssysteme

1. Einleitung

Es wird ein stochastischer Projektionsalgorithmus zur iterativen Lösung konsistenter oder inkonsistenter linearer Gleichungssysteme beschrieben. Ist das System lösbar, so liefert der Algorithmus eine (die) Lösung, ist es widerspruchsvoll, so erhält man eine verallgemeinerte Lösung, die in Spezialfällen mit der besten Approximation im Sinne der Methode der kleinsten Quadrate übereinstimmt.

Gegeben seien das lineare Gleichungssystem

$$A \underline{x} = \underline{b} \quad (1)$$

mit einer $M \times N$ -Koeffizientenmatrix A und eine Anfangsnäherung \underline{x}^1 .

2. Allgemeiner Schritt des Verfahrens

Es seien \underline{x}^D und

$$\underline{r}^D = \underline{b} - A \underline{x}^D \quad (2)$$

Näherungs- bzw. Restvektor vor dem n -ten Schritt. Durch einen Zufallsgenerator erzeugen wir die Komponenten von L ($1 \leq L \ll \min(M, N)$) Vektoren $\underline{z}_1^D \in \mathbb{R}^N$ und berechnen die zugehörigen Bildvektoren

$$\underline{w}_1^D = A \underline{z}_1^D, \quad l = 1, 2, \dots, L. \quad (3)$$

In dem durch die \underline{w}_1^D aufgespannten Unterraum des \mathbb{R}^M bestimmen wir die beste Approximation von \underline{r}^D

$$\left\| \underline{x}^n - \sum_{l=1}^L u_l^n \underline{w}_l^n \right\| = \inf_{u_l} \left\| \underline{x}^n - \sum_{l=1}^L u_l \underline{w}_l^n \right\|. \quad (4)$$

Verwendet man die energetische Norm $\| \underline{x} \|_G = (\underline{x}^T G \underline{x})^{1/2}$ mit einer symmetrischen, positiv definiten $M \times M$ -Matrix G , so sind die u_l^n aus dem Gleichungssystem

$$\sum_{l=1}^L c_{il}^n u_l^n = d_i^n, \quad i = 1, 2, \dots, L, \quad (5)$$

zu bestimmen, wo die Koeffizienten Skalarprodukte bezeichnen:

$$c_{il}^n = (\underline{w}_i^n, \underline{w}_l^n)_G, \quad d_i^n = (\underline{w}_i^n, \underline{x}^n)_G, \quad i, l = 1, 2, \dots, L. \quad (6)$$

Ist L klein gegen M und N , so ist lineare Abhängigkeit der Vektoren \underline{w}_l^n unwahrscheinlich, die Gramschen Matrizen (c_{il}^n) sind also i. a. nichtsingulär, und die u_l^n können (z. B. durch das Cholesky-Verfahren) eindeutig bestimmt werden. Tritt trotzdem einmal eine singuläre Koeffizientenmatrix auf, so läßt man den betreffenden Unterraum aus.

Mit Hilfe der u_l^n berechnet man den neuen Näherungsvektor aus

$$\underline{x}^{n+1} = \underline{x}^n + \sum_{l=1}^L u_l^n \underline{z}_l^n \quad (7)$$

und den neuen Restvektor aus

$$\underline{r}^{n+1} = \underline{r}^n + \sum_{l=1}^L u_l^n \underline{w}_l^n. \quad (8)$$

3. Rechenaufwand

Der Rechenaufwand eines Schrittes ergibt sich aus der Erzeugung von $L \cdot N$ Pseudo-Zufallszahlen für die Komponenten der \underline{z}_l^n , maximal $L \cdot M \cdot N$ Multiplikationen für die Berechnung der \underline{w}_l^n ,

$M \cdot L \cdot (L+3)/2$ Multiplikationen für die Berechnung der Skalarprodukte c_{i1}^n und d_i^n , L Wurzeln, $(L^3+6L^2-7L)/6$ Multiplikationen und $(L^2+3L)/2$ Divisionen für das Cholesky-Verfahren und schließlich $L(M+N)$ Multiplikationen für die \underline{x} - und \underline{r} -Verbesserung. Insgesamt sind das

$$L(M \cdot N + 2N + M(1 + \frac{L+3}{2})) + \frac{L^2}{6} + \frac{3L}{2} + \frac{4}{3} \quad (9)$$

wesentliche Rechenoperationen pro Schritt, zu denen der Einfachheit halber auch das Wurzelziehen und die Erzeugung der Zufallszahlen gerechnet wurden.

4. Konvergenz

Zum Nachweis der Konvergenz fassen wir die L Zufallsvektoren \underline{z}_l^n , $l = 1, 2, \dots, L$, zu einer $N \times L$ -Matrix Z_n zusammen. Die Koeffizientenmatrix des Systems (5) hat dann die Gestalt $Z_n^T A^T G A Z_n$. Wir setzen voraus, daß sie nichtsingulär ist, führen die Abkürzung

$$G_n = Z_n (Z_n^T A^T G A Z_n)^{-1} Z_n^T \quad (10)$$

ein und schreiben den Algorithmus damit in der Form eines in-stationären linearen Iterationsverfahrens erster Stufe

$$\underline{x}^{n+1} = T_n \underline{x}^n + \underline{v}^n, \quad T_n = I_N - G_n A^T G A, \quad \underline{v}^n = G_n A^T G \underline{b} \quad (11)$$

mit den zugehörigen Restvektoren

$$\underline{r}^{n+1} = S_n \underline{r}^n, \quad S_n = I_M - A G_n A^T G. \quad (12)$$

I_N und I_M sind dabei Einheitsmatrizen entsprechenden Formats.

Man überzeugt sich leicht, daß die S_n G -orthogonale Projektionsmatrizen sind:

$$S_n^2 = S_n, \quad (G S_n)^T = G S_n. \quad (13)$$

Als solche haben sie die G -Norm 1:

$$\|S_n\|_G = 1. \quad (14)$$

Wir spalten den Raum \mathbb{R}^M in die G-orthogonale direkte Summe aus dem Wertebereich von A und dem Nullraum von $A^T G$ auf (zur folgenden Beweisführung vergl. z. B. /1/),

$$\mathbb{R}^M = \mathcal{R}(A) \oplus \mathcal{N}(A^T G), \quad \mathcal{R}(A) \perp_G \mathcal{N}(A^T G), \quad (15)$$

und bezeichnen den G-orthogonalen Projektor auf $\mathcal{N}(A^T G)$ längs $\mathcal{R}(A)$ mit $P_{\mathcal{N}}$ und den entsprechenden Projektor auf $\mathcal{R}(A)$ mit

$P_{\mathcal{R}}$. Die Restriktion von S_n auf $\mathcal{R}(A)$ sei \tilde{S}_n

$$S_n = P_{\mathcal{N}} + \tilde{S}_n, \quad \tilde{S}_n = S_n P_{\mathcal{R}}. \quad (16)$$

Für aufeinanderfolgende S_n gilt

$$S_{n+1} S_n = P_{\mathcal{N}} + \tilde{S}_{n+1} \tilde{S}_n, \quad \tilde{S}_{n+1} \tilde{S}_n = S_{n+1} S_n P_{\mathcal{R}}, \quad (17)$$

denn es ist $P_{\mathcal{N}}^2 = P_{\mathcal{N}}$ und $\tilde{S}_{n+1} P_{\mathcal{N}} = S_{n+1} P_{\mathcal{R}} P_{\mathcal{N}} = 0$. Auch

$P_{\mathcal{N}} \tilde{S}_n = P_{\mathcal{N}} S_n P_{\mathcal{R}}$ verschwindet, denn $P_{\mathcal{N}}$ projiziert auf $\mathcal{N}(A)$,

und S_n bildet Elemente aus $\mathcal{R}(A)$ wieder in diesen Teilraum ab.

Wir fassen nun Produkte von jeweils K aufeinanderfolgenden S_n

bzw. \tilde{S}_n zu Matrizen $S^{(i)}$ bzw. $\tilde{S}^{(i)}$ zusammen:

$$S^{(i)} = \prod_{n=iK}^{(i-1)K+1} S_n, \quad i = 1, 2, \dots. \quad (18)$$

Dabei durchläuft der Produktindex n die Zahlen von iK bis $(i-1)K+1$ in absteigender Folge. Die natürliche Zahl K wird so gewählt, daß die zugehörigen Vektoren \underline{w}_1^n , $n = (i-1)K+1, \dots, iK$; $l = 1, 2, \dots, L$, den Raum $\mathcal{R}(A)$ aufspannen. Das tritt mit Wahrscheinlichkeit 1 ein, wenn K durch

$$K - 1 \leq N/L \leq K \quad (19)$$

festgelegt wird, Dann ist

$$\|\tilde{S}^{(i)}\|_G < 1, \quad (20)$$

denn aus (18) und (17) ergibt sich mit (14)

$$\|\tilde{S}^{(1)}\|_G = \|S^{(1)}P_R\|_G \leq \|S^{(1)}\|_G \leq 1. \quad (21)$$

Wäre $\|\tilde{S}^{(1)}\|_G = 1$, so gäbe es ein $\underline{r} \in \mathbb{R}^M$, $\underline{r} \neq 0$, mit $\|\tilde{S}^{(1)}\underline{r}\|_G = \|\underline{r}\|_G$. Dieses \underline{r} läge in $\mathcal{R}(A)$, denn $\underline{r} \notin \mathcal{R}(A)$ führt auf

$$\|\tilde{S}^{(1)}\underline{r}\|_G \leq \|S^{(1)}\|_G \|P_R \underline{r}\|_G \leq \|P_R \underline{r}\|_G < \|\underline{r}\|_G.$$

Für Elemente aus $\mathcal{R}(A)$ ist aber $P_R \underline{r} = \underline{r}$, also $\|\tilde{S}^{(1)}\underline{r}\|_G =$

$\|S^{(1)}\underline{r}\|_G = \|\underline{r}\|_G$. In den Schritten $(i-1)K+1$ bis iK würde sich die Norm von \underline{r} nicht ändern, und das hieße wegen (4), daß alle d_i^n verschwinden, \underline{r} also G-orthogonal zu $\mathcal{R}(A)$, d. h. auch Element von $\mathcal{N}(A^T G)$ wäre. Daraus ergäbe sich $\underline{r} = \underline{0}$ im Widerspruch zur Annahme. - Wählt man K so groß, daß für alle i

$$\|\tilde{S}^{(i)}\|_G \leq q < 1 \quad (22)$$

ist, so gilt $\|\tilde{S}^{(i)}\tilde{S}^{(i-1)}\dots\tilde{S}^{(1)}\|_G \leq q^i$, also verschwindet das

Produkt der $\tilde{S}^{(i)}$ für $i \rightarrow \infty$, und man erhält wegen (17)

Satz 1: Die vom Algorithmus (5)-(8) erzeugten Restvektoren \underline{r}^n konvergieren für beliebige Startvektoren $\underline{r}^1 = \underline{b} - A \underline{x}^1$ mit Wahrscheinlichkeit 1 gegen

$$\underline{r}^\infty = \lim_{n \rightarrow \infty} \underline{r}^n = \lim_{n \rightarrow \infty} \prod_{j=1}^n S_j \underline{r}^1 = P_R \underline{r}^1 = P_R \underline{b}. \quad (23)$$

Um die Konvergenz der Näherungsvektoren \underline{x}^n nachweisen zu können, fassen wir je K Schritte zu einem Zyklus zusammen und bezeichnen die entsprechenden Näherungsvektoren mit \underline{y}^i ,

$$\underline{y}^1 = \underline{x}^1, \quad \underline{y}^{i+1} = T^{(i)} \underline{y}^i + R_i \underline{b}. \quad (24)$$

Dabei sind die $T^{(i)}$ analog zu (18) als Produkte der T_n definiert, und für die R_i gilt

$$R_i = \sum_{(i-1)K+1}^{iK} Q_{iK,j} G_j A^T G, \quad (25)$$

mit

$$Q_{n,j} = \prod_{k=n}^{j+1} T_k, \quad j = 1, 2, \dots, n-1; \quad Q_{n,n} = I_N. \quad (26)$$

Daraus erhält man mit Hilfe der Definitionen (11) und (12)

$$T^{(i)} = I_N - R_i A, \quad S^{(i)} = I_M - A R_i. \quad (27)$$

Führt man die Iteration (24) durch wiederholtes Einsetzen auf den Startvektor zurück, so folgt

$$\underline{x}^{i+1} = \hat{T}^{(i)} \underline{x}^1 + B^{(i)} \underline{b}, \quad \hat{T}^{(i)} = \prod_{j=n}^1 T^{(j)}, \quad (28)$$

wo die $B^{(i)}$ analog zu (25), (26) durch

$$B^{(i)} = \sum_{j=1}^i Q_j^{(i)} R_j, \quad Q_j^{(i)} = \prod_{k=i}^{j+1} T^{(k)}, \quad Q_i^{(i)} = I_N, \quad (29)$$

definiert sind. Daraus ergibt sich mit (27) für $\hat{T}^{(i)}$ und das analog erklärte Produkt $\hat{S}^{(i)}$ der $S^{(j)}$

$$\hat{T}^{(i)} = I_N - B^{(i)} A, \quad \hat{S}^{(i)} = I_M - A B^{(i)}, \quad (30)$$

und für die Differenz aufeinanderfolgender $B^{(i)}$

$$B^{(i+1)} - B^{(i)} = R_{i+1} \hat{S}^{(i)} = R_{i+1} \prod_{j=1}^i \hat{S}^{(j)}, \quad (31)$$

denn $R_{i+1} P_{\mathcal{N}}$ verschwindet wegen (25). Also erhält man wegen (22) die Abschätzung

$$\|B^{(i+1)} - B^{(i)}\|_G \leq \|R_{i+1}\|_G q^i, \quad (32)$$

und die $B^{(i)}$ konvergieren, da die $\|R_i\|_G$ beschränkt sind, gegen

$$B^{\infty} = \lim_{n \rightarrow \infty} \prod_{i=1}^n Q_{n,i} Z_i (Z_i^T A^T G A Z_i)^{-1} Z_i^T A^T G. \quad (33)$$

Man kann B^{∞} numerisch bestimmen, indem man die Iteration mit dem Startvektor $\underline{x}^1 = \underline{0}$ und den Einheitsvektoren des \mathbb{R}^M als rechter Seite durchführt. Als Grenzvektor der Iteration (7) bzw. (11) ergibt sich wegen (28) und (30)

$$\underline{x}^{\infty} = (I_N - B^{\infty} A) \underline{x}^1 + B^{\infty} \underline{b}. \quad (34)$$

Satz 2: Die vom Algorithmus (5)-(8) erzeugten Näherungsvektoren konvergieren für jeden Startvektor $\underline{x}^1 \in \mathbb{R}^N$ gegen den durch (34) definierten Grenzvektor \underline{x}^{∞} . Ist (1) lösbar, so ist \underline{x}^{∞} Lösung von (1), ist (1) widerspruchsvoll, so ist \underline{x}^{∞} bzgl. $A^T G$ verallgemeinerte Lösung von (1), d. h. genügt der Gleichung

$$A^T G (\underline{b} - A \underline{x}^{\infty}) = 0. \quad (35)$$

Die Matrix B^{∞} ist eine verallgemeinerte Inverse von A mit den Eigenschaften

$$A B^{\infty} A = A, \quad B^{\infty} A B^{\infty} = B^{\infty}, \quad (G A B^{\infty})^T = G A B^{\infty}, \quad (36)$$

also eine g-Inverse vom Typ $A_1^-(G)_r$ (bzgl. der Terminologie vgl. z. B. /2/).

Beweis: Nach (23) und (30) ist

$$\lim_{i \rightarrow \infty} \hat{S}^{(i)} = P_{\mathcal{N}} = I_M - A B^{\infty}. \quad (37)$$

Wegen $B^{\infty} P_{\mathcal{N}} = 0$ und $P_{\mathcal{N}} A = 0$ erhält man daraus die ersten beiden Eigenschaften (36), und die dritte folgt aus der G-Orthogonalität des Projektors $P_{\mathcal{N}}$. Wegen (15) ist $P_{\mathcal{N}} + P_{\mathcal{R}} = I_M$, also gilt mit (34)

$$A \underline{x}^{\infty} = A B^{\infty} \underline{b} = P_{\mathcal{R}} \underline{b}. \quad (38)$$

Ist (1) lösbar, also $\underline{b} \in \mathcal{R}(A)$, so ist $P_{\mathcal{N}} \underline{b} = \underline{b}$, d. h., \underline{x}^{∞} ist Lösung. Im allgemeinen Fall ist $\underline{b} - A \underline{x}^{\infty} = \underline{b} - P_{\mathcal{N}} \underline{b} = P_{\mathcal{N}} \underline{b}$ und folglich $A^T G (\underline{b} - A \underline{x}^{\infty}) = 0$.

Literatur

- /1/ Tanabe, K., Projection method for solving a singular system of linear equations and its applications, Numer. Math. 17, 203 - 214 (1971).
- /2/ Plemmons, R. J., Direct iterative methods for linear systems using weak splittings, Acta Univ. Carolinae, Math. Phys. 15, 117 - 120 (1974).

eingegangen: 14. 12. 1976

Anschrift des Verfassers:

Doz. Dr. Gerhard Maeß
Wilhelm-Pieck-Universität Rostock
Sektion Mathematik
DDR-25 Rostock
Universitätsplatz 1

Numerische Ermittlung des Trägheitsellipsoids eines starren Körpers (Massepunktsystem)

Zu einer gegebenen Menge von mit einer Masse versehenen Punkten $\{(m_1, x_{11}, x_{21}, x_{31})\}$ sind die Achsen eines Ellipsoids zu bestimmen, das bezüglich einer räumlichen Bewegung bei homogen verteilter Masse gleiche Trägheitseigenschaften wie die gegebene Punktmenge aufweist.

Dazu betrachtet man die Funktion

$$f(\underline{z}, \lambda; \underline{s}) = \lambda(1 - \underline{z}^2) - \sum_1 m_1 ((\underline{x}_1 - \underline{s})^2 - (\underline{z}^T \cdot (\underline{x}_1 - \underline{s}))^2) \quad (1)$$

mit

$$\underline{s}^T = (s_1, s_2, s_3), \quad s_k = \sum_1 m_1 x_{k1} / \sum_1 m_1 \quad \text{und} \quad (2)$$

$$\underline{x}_1^T = (x_{11}, x_{21}, x_{31}) .$$

Die Funktion f entspricht im zweiten Term dem Trägheitsmoment gegen eine Drehung um \underline{z} , falls $|\underline{z}| = 1$ und \underline{z} durch \underline{s} verläuft. Die Achse \underline{z} wird lokal optimal bezüglich des Trägheitsmoments, falls

$$\lambda \underline{z} = A \underline{z}, \quad A = (a_{jk}) \quad \text{mit} \quad (3)$$

$$a_{jk} = \sum_1 m_1 (x_{k1} - s_k)(x_{j1} - s_j) = \sum_1 m_1 x_{k1} x_{j1} - s_k s_j \sum_1 m_1 . \quad (4)$$

Die Schwerpunktkoordinaten können auch als Komponenten des ersten Momentes von Eingabedaten \underline{x}_1 mit den Häufigkeiten m_1 aufgefaßt werden und die Koeffizienten von A als Komponenten des zweiten Momentes. Zur Berechnung der Wölbung sowie der Schiefe der Verteilung benötigte man darüber hinaus noch das vierte bzw. dritte Moment der Eingabedaten, deren Komponenten in vier- bzw. dreidimensionalen symmetrischen Tensoren entsprechender

Darstellung enthalten sind, die hier aber nicht weiter betrachtet werden (vgl. /1/). Aus dem gegebenen physikalischen Zusammenhang ergibt sich, daß die Matrix A positiv (semi-) definit ist, ferner ist sie ersichtlich symmetrisch, d. h., die Eigenwertaufgabe (3) hat reelle, nicht negative Eigenwerte λ_i und man kann letztere mit einem modifizierten Potenzverfahren (Mises-Iteration) nacheinander bestimmen:

Vorbereitungsschritt: $\lambda_i = 0, \underline{e}_i = \underline{0}, i = 1, 2, 3,$

Anfangswert \underline{z} vorgeben,

$$\begin{aligned} \text{Zyklus:} \quad \underline{z}' &= A \cdot \left(\underline{z} - \sum_{i=1}^3 (\underline{e}_i, \underline{z}) \underline{e}_i \right) \\ &= A \cdot \underline{z} - \sum_{i=1}^3 (\underline{e}_i, \underline{z}) \lambda_i \underline{e}_i \end{aligned} \quad (5)$$

$$\Lambda = |\underline{z}'|$$

$$\underline{z} = \underline{z}' / \Lambda$$

Test: Λ hinreichend wenig gegenüber letztem Iterationsschritt geändert? Nein: Zyklus erneut durchlaufen

Eintragung: $\underline{e}_k = \underline{z}, \lambda_k = \Lambda$

Test: Sind bereits alle EW und EV bestimmt? Nein: Anfangswert erneut festlegen

Über das Konvergenzverhalten des Potenzverfahrens zur EW- und EV-Berechnung gibt es hinreichend bekannte Aussagen (vgl. /3/). Zur hier gegebenen Modifizierung muß lediglich vermerkt werden, daß die Iteration bei ungeänderter Koeffizientenmatrix A für noch nicht bestimmte EV jeweils in einem Unterraum ausgeführt wird, der auf allen bereits ermittelten EV orthogonal ist. Damit ergibt sich unmittelbar das qualitativ gleiche Konvergenzverhalten von (5) für jeweils weitere ermittelte Eigenvektoren. Um das zu erkennen, muß man sich nur den iterierten Vektor \underline{z} als Linearkombination der Eigenvektoren zerlegt denken

(vgl. /2/, S. 331). Im vorliegenden physikalischen Zusammenhang (Dimension gleich 3, geringfügig verschiedene EW können wie zusammenfallende EW behandelt werden, ohne die physikalische Aussage wesentlich zu beeinträchtigen) sind die genannten Mittel als angemessen anzusehen, zumal eine entlang einer Geraden angeordnete Punktmenge (Hauptanwendungsfall) zu einem stark dominanten EW ($\lambda_1 \gg \lambda_i, i \neq 1$) führt und damit eine approximierende "Schwergerade" dann mit hinreichender Schnelligkeit (lineare Konvergenzgeschwindigkeit, Faktor $\ll 1$) berechenbar ist. Im betrachteten Fall (Dimension 3, Definitheit und Symmetrie) kann im übrigen durch eine EW-Verschiebung, an Stelle von (3) betrachtet man $(A + \alpha I)\underline{x} = (\lambda + \alpha)\underline{x}$, immer eine hinreichende Konvergenzgeschwindigkeit (linear, Faktor 0.5) erzielt werden, vorausgesetzt, die Verschiebungsgröße α ist geeignet ausgewählt worden, wobei die unbekanntenen EW selbstverständlich nicht für die Auswahl von α herangezogen werden können. Hier wird (ohne weitere Begründung) empfohlen, am Anfang der Iteration einmalig bei der Bestimmung des größten EW eine Verschiebung α unter der Bedingung

$$\min_i |a_{ii} + \alpha| \geq \max_{i \neq j} |a_{ij}|, \quad (6)$$

wo wenigstens einmal das Gleichheitszeichen gilt, vorzunehmen. Im anschließend angegebenen Beispiel wäre demgemäß $\alpha = -5$ bzw. $\alpha = -116$ (dort die ungünstigere Möglichkeit für eine Auswahl von α) zu wählen.

Wir kommen nun auf die physikalische Interpretation zurück. Das Trägheitsellipsoid mit dem Mittelpunkt \underline{s} und den Halbachsen

$\sqrt{\lambda_k} \underline{e}_k$ sowie der homogen verteilten Masse $m = \sum_1^3 m_i$ weist bei

räumlicher Bewegung das gleiche Massenträgheitsmoment auf, wie die ursprünglich gegebene Massepunktmenge. Je nach Rang von A kann es ersetzt werden durch den Schwerpunkt \underline{s} ($\text{Rg}(A) = 0$) mit der Gesamtmasse m , durch 3 Punkte (Masse = $m - 2$ bzw. 1, Lage = \underline{s} bzw. $\underline{s} \pm \sqrt{\lambda_1/2} \underline{e}_1$, $\text{Rg}(A) = 1$), durch 5 Punkte (Masse = $m - 4$ bzw. 1, Lage = \underline{s} bzw. $\underline{s} \pm \sqrt{\lambda_1/2} \underline{e}_1, i = 1, 2$,

$R_g(A) = 2$) oder durch 7 Punkte (Masse = $m - 6$ bzw. 1, Lage = \underline{s} bzw. $= \underline{s} \pm \sqrt{\lambda_1/2} \underline{e}_i$, $i = 1, 2, 3$, $R_g(A) = 3$). Im letzteren Fall wird das Trägheitsellipsoid durch $\underline{x}^T A^{-1} \underline{x} = 1$ analytisch beschrieben.

Betrachten wir ein Beispiel:

i	1	2	3	4	5	6	7	8	9	10	11	12						
m_i	1	1	2	2	2	2	1	3	3	3	2	2	m	24	A =	76.5	39.5	55.5
x_{1i}	0	1	2	3	2	2	5	-1	0	-1	2	-2	s_1	0.75		39.5	76.5	55.5
x_{2i}	0	1	3	2	2	2	5	0	-1	-1	-2	2	s_2	0.75		55.5	55.5	60.5
x_{3i}	0	1	2	2	3	2	5	-1	-1	0	0	0	s_3	0.75				

k		1		2		3	Achse 1	Achse 2	Achse 3
λ_k		171.5		37		5			
e_{1k}	·	57735	-	.70711	-	.40825	7.561	4.301	-0.913
e_{2k}	·	57735	.70711	-	.40825		7.561	-4.301	-0.913
e_{3k}	·	57735	0		.81650		7.561	0	1.826

Punkte 1,2	Punkte 3,4	Punkte 5,6
0.75 ± 5.346	0.75 ± 3.041	0.75 ± -0.6455
0.75 ± 5.346	0.75 ± -3.041	0.75 ± -0.6455
0.75 ± 5.346	0.75	0.75 ± 1.291

Falls $\lambda_{1,2} > \lambda_3 = 0$ ($R_g(A) = 2$), liegen sämtliche Punkte \underline{x}_i in der durch \underline{e}_1 und \underline{e}_2 aufgespannten Ebene

$$0 = (\underline{x} - \underline{s})^T \cdot (\underline{e}_1 \times \underline{e}_2) . \quad (7)$$

Man kann daher mit (7) die Punktmenge \underline{x}_i approximieren, falls $\lambda_{1,2} \gg \lambda_3$. Dabei entspricht der Quotient λ_3 / λ_2 der Genauigkeit der Approximation.

Im obigen Beispiel approximiert die Ebene

$$\begin{aligned}
 & ((x_1, x_2, x_3) - (0.75, 0.75, 0.75)) \cdot (0.408, 0.408, -0.8165)^T = \\
 & = 0.408x_1 + 0.408x_2 - 0.8165x_3 = 0
 \end{aligned}$$

die gegebene Punktmenge, wobei $\lambda_3 / \lambda_2 = 0.15$ eine verhältnismäßig geringe Genauigkeit der Approximation signalisiert.

In praktischen Anwendungen benötigt man allerdings eine Approximation von Punkt Mengen durch Gerade oder Ebene kaum. Punkt Mengen (etwa als Ergebnis einer Aufmeßreihe) sind vielmehr durch geeignete Kurven bzw. Flächen zu approximieren, wobei in der Regel bereits Informationen über die analytische Form der approximierenden Flächen vorliegen. Man kann nun z. B. die gegebene Punktmenge durch eine Linearkombination

$$\sum_{k=1}^3 c_k f_k(\underline{x}) \text{ approximieren, indem man nach der Abbildung}$$

$$\underline{x}'_k = f_k(\underline{x}), \quad k = 1, 2, 3 \quad (8)$$

das Trägheitsellipsoid der Punktmenge $\{\underline{x}'_k\}$ bestimmt und damit die Funktion

$$(\underline{x}' - \underline{e}')^T \cdot (\underline{e}_1 \times \underline{e}_2) = (\underline{f}(\underline{x}) - \underline{e}')^T \cdot (\underline{e}_1 \times \underline{e}_2) = 0 \quad (9)$$

zur Approximation der Punktmenge gewinnt. Wiederholt man diesen Vorgang mehrfach, dient der Quotient λ_3 / λ_2 der Orientierung über die Genauigkeit der gewonnenen Approximation.

Betrachten wir nochmals das obige Beispiel und verwenden die Funktionen

$$\underline{x}'_1 = f_1(\underline{x}) = 0.408x_1 + 0.408x_2 - 0.8165x_3 \quad (\text{bisherige beste Approximation})$$

$$\underline{x}'_2 = f_2(\underline{x}) = (x_1 - x_2)^2$$

$$\underline{x}'_3 = f_3(\underline{x}) = x_3^2.$$

Man erhält $m = 24$,

$$\begin{array}{rcl}
 s_1 = -.000104 & e_{i1} = & .00701 \quad -.00449 \quad .999965 \\
 s_2 = 3.083333 & e_{i2} = & .81535 \quad .578966 \quad -.003117 \\
 s_3 = 3.08333 & e_{i3} = & -.57893 \quad .81534 \quad .007719 \\
 & \lambda_i = & 952.257 \quad 515.466 \quad 4.94116 \\
 & i = & 1 \quad 2 \quad 3
 \end{array}$$

A =

$$\begin{array}{r}
 4.998 \quad 4.0877 \quad -5.7133 \\
 4.0877 \quad 805.83 \quad -206.17 \\
 -5.7133 \quad -206.17 \quad 661.83
 \end{array}$$

und damit

$$\begin{aligned}
 & 0.999965(0.408x_1 + 0.408x_2 - 0.8165x_3) - 0.003117(x_1 - x_2)^2 \\
 & + 0.007719 x_3^2 - 0.014086 = 0
 \end{aligned}$$

als approximierende Fläche.

Der Quotient $\lambda_3/\lambda_2 = 0.00959$ ist etwa um den Faktor 10^{-1} kleiner als im 1. Approximationsschritt.

Abschließend geben wir ein weiteres anschauliches Beispiel zu einer nicht-linearen Approximation:

$$x'_1 = f_1(\underline{x}) = x_1^2 + x_2^2 + x_3^2$$

$$x'_2 = f_2(\underline{x}) = x_1 + x_2 + x_3$$

$$x'_3 = f_3(\underline{x}) = x_1 \cdot x_2 \cdot x_3$$

$$\begin{aligned}
 i &= 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \\
 m_i &= 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\
 x_{1i} &= 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\
 x_{2i} &= 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\
 x_{3i} &= 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1
 \end{aligned}$$

$$\begin{aligned}
 A &= \\
 6 \quad 6 \quad 1.5 \\
 6 \quad 6 \quad 1.5 \\
 1.5 \quad 1.5 \quad 0.875
 \end{aligned}$$

$$\begin{aligned}
 m &= 8 \quad k = 1 \quad 2 \quad 3 \\
 s_1 &= 1.5 \quad e_{1k} = .6954 \quad -.1281 \quad -.5 \\
 s_2 &= 1.5 \quad e_{2k} = .6954 \quad -.1281 \quad .5 \\
 s_3 &= .125 \quad e_{3k} = .18116 \quad .98345 \quad 0 \\
 &\quad \lambda_k = 12.39 \quad .48423 \quad 0
 \end{aligned}$$

Als approximierende Fläche (die Punkte liegen exakt auf ihr) erhalten wir

$$0 = 0.5(x_1 + x_2 + x_3) - 0.5(x_1^2 + x_2^2 + x_3^2), \text{ d. h.}$$

$$0 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 + (x_3 - 0.5)^2 - 0.75 .$$

Literatur

- /1/ Frieberger, W. F., International Dictionary of Applied
van Nostrand, D. Mathematics, Princeton 1960.
- /2/ Gastinel, N., Lineare numerische Analysis, Berlin 1972
- /3/ Kieseletter, H., Elementare Methoden der numerischen
Maß, G. Mathematik, Berlin 1974.

eingegangen: 14. 3. 1977

Anschrift des Verfassers

Dr. Helmut Thielcke
 Wilhelm-Pieck-Universität Rostock
 Sektion Mathematik
 DDR-25 Rostock
 Universitätsplatz 1

Eine Methode zur Abschätzung des Quadraturfehlers im 2-Dimensionalen

Zusammenfassung:

Zur numerischen Integration von Funktionen werden als Quadraturformeln Hermite-Entwicklungen verwandt. Unter Einführung von Hilfsfunktionen wird eine Abschätzung für den Quadraturfehler im 2-Dimensionalen hergeleitet, die auf den Satz von Rolle zurückgeht.

Bei der numerischen Integration von Funktionen

$$F(u, v) = \{X(u, v), Y(u, v), Z(u, v)\} \text{ mit } u \in [u_0, u_1], \\ v \in [v_0, v_1] \quad (1)$$

spielen Quadraturformeln eine große Rolle /1,2,3/. Als Quadraturformeln für gewisse numerische Fragestellungen haben sich Hermite-Entwicklungen der Form

$$P(u, v) = \sum_{k, l=0}^1 \sum_{i, j=0}^n F_{ij}(u_k, v_l) \cdot f_{ik}(u) \cdot f_{jl}(v) \quad (2)$$

bewährt, mit

$$F_{ij}(u, v) := \frac{\partial^{i+j}}{\partial u^i \partial v^j} F(u, v) \quad (3)$$

und den bekannten Mischfunktionen $f_{ik}(t)$:

$$f_{ik}(t) = 0 \text{ für } t \in [t_0, t_1] \quad (4)$$

$$f_{ik}(t) \text{ n-mal stetig differenzierbar} \quad (4')$$

$$\left. \frac{d^j}{dt^j} f_{ik}(t) \right|_{t=t_1} = \begin{cases} 1 & \text{für } j=i \text{ und } k=l \\ 0 & \text{sonst} \end{cases} \quad (4'')$$

für $i, j=0, 1, \dots, n$ und $k, l=0, 1$.

Die Forderungen (4)...(4'') sind durch Polynomansätze $(2n+1)$ -ten Grades realisierbar /6,7/.

Das Problem der numerischen Berechnung von Integralen

$$I(F) = \int_{u_0}^u \int_{v_0}^v du dv F(u, v) \quad (5)$$

besteht darin, daß in (5) für $F(u, v)$ ein interpolierender Ausdruck verwandt wird:

$$\bar{I}(P) = \int_{u_0}^u \int_{v_0}^v du dv P(u, v) \quad (6)$$

$$I(P) = \sum_{i, j=0}^n \sum_{k, l=0}^1 F_{ij}(u_k, v_l) \int_{u_0}^u du f_{ik}(u) \int_{v_0}^v dv f_{jl}(v) \quad (7)$$

In /6/ sind einige Spezialfälle von (7) ausgeführt. Wir betrachten die Maximumnorm über

$$IG = \left\| \int_{u_0}^u \int_{v_0}^v du dv [F(u, v) - P(u, v)] \right\| \text{ für } u \in [u_0, u_1] \quad (8)$$

und $v \in [v_0, v_1]$. Zur Abschätzung von (8) führen wir eine Hilfsfunktion ein, die es uns gestattet, die Betrachtungen auf die bekannten Techniken im 1-Dimensionalen zurückzuführen /7/:

$$K2(u, v) = \sum_{i=0}^n \sum_{k=0}^1 f_{ik}(u) [F_{i0}(u_k, v) - P_{i0}(u_k, v)] \quad (9)$$

bzw.

$$K1(u, v) = \sum_{j=0}^n \sum_{l=0}^1 f_{j1}(v) [F_{0j}(u, v_1) - P_{0j}(u, v_1)]. \quad (10)$$

Durch die Anwendung von $K1(u, v)$ bzw. $K2(u, v)$ in (8) werden die entsprechenden Randkurven von $F(u, v)$ und $P(u, v)$ ausgeglichen. Wir betrachten

$$\begin{aligned} IG &= \left\| \int_{u_0}^u \int_{v_0}^v du dv [F(u, v) - P(u, v)] \right\| = \\ &= \left\| \int_{u_0}^u \int_{v_0}^v du dv F - [P + K1] + K1 \right\| \leq \\ &\leq \left\| \int_{u_0}^u \int_{v_0}^v du dv F(u, v) - [P(u, v) + K1(u, v)] \right\| + \\ &+ \left\| \int_{u_0}^u \int_{v_0}^v du dv K1(u, v) \right\| \leq \\ &\leq \int_{u_0}^u \int_{v_0}^v du dv \|F(u, v) - [P(u, v) + K1(u, v)]\| + \\ &+ \int_{u_0}^u \int_{v_0}^v du dv \|K1(u, v)\|. \end{aligned} \quad (11)$$

Die einzelnen Integranden von (11) werden nach folgender Vorgehensweise abgeschätzt:

Es wird eine Hilfsfunktion $H1(u, v)$ mit $u=u^X$ beliebig aber fest und v variabel eingeführt:

$$H1(u^X, v) = F(u^X, v) - [P(u^X, v) + K1(u^X, v)] - c(v-v_0)^{n+1}(v-v_1)^{n+1} \quad (12)$$

Unter Anwendung des Satzes von Rolle ergeben sich dann nach /5/ folgende Abschätzungen:

$$\|F(u, v) - [P+K1]\| \leq \frac{\|F_0\|}{(2n+2)!} \cdot (1-t)^{n+1} \cdot t^{n+1} \cdot k_1^{2n+2}, \quad (13)$$

$$\begin{aligned} \|K1(u, v)\| &\leq \\ &\leq \sum_{j=0}^n \sum_{l=0}^1 \frac{\|F_{2n+2-j}(u, v_1)\|}{(2n+2)!} h_1^{2n+2} (1-t)^{n+1} t^{n+1} \|f_{j1}\| \end{aligned} \quad (14)$$

Somit gilt:

$$\int_{u_0}^{u_1} \int_{v_0}^{v_1} du dv \|F - [P+K1]\| \leq \frac{\|F_0\|}{(2n+2)!} k_1^{2n+3} \cdot c \cdot h_1 \quad (15)$$

$$\begin{aligned} &\int_{u_0}^{u_1} \int_{v_0}^{v_1} du dv \|K1(u, v)\| \leq \\ &\leq \sum_{j=0}^n \sum_{l=0}^1 \|f_{j1}\| \frac{\|F_{2n+2-j}(u, v_1)\|}{(2n+2)!} c \cdot h_1^{2n+3} \cdot k_1 \end{aligned} \quad (16)$$

mit

$$c = \int_0^1 dt t^{n+1} (1-t)^{n+1} \quad \text{und} \quad h_1 = u_1 - u_0 \quad \text{und} \quad (17)$$

$$k_1 = v_1 - v_0.$$

Mit (15) und (16) läßt sich eine Abschätzung für (11) finden:

$$\begin{aligned} &\left\| \int_{u_0}^{u_1} \int_{v_0}^{v_1} du dv [F(u, v) - P(u, v)] \right\| \leq \\ &\leq c h_1 k_1 \frac{\|F_0\| \|k_1\|^{2n+2} + \sum_{j=0}^n \sum_{l=0}^1 \|f_{j1}\| \|F_{2n+2-j}(u, v_1)\| h_1^{2n+2}}{(2n+2)!} \end{aligned} \quad (18)$$

Unter Verwendung von K2(u, v) ergibt sich:

$$\left\| \int_{u_0}^{u_1} \int_{v_0}^{v_1} du dv [F(u,v) - P(u,v)] \right\| \leq$$

$$\leq c h_1 k_1 \frac{\|F\|_{2n+2} \|h_1\|^{2n+2} + \sum_{i=0}^n \sum_{k=0}^1 \|F_{ik}\| \|F_j\|_{2n+2}(u_k, v) \|k_1\|^{2n+2}}{(2n+2)!}$$

(19)

Die Formeln (18) und (19) bewerten den Quadraturfehler, der bei der numerischen Integration auftritt, wenn als Quadraturformeln Hermite-Entwicklungen n-ter Ordnung genommen werden. Für n=0 und n=1 ergeben sich aus (18) bzw. (19) Fehlerformeln für häufig verwandte Spezialfälle von (7).

Literatur

- /1/ Ahlberg, J. H., E. N. Nilson and J. L. Walsh:
The theory of splines and their applications.
Ac. Press, New-York-London, 1967.
- /2/ Altman, W.: Fehlerabschätzung nach Sard für ein- und zwei-
dimensionale Polynominterpolation mit Anwen-
dungen auf Differentialgleichungen. Diplomar-
beit, Universität Rostock, 1972.
- /3/ Antes, H.: Splinefunktionen bei der Lösung von Integral-
gleichungen. Num. Math. 19, 166 - 172 (1972).
- /4/ Fehlaue, K.-U.: Interpolation und Darstellung von Flächen-
stücken im Zusammenhang mit dem DIGRA 73-Sy-
stem, Dissertation, Rostock, 1976.
- /5/ Fehlaue, K.-U.: Fehlerabschätzung bei der Interpolation
von Flächenstücken. Erscheint demnächst im
Rost. Math. Koll.

- /6/ Kieseewetter, H.: Interpolation mit Intervallfunktionen und einige Anwendungen. Wiss. Zeitschr. Univ. Rostock, Math.-Nat. Reihe 20, 337 - 341 (1971).
- /7/ Kieseewetter, H. und Maeß, G.: Elementare Methoden der numerischen Mathematik. Akademie-Verlag Berlin, 1974.

eingegangen: 1. 2. 1977

Anschrift des Verfassers:

Dr. Klaus-Uwe Fehlauer
Wilhelm-Pieck-Universität Rostock
Rechenzentrum
DDR-25 Rostock

Über Abschlußeigenschaften von biologisch motivierten Sprachen

1. Einleitung

1968 führte A. Lindenmayer L-Systeme (formale Grammatiken) als mathematisches Modell zur Beschreibung der Entwicklung von Organismen ein. Schon bald danach setzte die Untersuchung von Abschlußeigenschaften der verschiedenen Typen von L-Sprachen gegenüber Operationen ein. Dabei galt das Interesse vor allem den AFL-Operationen Vereinigung, Konkatenation, Kleeneschem $*$ -Operator, Homomorphismen, inversen Homomorphismen und Durchschnitten mit regulären Mengen und den mengentheoretischen Operationen. Es ergaben sich nur sehr schwache Abschlußeigenschaften. In dieser Arbeit untersuchen wir im Kapitel 3 die Abschlußeigenschaften von L-Sprachen gegenüber den Operationen

- i) $d_R(L) = \{q : \exists p (p \in R \wedge pq \in L)\}$,
 $b_R(L) = \{q : \exists p (p \in R \wedge qp \in L)\}$,
- ii) $Shuf_R(L) = \{w_1y_1w_2y_2 \dots w_ny_n : n \geq 1, w_1 \dots w_n \in L, y_1 \dots y_n \in R\}$
- iii) $Ashuf_R(L) = \{x_1y_1x_2y_2 \dots x_ny_n : x_1 \dots x_n \in L, y_1 \dots y_n \in R,$
 $x_i, y_j \in V \cup \{\lambda\},$
 $x_i = \lambda \text{ impliziert } x_{i+1} = \lambda ,$
 $y_j = \lambda \text{ impliziert } y_{j+1} = \lambda \}$
- iv) Abbildungen, die von Zwei-Band-Automaten erzeugt werden, auf deren einem Band ein Element aus R steht, wobei R eine reguläre Menge, V das zugrundegelegte Alphabet und λ das Leerwort ist. Gegenüber diesen Operationen ist jede volle AFL abgeschlossen.

Den L-Sprachen sehr verwandt sind die τ -Sprachen von Alad'ev, die von linearen homogenen Strukturen erzeugt werden. In Kapitel 4 geben wir die Abschlußeigenschaften von τ -Sprachen gegenüber den Operationen i) .. iv) an.

In Kapitel 5 untersuchen wir das Verhalten von Eindeutigkeit und Mehrdeutigkeit (im Sinne von Reedy und Savitch) von OL-Sprachen bei Operationen.

2. Zwei-Band-Automaten

In /2/ definierten Ginsburg und Greibach eine volle AFL (abstract family of languages) als eine Familie von Sprachen, die gegenüber Vereinigung, Konkatenation, Kleeneschem $*$ -Operator, Homomorphismen, inversen Homomorphismen und Durchschnitten mit regulären Mengen abgeschlossen ist. Sie bewiesen ferner, daß eine volle AFL \mathcal{L} auch gegenüber Abbildungen, die von verallgemeinerten sequentiellen Maschinen induziert werden, und den Operationen i) und ii) der Einleitung abgeschlossen ist (/2/, Corollary 1, Corollary 2 zu Theorem 2.1. und Lemma 5.1.).

Wir wollen jetzt noch zwei weitere Operationen angeben, unter denen eine AFL abgeschlossen ist.

Ein Zwei-Band-Automat ist ein 8-Tupel

$M = (K, X, Y, \sigma, \lambda, s_0, S_1, S_2)$, wobei

- i) K, X, Y , endliche Mengen sind (Zustände, Eingabe, Ausgabe),
- ii) σ ist eine Abbildung von $K \times X$ in K ,
- iii) λ ist eine Abbildung von $K \times X$ in Y^* ,
- iv) $s_0 \in K$ (Initialzustand),
- v) $S_1 \cup S_2 = K, S_1 \cap S_2 = \emptyset$.

Sei $(x_{11}x_{12}\dots x_{1k}, x_{21}x_{22}\dots x_{2m}) \in X^* \times X^*$ und

$M = (K, X, Y, \sigma, \lambda, s_0, S_1, S_2)$ ein Zwei-Band-Automat. Ferner

sei $z_0 = x_f(0)$ und $s_0 \in S_f(0)$. Dann setzen wir $s_1 = \sigma(s_0, z_0)$.

Seien s_0, s_1, \dots, s_n bereits definiert und $x_{11}\dots x_{1r}$ und

$x_{21}\dots x_{2t}$ mit $n = r+t$ bereits abgearbeitet. Außerdem sei noch

$s_n \in S_f(n)$. Ist $f(n) = 1$, so definieren wir $s_{n+1} = \sigma(s_n, x_{1,r+1})$,

falls $x_{1,r+1}$ existiert, und stoppen im anderen Fall (ohne

$x_{2,t+1}\dots x_{2m}$ abzarbeiten). Ist $f(n) = 2$, so setzen wir

$s_{n+1} = \delta(s_n, x_{2,t+1})$, falls $x_{2,t+1}$ existiert, und stoppen im anderen Fall (ohne $x_{1,t+1} \dots x_{1k}$ abzuarbeiten). Seien s_0, s_1, \dots, s_n und z_0, z_1, \dots, z_n alle so sukzessiv definierten Eingaben und Zustände. Dann sei

$$M(x_{11} \dots x_{1k}, x_{21} \dots x_{2m}) = \lambda(s_0, z_0) \lambda(s_1, z_1) \dots \lambda(s_n, z_n).$$

Für Mengen $L, R \subseteq X^*$ setzen wir

$$M_R(L) = \{M(w_1, w_2) : w_1 \in L, w_2 \in R\}.$$

Lemma 1: Sei \mathcal{L} eine volle AFL und M ein Zwei-Band-Automat.

Dann ist $M_R(L)$ für jede reguläre Menge R und jedes $L \in \mathcal{L}$ wieder in \mathcal{L} .

Folgerung: Sei \mathcal{L} eine volle AFL, $L \in \mathcal{L}$ und R eine reguläre Menge. Dann gilt $\text{Ashuf}_R(L) \in \mathcal{L}$.

Sowohl das Lemma wie auch die Folgerung beweist man wie in /3/ Theorem 2.1. und Corollary.

Lemma 2: Sei $h : X \rightarrow X'$ ein Homomorphismus mit $h(a) \in \{a, \lambda\}$. Dann existieren ein Zwei-Band-Automat M und eine reguläre Menge R , die für jede Sprache L über X' die Beziehung

$$M_{RC}(Lc) = h^{-1}(L)$$

erfüllen, wobei $c \notin X$ ist.

Beweis: Es ist $X \setminus X'$ die Menge der $a \in X$ mit $h(a) = \lambda$. Dann setzen wir $R = ((X \setminus X')^* d)^+$ mit $d \notin X$. R ist sicher regulär.

Wir definieren nun M durch

$$M = (\{s_0, s_1, s_2, s_3\}, X \cup \{c, d\}, X, \delta, \lambda, s_0, \{s_1, s_3\}, \{s_0, s_2\})$$

mit

$$\lambda(s_i, x) = x \text{ für } x \in X$$

$$\lambda(s_i, d) = \lambda(s_i, c) = \lambda \quad \text{für } i = 0, 1, 2, 3,$$

$$\delta(s_0, x) = s_0 \text{ für } x \in X, \quad \delta(s_0, c) = s_3, \quad \delta(s_0, d) = s_1,$$

$$\delta(s_1, x) = s_0 \text{ für } x \in X, \quad \delta(s_1, c) = s_2, \quad \delta(s_1, d) = s_1,$$

$$\delta(s_2, x) = s_2 \text{ für } x \in X \cup \{c, d\},$$

$$\delta(s_3, x) = s_3 \text{ für } x \in X \cup \{c, d\}.$$

Man verifiziert nun leicht $M_{RC}(Lc) = h^{-1}(L)$.

3. Abschlußeigenschaften von L-Sprachen

Wir geben zuerst die Definitionen für die Standardtypen von L-Sprachen. (Für eine ausführliche Diskussion der Begriffe siehe /7/.)

Ein ETOL-System ist ein Quadrupel $G = (V, V_1, \mathcal{K}, w)$ mit

- i) V ist eine endliche Menge, V_1 ist eine nicht leere Teilmenge von V ,
- ii) $w \in V^+$,
- iii) $\mathcal{K} = \{P_1, P_2, \dots, P_r\}$, wobei jedes P_i folgende Bedingungen erfüllt:
 - 1) P_i ist eine endliche Teilmenge von $V \times V^*$,
 - 2) $pr_1(P_i) = V$ (pr_1 bezeichne die Projektion auf die erste Koordinate).

Für $(a, \alpha) \in P_i$ schreiben wir meist $a \rightarrow \alpha$.

Ein ETOL-System mit

- a) $V = V_1$ heißt TOL-System,
- b) $r = 1$ heißt EOL-System,
- c) $V = V_1$ und $r = 1$ heißt OL-System.

Ein System heißt fortpflanzend (propagating, abgekürzt durch P), falls mit $(a, \alpha) \in P_i$ stets $\alpha \neq \lambda$ gilt. Ein System heißt deterministisch (abgekürzt durch D), falls aus $a \rightarrow \beta \in P_i$,

$a \rightarrow \gamma \in P_i$ stets $\beta = \gamma$ folgt. So erhalten wir EDTOL-, PTOL-,

DOL-, EPOL-, EPDTOL-Systeme usw.

Sei $G = (V, V_1, \{P_1, \dots, P_r\}, w)$ ein ETOL-System. Wir sagen,

$x \in V^+$ generiert $y \in V^*$, geschrieben als $x \Rightarrow y$, wenn

- i) $x = x_1 x_2 \dots x_k$, $x_i \in V$, $1 \leq i \leq k$,
- ii) $y = y_1 y_2 \dots y_k$ ist und
- iii) ein j , $1 \leq j \leq r$, so existiert, daß $x_i \rightarrow y_i \in P_j$ für $i=1, 2, \dots, k$ ist.

$\xRightarrow{*}$ bezeichne den transitiven und reflexiven Abschluß von \Rightarrow . Die von G erzeugte Sprache definieren wir als

$$L(G) = \{x : w \xrightarrow{*} x, x \in V_1^*\}.$$

L heißt X-Sprache, $X \in \{ETOL, TOL, EOL, OL\}$, falls ein X-System G mit $L = L(G)$ existiert.

Die Menge aller X-Sprachen bezeichnen wir mit $\mathcal{F}(X)$.

Bei diesen Systemen erfolgt die Ersetzung kontextfrei. Wenn k linke und l rechte Nachbarn erfaßt werden, spricht man von $\langle k, l \rangle$ L-Systemen, notfalls mit den Zusätzen E, P, D, T. Eine Sprache heißt IL-Sprache, wenn k, l so existieren, daß l eine $\langle k, l \rangle$ L-Sprache ist (notfalls erneut mit den entsprechenden Zusätzen).

Satz 1: Es gilt die folgende Tabelle, wobei + bzw. - bedeutet, daß die entsprechende Sprachfamilie $\mathcal{F}(X)$ bzgl. der angegebenen Operation abgeschlossen bzw. nicht abgeschlossen ist.

	d_R	b_R	$Shuf_R$	$Ashuf_R$	M_R
OL	-	-	-	-	-
TOL	-	-	-	-	-
EOL	+	+	-	+	-
ETOL	+	+	+	+	+
IL	-	-	-	-	-
EIL	+	+	+	+	+
ETIL	+	+	+	+	+

Beweis: i) Da alle betrachteten Sprachfamilien gegenüber der Inversion (reversal) abgeschlossen sind, folgt die Gleichheit der Spalten für d_R und b_R sofort.

ii) Die Resultate für $\mathcal{F}(ETOL)$, $\mathcal{F}(EIL)$ und $\mathcal{F}(ETIL)$ folgen daraus, daß diese Sprachfamilien volle AFL's sind. (vgl. /7/)

iii) Die negativen Aussagen bzgl. M_R folgen für $\mathcal{F}(OL)$,

$\mathcal{F}(TOL)$, $\mathcal{F}(IL)$ aus der Tatsache, daß die entsprechenden Familien gegenüber Homomorphismen nicht abgeschlossen sind (vgl. /7/, /11/, /13/), und der Möglichkeit, jeden Homomorphismus durch einen Zwei-Band-Automaten zu simulieren. Für $\mathcal{F}(EOL)$ und M_R folgt die Aussage aus Lemma 2, da R_c eine reguläre Sprache ist,

$\{a^{2^i}c : i \geq 0\} \in \mathcal{F}(\text{EOL})$ wegen /5/ gilt und $h^{-1}(a^{2^i} : i \geq 0)$ für $h(a) = a, h(b) = \lambda$ nach /5/ keine EOL-Sprache ist.

iv) $L_1 = \{c, ba^2, da^4\} \in \mathcal{F}(\text{OL})$ wird von dem OL-System

$G_1 = (\{a, b, c, d\}, \{a \rightarrow a, b \rightarrow da^2, c \rightarrow ba^2, d \rightarrow d\}, c)$ erzeugt,

und es gilt $L_2 = \{a^{2^i} : i \geq 2\} \in \mathcal{F}(\text{OL})$. Nach /11/ sind aber

$d_{\{b, d\}}(L_1) = \{a^2, a^4\}$ und

$\text{Shuf}_{\{a^2\}}(L_2) = \text{Ashuf}_{\{a^2\}}(L_2) = \{a^{2^i+2} : i \geq 2\}$

nicht in $\mathcal{F}(\text{TOL})$. Damit sind auch die restlichen Aussagen für $\mathcal{F}(\text{OL})$ und $\mathcal{F}(\text{TOL})$ bewiesen.

v) Die Abgeschlossenheit von $\mathcal{F}(\text{EOL})$ gegenüber d_R folgt aus /8/ Theorem 9.5., /12/ Theorem 4.2 und dem Fakt, daß $\mathcal{F}(\text{EOL})$ gegenüber endlichen Substitutionen abgeschlossen ist (vgl. /9/).

vi) Da $\text{Shuf}_{b^*}(\{a^{2^i} : i \geq 0\}) = h^{-1}(\{a^{2^i} : i \geq 0\})$ mit

$h(a) = a, h(b) = \lambda$ gilt, ist nach /5/ gezeigt, daß $\mathcal{F}(\text{EOL})$ gegenüber Shuf_R nicht abgeschlossen ist.

vii) Seien $L \in \mathcal{F}(\text{EOL})$ und R eine reguläre Sprache, beide über X . Wir definieren nun die Substitutionen

$f : x \rightarrow \{(a, x)(b, y) : y \in X\} \cup \{(c, x)\},$

$f' : x \rightarrow \{(a, y)(b, x) : y \in X\} \cup \{(d, x)\},$

wobei a, b, c, d Symbole nicht aus X sind. Dann ist nach /9/ und /12/ Theorem 4.2.

$L_1 = f(L) \cup f(L)\{(d, x) : x \in X\}^* \in \mathcal{F}(\text{EOL}),$

und andererseits ist

$R_1 = f'(R) \cup f'(R)\{(c, x) : x \in X\}^*$

eine reguläre Sprache. Daher ist

$L_1 \cap R_1 = \{(a, x_1)(b, y_1) \dots (a, x_n)(b, y_n)(q, z_1) \dots (q, z_m) :$

$q \in \{c, d\}, x_1, y_1, z_1 \in X,$

$(x_1 \dots x_n z_1 \dots z_m \in L, y_1 \dots y_n \in R \text{ für } q = c),$

$(x_1 \dots x_n \in L, y_1 \dots y_n z_1 \dots z_m \in R \text{ für } q = d)\}$

eine EOL-Sprache. Ferner sei h der Homomorphismus $h((g,x)) = x$ für $g \in \{a,b,c,d\}$, $x \in X$. Dann ist $Ashuf_R(L) = h(L_1 \cap R_1) \in \mathcal{F}(EOL)$.

$$viii) L = \{f\} \cup \{ca^{2n}b^n d : n \geq 1\} \cup \{c'a^n b^{2n} d' : n \geq 1\}$$

ist eine POL-Sprache, die von

$$G = (\{a,b,c,d,f,c',d'\}, \{a \rightarrow a, b \rightarrow b, c \rightarrow ca^2, d \rightarrow bd, c' \rightarrow c'a, d' \rightarrow b^2 d', f \rightarrow ca^2 b d, f \rightarrow c' a b^2 d'\}, f)$$

erzeugt wird. Dann ist

$$L_1 = d\{c,c'\}(L) = \{a^{2n}b^n d : n \geq 1\} \cup \{a^n b^{2n} d' : n \geq 1\}.$$

1. Fall: $L_1 \notin \mathcal{F}(IL)$. Dann ist bewiesen, daß $\mathcal{F}(IL)$ gegenüber d_R nicht abgeschlossen ist.

2. Fall: $L_1 \in \mathcal{F}(IL)$. Dann ist nach /13/, Theorem 2 auch

$$L_2 = L_1^R = \{db^n a^{2n} : n \geq 1\} \cup \{d'b^{2n} a^n : n \geq 1\} \notin \mathcal{F}(IL).$$

$d\{d,d'\}(L_2) = \{b^n a^{2n} : n \geq 1\} \cup \{b^{2n} a^n : n \geq 1\} \notin \mathcal{F}(IL)$ gilt aber nach /13/ Lemma 3.

ix) Wir betrachten die Sprache

$$L = Shuf_{\{a,a^2\}}(\{a^{2^i} : i \geq 0\}) = Ashuf_{\{a,a^2\}}(\{a^{2^i} : i \geq 0\}) = \{a^{2^i+1} : i \geq 0\} \cup \{a^{2^i+2} : i \geq 0\},$$

und zeigen $L \notin \mathcal{F}(IL)$.

Dazu nehmen wir an, daß $L = L(G)$ für ein $\langle k,l \rangle$ -System G gilt.

Wir setzen $m = k + l$ und beweisen zuerst, daß es ein $u \geq 2$ mit

$\langle a^k, a, a^l \rangle \rightarrow a^u$ gibt. Angenommen, es wäre für

$\langle a^k, a, a^l \rangle \rightarrow a^u$ immer $u \in \{0,1\}$. Ist immer $u = 0$, so ist L endlich, womit wir den Widerspruch hätten. Es existiere also die

Regel $\langle a^k, a, a^l \rangle \rightarrow a$. Wir wählen $t = 2^i + 1 \geq m$. a^s sei das Wort maximaler Länge, das aus a^k am Wortanfang von a^t direkt abge-

leitet werden kann, und a^r sei entsprechend für das Wortende

a^l definiert. Dann ist $a^s a^{t-m} a^r = a^{t+s+r-m}$ das Wort maximaler

Länge, das aus a^t direkt ableitbar ist. Ist $s + r - m \leq 0$, so

ist erneut L endlich. Ist $s + r - m = 1$, so generiert a^{t+1}

das Wort $a^s a^{t+1-m} a^r = a^{t+2} \notin L$. Ist $s+r-m > 1$, so wird für hinreichend großes i die Differenz $2^{i+1}+1 - (s^i+1) = 2^i$ größer als $s+r-m$, und dann ist das von a^t erzeugte Wort $a^{t+s+r-m} \notin L$. Mit dem existierenden $u \geq 2$ gilt daher für hinreichend großes i $a^{2^i+1} \Rightarrow a^{s+r+(2^i+1-m)} u = a^{2^j+\delta}$ mit $j \geq i$, $\delta \in \{1,2\}$, und folglich

$$a^{2^i+2} \Rightarrow a^{s+r+(2^i+2-m)} u = a^{2^j+\delta+u} \notin L.$$

Dieser Widerspruch zeigt $L \notin \mathcal{F}(IL)$.

Damit ist Satz 1 bewiesen.

Bemerkungen:

1. Betrachten wir nur die Familien $\mathcal{F}(OL)$, $\mathcal{F}(TOL)$, $\mathcal{F}(EOL)$ und $\mathcal{F}(IL)$ und fortpflanzende Systeme, so bleiben alle Aussagen der Tabelle in Satz 1 gültig.

2. Betrachten wir erneut nur die Familien aus 1. und nur deterministische Systeme, so erhalten wir in allen Fällen ein negatives Resultat. Die fehlenden Aussagen für $\mathcal{F}(DEOL)$ beweist man mittels /5/ Theorem 9 leicht.

3. In /11/ bzw. /13/ und /14/ wird sogar noch mehr gezeigt, da bewiesen wird, daß der Abschluß unter den dort betrachteten Operationen von $\mathcal{F}(DOL)$ bzw. $\mathcal{F}(POL)$ nicht in $\mathcal{F}(TOL)$ bzw. $\mathcal{F}(IL)$ enthalten ist. Bezüglich unserer Operationen gilt das gleiche.

4. In /14/ wird der Fall des einelementigen Alphabets diskutiert. Bezüglich der in der vorliegenden Arbeit betrachteten Operationen ist $\mathcal{F}(UOL)$ in $\mathcal{F}(IL)$ nicht abgeschlossen.

4. Abschlußeigenschaften von Sprachen, die von linearen homogenen Strukturen erzeugt werden

Wir betrachten den endlichen deterministischen Automaten

$A = (V^D, V, \delta)$ mit V^D - Eingangsalphabet

V - Zustandsmenge

δ - Überföhrungsfunktion.

$0 \in V$ sei ein ausgezeichneteter Zustand, und es gelte

$\delta((0, \dots, 0), 0) = 0$. Es seien unendlich viele Kopien des Automaten A vorhanden, die durch die ganzen Zahlen indiziert seien

$$\dots A_{-2}, A_{-1}, A_0, A_1, A_2, \dots$$

Sei nun $q_i(t)$ der Zustand von A_i im Moment t . Dann gelte

$$q_i(t+1) = \delta((q_i(t), q_{i+1}(t), \dots, q_{i+n-1}(t)), q_i(t)).$$

Wir schreiben dafür kurz

$$[q_i(t), \dots, q_{i+n-1}(t)] \rightarrow q_i(t+1).$$

Die Folge von Zuständen $w = (\dots, q_{-2}(t), q_{-1}(t), q_0(t), q_1(t), \dots)$

heiße Konfiguration im Takt t . Wir legen fest, daß für $t = 0$ nur endlich viele $q_i(0)$ von 0 verschieden sind. Diese Eigenschaft gilt dann wegen obiger Forderung für alle t . Wir betrachten nur noch den maximalen Teil einer Konfiguration, der mit einem $z \neq 0$ beginnt und mit einem $z' \neq 0$ endet. Es entsteht dann eine Folge von Maximalteilen von Konfigurationen

$$w_0 \Rightarrow w_1 \Rightarrow w_2 \rightarrow \dots$$

wobei w_t die gekürzte Konfiguration zur Zeit t ist. Dann sei

für die unendliche indizierte Menge G von Automaten A

$$\{L(G) = \{w_i : i \geq 0\}.$$

Es ist klar, daß jede Sprache $L(G)$ über V von diesem Typ durch ein $D(n, n)$ L-System erzeugt wird. $\mathcal{F}(\tau_n)$ sei die Menge aller so erzeugten Sprachen, bei denen Automaten mit n Eingängen verwendet werden, und $\mathcal{F}(\tau)$ sei die Menge aller so erzeugten Sprachen.

Lemma 3: Es gilt

$$L_{r,s,t} = \{a^i b a^j : i \geq r, j \geq s\} \cup \{a^k : k \geq t\} \notin \mathcal{F}(\tau),$$

$$L_{r,t} = \{a^i b a^j : j \geq i \geq r\} \cup \{a^k : k \geq t\} \notin \mathcal{F}(\tau)$$

für alle r, s, t .

Beweis: Wir führen den Beweis nur für $r = s = 0, t = 1$. In den anderen Fällen sind nur geringfügige Modifikationen erforderlich.

Angenommen $L = L_{0,0,1}$ wäre für ein n eine τ_n -Sprache. In meh-

rerer Schritten werden wir diese Annahme zum Widerspruch führen.

i) $[a, \dots, a] \rightarrow a$

Sei zuerst $[a, \dots, a] \rightarrow b$. Dann könnten wir ein Wort mit mindestens zwei b's aus a^{n+1} erzeugen.

Sei deshalb $[a, \dots, a] \rightarrow 0$. Wir erhalten jetzt für alle $i \geq 0$
 $a^{2n+i} \Rightarrow w_1 \dots w_{n-1} 0^{n+1+i} v_1 \dots v_{n-1} = w$ mit $w_i, v_j \in \{0, a, b\}$ und

$w_1 \dots w_{n-1} = 0^{n-1}$ oder $v_1 \dots v_{n-1} = 0^{n-1}$ (da L kein Wort mit 0's

enthält). Deshalb erzeugen alle Worte a^{2n+i} das gleiche Wort w, und es gilt

$$a^{2n+1} \Rightarrow w \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_p \Rightarrow a^{2n+k} \Rightarrow w \Rightarrow u_1 \Rightarrow \dots,$$

womit L als endlich nachgewiesen ist. Damit ist i) gezeigt.

ii) $\underbrace{[a, \dots, a, 0, \dots, 0]}_{k \text{ mal}} \rightarrow b$ gilt für kein k

Wenn $[a, \dots, a, 0, \dots, 0] \rightarrow b$ gelten würde, hätten wir
 $[a, \dots, a, b, a, \dots, a] \rightarrow a$, weil wir kein Wort ableiten können,
das mehr als ein b enthält. Wir betrachten das Wort

$w = a^{4n} b a^{4n}$. w kann nur von einem Wort w' der Länge $l(w') \geq 7n$
erzeugt werden. Weil

$w' = a^i \Rightarrow w_1 b y_1 \dots y_{k-1} \neq w$ und $w' = a^j b a^l \Rightarrow w_2 b x_1 \dots x_{k-1} \neq w$
für alle i und l, die größer als n sind, gilt ($y_m, x_m \in \{0, a, b\}$),

ist $w' = a^j b a^l$ mit $l \leq n$. Daher ist $j \geq 6n$, und somit gilt

$$w' \Rightarrow x_1 \dots x_{n-1} a^{5n} w_3 \neq w.$$

iii) $\underbrace{[a, \dots, a, 0, \dots, 0]}_{k \text{ mal}} \rightarrow a$ gilt für ein k, $1 \leq k \leq n-1$, nicht.

Nehmen wir das Gegenteil an. Dann gilt $w a^{n+i} \Rightarrow w' a^{n+i}$, und
somit kann nur eine endliche Anzahl von Worten der Form $a^k b a^j$
mit $j < n$ erzeugt werden.

iv) Es gibt ein k, $1 \leq k \leq n-1$, so daß $\underbrace{[a, \dots, a, 0, \dots, 0]}_{k' \text{ mal}} \rightarrow 0$
für alle $k' \leq k$ gilt.

Dies folgt aus ii) und iii) und dem Fakt, daß kein Wort in L den Buchstaben 0 enthält.

v) $[0, \dots, 0, \underbrace{a, \dots, a}_{1 \text{ mal}}] \rightarrow b$ gilt für kein $l, 1 \leq l \leq n$.

Dies wird analog zu ii) bewiesen, und wie oben schließen wir auf

vi) Gilt $[0, \dots, 0, \underbrace{a, \dots, a}_{1 \text{ mal}}] \rightarrow a$ für ein $l, 1 \leq l \leq n-1$, dann

ist für alle $l' \leq l$ ebenfalls $[\underbrace{0, \dots, 0}_{l' \text{ mal}}, a, \dots, a] \rightarrow a$.

Wenn nun $k \leq l$ ist, dann können wir nur Worte a^m aus a^r mit $m \geq r > n$ erzeugen. Dann enthält L nur endlich viele Worte mit einem b. Dies beweist $l < k$.

(*) Aus a^r können nur Worte $a^{r'}$ mit $r' < r$ abgeleitet werden, wenn $r > n$ ist.

Weiterhin existiert eine Regel $[\underbrace{a, \dots, a}_h, b, a, \dots, a] \rightarrow b$,

da sonst vor bzw. hinter b nur beschränkt viele a's stehen können. Weiterhin gibt es sicher nur ein solches h. Somit erhalten wir

$$v = a^{n+m} b a^{n+m'} \implies a^{n+m+l-h} b a^{n+m'+h-k} = v'.$$

Wenn $h \geq l$ ist, dann stehen in v' höchstens soviel a's vor b wie in v. Gleiches gilt bei $h < l$ und damit $h < k$ für die Anzahl der a's hinter b. Deshalb kann $a^i b a^j$ für hinreichend große i und j nur aus einem Wort aus a^* erzeugt werden. Dies widerspricht (*).

Wir erhalten den gleichen Widerspruch wenn wir annehmen, daß vi') gilt mit

vi') $[\underbrace{0, \dots, 0}_l, a, \dots, a] \rightarrow 0$ gilt für alle l.

Satz 2: $\mathcal{F}(\tau)$ ist nicht abgeschlossen gegenüber Shuf_R ,

Ashuf_R , d_R und M_R .

Beweis: Die folgenden Beziehungen gelten

$$\text{Shuf}_{a^*}(\{b, a\}) = I_{0,0,1} \text{ und } \text{Ashuf}_{a^* b a^* \cup a^*}(\{a\}) = I_{1,0,1}.$$

Damit folgen die ersten beiden Aussagen des Satzes aus Lemma 3.

$L = \{a^n b a^n : n \geq 1\}$ ist eine τ_3 -Sprache, wie man leicht nachprüft, und nun ergibt sich die Aussage für d_R aus

$d_{a^* \cup a^* b}^d(L) = L_{0,1}$. Die Nichtabgeschlossenheit von $L(\tau)$ ergibt

sich aus der Nichtabgeschlossenheit dieser Sprachfamilie gegenüber Homomorphismen.

5. Eindeutigkeit und Operationen

In /4/ diskutierten Ginsburg und Ullian die folgende Fragestellung: Ist das Resultat einer Operation eine eindeutige (bzw. mehrdeutige) kontext-freie Sprache, wenn die Ausgangssprachen es waren. In diesem Paragraphen untersuchen wir das gleiche Problem für OL-Sprachen. Da $\mathcal{F}(OL)$ gegenüber den meisten Operationen nicht abgeschlossen ist, machen wir die Zusatzvoraussetzung, daß nach der Ausführung der Operation eine OL-Sprache vorliege. Wir betrachten dabei nur die Oberflächenmehrdeutigkeit im Sinne von Reedy und Savitch /10/.

Bezüglich eines OL-Systems G heißen $x \in L(G)$ und $y \in L(G)$ äquivalent, wenn $x \Rightarrow y$ und $y \Rightarrow x$ gilt. Ein OL-System G heißt mehrdeutig, wenn $x_1, x_2, x_3 \in L(G)$ so existieren, daß sie paarweise inäquivalent sind bezüglich G und $x_1 \Rightarrow x_3$ und $x_2 \Rightarrow x_3$ gilt.

Sonst heißt G eindeutig. Eine OL-Sprache L heißt eindeutig, wenn ein eindeutiges OL-System G mit $L = L(G)$ existiert. Sonst heißt L mehrdeutig.

Beispiele:

1. $L_1 = \{\lambda, a, ab\}$ ist eindeutig, da das determinierte OL-System $G_1 = (\{a, b\}, \{a \rightarrow \lambda, b \rightarrow a\}, ab)$ L_1 erzeugt, und DOL-Systeme sind nach /10/ Theorem 20 eindeutig.

2. $L_2 = \{a\}^*$ ist eindeutig, denn es gibt ein L_2 erzeugendes System, bezüglich dessen nur die Äquivalenzklassen $[a]$ und $[\lambda]$ existieren (siehe /10/).

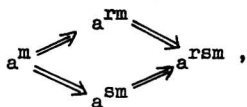
3. $L_3 = \{a^2, a, \lambda\}$ ist mehrdeutig, denn L_3 wird nur durch das

System $G_3 = (\{a\}, \{a \rightarrow a, a \rightarrow \lambda\}, a^2)$ erzeugt.

4. $L_4^m = \{a^n : n \geq m\}$ ist für $m \geq 1$ mehrdeutig. Sei G ein OL-System mit $L(G) = L_4^m$. Man erkennt sofort, daß G fortpflanzend

sein muß. Folglich ist a^m das Axiom. Weiterhin muß es in P mindestens zwei Produktionen geben, da sonst

$L(G) = \{a^{mr^1} : r \geq 0\} \neq L_4^m$ wäre. Seien $a \rightarrow a^r$ und $a \rightarrow a^s$ mit $r \neq s$ zwei Produktionen aus P . Dann gilt



und wir erhalten die gewünschte Mehrdeutigkeit von G .

5. $L_5 = \{a, ab, b^2, b, \lambda\}$ ist mehrdeutig.

Als Produktionen für b kommen in G nur $b \rightarrow w \in \{b, \lambda\}$ in Frage, da sonst $b^2 \Rightarrow ww \notin L_5$ gelten würde. Wir diskutieren nun die beiden folgenden Fälle:

i) $b \rightarrow \lambda \in P$.

Dann gilt $b \Rightarrow \lambda$ und $b^2 \Rightarrow \lambda$, und damit ist die Mehrdeutigkeit von G gezeigt, da b und b^2 inäquivalent sind.

ii) $b \rightarrow \lambda \notin P$.

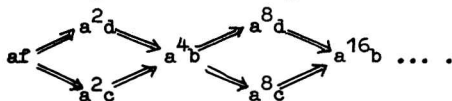
Dann ist $ab \xRightarrow{*} a$ unmöglich, und somit ist a das Axiom. Ferner muß $a \Rightarrow ab$ gelten und daher auch $ab \Rightarrow ab^2 \notin L_5$, womit wir den gewünschten Widerspruch hätten.

6. $L_6 = \{af\} \cup \{a^{4^n} b : n \geq 1\} \cup \{a^{2 \cdot 4^n} c : n \geq 0\} \cup \{a^{2 \cdot 4^n} d : n \geq 0\}$ ist mehrdeutig.

Eine Regel in der Form $a \rightarrow x \notin \{a\}^*$ würde $a^4 b \Rightarrow xxxxy \notin L$ liefern.

Daher hat jede Produktion für a die Form $a \rightarrow a^r$. Wenn $r \leq 1$ in allen Fällen wäre, so ließen sich die Sprünge von $a^{2^s} x$ zu

$a^{2^{s+1}}$ nicht erzeugen. Ferner kann es keine zwei Regeln der Form $a \rightarrow a^r$ und $a \rightarrow a^s$, $r \neq s$ geben, da dann wegen $a^{2^k} b \xrightarrow{*} a^{2^{s'}} b$ auch $a^{2^{s'} + (r-s)} b \notin L_G$ erzeugbar wäre. Also gibt es für a nur eine Produktion, und diese hat die Form $a \rightarrow a^r$ mit $r \geq 2$. Ferner ist G fortpflanzend, da sonst ein Element aus $\{a\}^*$ erzeugbar wäre. Somit ist a^d das Axiom, und a^{2^d} ist nur aus a^d herleitbar. Diese Ableitung ist aber nur durch $a \rightarrow a^2$ und $f \rightarrow d$ realisierbar. Also ist $r = 2$. Nun stellt man sofort fest, daß nur der folgende Ableitungsbaum in Frage kommt, und der zeigt die Mehrdeutigkeit von G .



Satz 3: Sei f eine n -stellige Operation zwischen Sprachen, und sei $L = f(L_1, L_2, \dots, L_n) \in \mathcal{F}(OL)$ für die OL-Sprachen L_1, \dots, L_n .

- i) Ist $n = 1$, f die Inversion (reversal), so ist mit L_1 auch L eindeutig bzw. mehrdeutig.
- ii) Die Eindeutigkeit bzw. Mehrdeutigkeit bleibt bei folgenden Operationen nicht erhalten: Vereinigung, Konkatenation, + bzw. * - Operator, Homomorphismen, Durchschnitten mit regulären Mengen, d_R , b_R , $Shuf_R$, $Ashuf_R$, Abbildungen, die von verallgemeinerten sequentiellen Maschinen erzeugt werden, und deren inversen Abbildungen.

Beweis: i) folgt leicht, da man die Inversion $L = L_1^R$ von L durch $P' = \{a \rightarrow a^R : a \rightarrow a \in P\}$ erzeugen kann und die entsprechenden Ableitungsbäume isomorph sind. (P bezeichnet dabei die Menge der Produktionen, um L zu generieren.)
 ii) beweist man mittels vorstehender Beispiele und den Aussagen aus /10/ sehr leicht.

Literatur

- /1/ V. Alad'ev, Operationen zwischen Sprachen, die von \mathcal{T}_n -Grammatiken erzeugt werden, Comm. Math. Univ. Carolinae 15, 211 - 220 (1974).
- /2/ Ginsburg, S. und Greibach, S., Abstract Families of Languages, Memoirs of the American Math. Society 87, 1 - 32 (1969).
- /3/ Ginsburg, S. und Spanier, E. H., Mappings of languages by two-tape -devices, Journal ACM 12, 423 - 434 (1965).
- /4/ Ginsburg, S. und Ullian, J., Preservation of unambiguity and inherent ambiguity in context-free languages, Journal ACM 13, 364 - 368 (1966).
- /5/ G. T. Herman, Closure properties of some families of languages associated with biological systems, Inf. and Control 24, 101 - 121 (1974).
- /6/ Herman. G. T., Lee, K. P., Leeuwen, J. van, und Rozenberg, G., Characterization of unary developmental languages, Discrete Math. 6, 235 - 247 (1973).
- /7/ Herman, G. T. und Rozenberg, G., Developmental systems and languages. North-Holland, 1975.
- /8/ Hopcroft, J. E. und Ullman, J. D., Formal Languages and Their Relation to Automata Theory. Reading 1969.

- /9/ Leeuwen, J. van, Notes on pre-set pushdown automata. In
Rozenberg, G. und Salomaa, A.: L-Systems,
Springer Lecture Notes in Computer
Science 15, (1974).
- /10/ Reedy, A. und Savitch, W., Ambiguity in the developmental
systems of Lindenmayer. Techn. rep.,
1974.
- /11/ Rozenberg, G., TOL systems and languages, Inf. and Con-
trol 23, 357 - 381 (1973).
- /12/ Rozenberg, G., Theory of L systems: from the point of
formal language theory. In Rozenberg, G.
und Salomaa, A.: L Systems. Springer Lec-
ture Notes in Computer Science 15.
- /13/ Rozenberg, G. und Lee, K. P., Some properties of the class
of L languages with interaction, Journal
Comp. Syst. Sc. 11, 129 - 147 (1975).
- /14/ Rochonen, K., Three results of comparison between L
languages with and without interaction.
Techn. rep., University of Turku (1975).

eingegangen: 20. 1. 1977

Anschrift des Verfassers

Dr. rer. nat. Jürgen Dassow
Wilhelm-Pieck-Universität Rostock
Sektion Mathematik
DDR - 25 Rostock
Universitätsplatz 1

Einige Eigenschaften der Programmiersprache BALG

1. Einleitung

Die Programmiersprache BALG /Go75/ ist der Versuch eine Systemprogrammiersprache ausgehend von den Charakteristiken höherer Programmiersprachen zu schaffen. Systemprogrammieren ist die Konstruktion von Programmsystemen, welche den Rahmen, die Umgebung, für die Lösung von Anwendungsproblemen abgeben (vgl. /Sa71/). Programmiersysteme bauen diese Umgebung stufenweise auf, indem sie Eigenschaften der zugrundeliegenden Maschine verdecken und durch neue, "höhere" Konstruktion ersetzen, welche sich besser als Grundlage für die weitere Programmierung eignen.

Von einer Systemprogrammiersprache wird man daher erwarten, daß sie neben den "üblichen" Ablauf- und Datenstrukturen vor allem Hilfsmittel für diese stufenweise Schichtung von Programmteilen, für die Modularisierung und die Kontrolle der Modulschnittstellen anbietet. Solche Hilfsmittel zur Beherrschung der Programmstruktur unterstützen den Systemprogrammierer auch im Hinblick auf den nicht unerheblichen Umfang und die Komplexität seiner Aufgaben, die besondere Anstrengungen auf dem Gebiet der Programmorganisation erforderlich machen.

Viele bisherige Systemprogrammiersprachen sind als Vorläufer und unter dem Zeitdruck größerer Systemprogramme, namentlich Betriebssysteme, entstanden (vgl. etwa PL360 /W168/, PS 440 /Go70/, System SUE Sprache /CL74/). Dies hatte zur Folge, daß die Probleme der untersten Schichten von Betriebssystemen: Beherrschung von Spezialbefehlen, namentlich für die E/A- und Unterbrechungsbehandlung, absolute Adressierung, maschinenabhängige Speicherzuteilungsmethoden, fehlendes Laufzeitsystem,

usw. im Vordergrund des Interesses standen. Obwohl bekannt ist, daß Probleme dieser Art höchstens 5 - 10 % des gesamten Betriebssystems ausmachen, verdrängte doch die Beschäftigung mit diesen Fragen vielfach die Lösung von Problemen der Programmorganisation, der Sicherung der Integrität von Daten und Programmabläufen und der Bereitstellung höherer sprachlicher Ausdrucksmittel. Charakteristisch ist die Verwendung typfreier Sprachen, bei denen wie im Assembler der Typ der Datenobjekte nur implizit aus den ausgeführten Operationen hervorgeht. Ein anderes Beispiel liefert Burroughs' Implementierungssprache ESPOL /Bu72/, ein ALGOL60-Dialekt, in welchem erst kürzlich die gut ausgebauten E/A-Möglichkeiten des Systems in der auch dem Anwender verfügbaren Schreibweise zugänglich gemacht wurden; bis dahin wurde der Schluß gezogen, daß eine Sprache keine problemorientierte Schreibweise der E/A erlauben könne, wenn in ihr zugleich die Teile des Betriebssystems geschrieben sind, welche die Ein/Ausgabe implementieren.

Erfahrungen mit Burroughs-Systemen und mit PASCAL /Wi74/ legen schließlich den Versuch nahe, wesentlich mehr Zugriffsbeschränkungen bereits durch den Übersetzer kontrollieren zu lassen und dadurch die Laufzeit zu entlasten. Ein wesentliches Hilfsmittel bilden dabei die aus dem CLASS-Konzept von SIMULA /Da67/ hergeleiteten Strukturmoduln, die vor allem der Definition von Datenstrukturen samt der auf ihnen erklärten Operationen dienen.

Diese Überlegungen führten zu folgenden Entwicklungszielen für BALG:

- höhere Programmiersprache, in der zusätzlich niedere Spracheigenschaften in speziell gekennzeichneten Programmmoduln zugänglich sind;
- Verallgemeinerung des Modulbegriffs zu hierarchisch geschichteten, funktionalen Programmmoduln, welche sowohl Strukturmoduln als auch Prozeduren einschließen;

- Verallgemeinerung des Variablenbegriffs zu einem Objektbegriff, der neben der unbeschränkten Weitergabe der Zugriffsrechte auf das Objekt an andere Moduln auch eine eingeschränkte Weitergabe (z. B. nur Lesen erlaubt) gestattet;
- Explizite Kontrolle der Schnittstellen zwischen Moduln mit der Möglichkeit, den Zugriff einzuschränken;
- Kellerorientierte Speicherorganisation, welche in vom Programmierer gesteuerter Form verschiedene Formen der Freispeicherorganisation einschließt.

Zur Erreichung dieser Ziele gehen wir von den Datentypen PASCALs /Wi74/ aus. Die Ablaufsteuerung für sequentielle Abläufe sowie viele Schreibweisen sind ALGOL68 /Wi69/ entnommen. Die Steuerung paralleler Abläufe soll mit Hilfe von Monitoren /Ho74/ erfolgen; Prozesse und Ein/Ausgabe sollen mittels spezieller Strukturmoduln dargestellt werden. Diese Teile fehlen bisher noch; ebenso ist die Bereitstellung der notwendigen Informationen für die Speicherbereinigung durch den Übersetzer noch nicht ganz geklärt.

Wir gehen im folgenden nur auf einige wenige Spracheigenschaften ein, darunter die Situationsklausel, funktionelle Programmmoduln, die Grundideen der Speicherverwaltung und Schnittstellenbeschreibungen. Die Grundüberlegungen, die dem Objektbegriff zugrundeliegen, wurden in /Go74/ erklärt.

2. Situationsklauseln

Zu den Programmabläufen, die mangels besserer Ausdrucksmöglichkeiten in den meisten Programmiersprachen mit Sprunganweisungen umschrieben werden müssen, gehört die folgende: Gegeben sei ein Programmstück p , vorzüglich eine Schleife, durch dessen Ausführung man eine der "Situationen" s_1, \dots, s_n erreicht; die Ausführung einer situationsabhängigen Anweisung A_1 oder die Berechnung eines entsprechenden Ausdruckes beendet dann

die Ausführung von p und führt zum gewünschten Ergebnis. Das einfachste Beispiel eines derartigen Ablaufs ist die sequentielle Suche mit nachfolgendem Eintrag (zunächst in ALGOL68-Notation):

```
begin
  for i to n
    do if x = a [i] then index := i; goto gefunden fi
  od;
  index := n := n+1;
  a [index] := x
gefunden : index
end
```

Knuth /Ku74/ diskutiert zahlreiche Formulierungsmöglichkeiten für dieses Beispiel. Die eleganteste Formulierung liefert wohl die von Zahn /Za74/ eingeführte Situationsklausel (hier in etwas fortentwickelter Schreibweise):

```
until gefunden, nicht_gefunden:
  for i to n
    loop if x = a [i] then gefunden (i) fi
  repeat;
  nicht_gefunden
on gefunden (int; j) : index := j
on nicht_gefunden : begin index := n:=n+1;
                      a [index] := x;
                      index
                    end
end
```

Die "Situationsanweisungen" A_i , welche bei Eintreten einer der Situationen "gefunden" oder "nicht_gefunden" ausgeführt werden sollen, werden hier in prozedurähnlicher Schreibweise formuliert und können auch mit Parametern versehen sein. Aufgefaßt als Ausdruck liefert die Situationsklausel das Ergebnis einer der Anweisungen A_i als Gesamtergebnis.

Situationsklauseln können mit Vorteil auch zum Abfangen impliziter Fehlersituationen eingesetzt werden. BALG kennt eine Reihe von Standard-Fehlersituationen, darunter z.B. overflow und invalid_index, deren Eintreten implizit festgestellt wird (durch Hardware-Reaktion oder durch vom Übersetzer erzeugte Tests). Die Behandlung erfolgt jedoch explizit durch Situationsklauseln wie

```
until overflow:  
    Programm  
on overflow : A  
end.
```

Auch die Schachtelung, welche in verschiedenen Programmteilen unterschiedliche Fehlerbehandlungen erlaubt, ist möglich:

```
until overflow :  
    Hauptprogramm 1. Teil ;  
    until overflow :  
        Teilprogramm  
    on overflow : ATP  
    end;  
    Hauptprogramm 2. Teil  
on overflow : AHP  
end.
```

Schließlich lassen sich Prozeduren und andere Programmmoduln so parametrisieren, daß bei Eintreten einer Situation die an der Aufrufstelle gültige Situationsanweisung anstelle der an der Vereinbarungsstelle gültigen Situationsanweisung ausgeführt wird (und damit zugleich der Modul verlassen wird).

3. Funktionale Programmmoduln

Für den Begriff Programmmodul finden sich verschiedenartige Definitionen, z.B.

- Moduln als getrennt übersetzbare Programmeinheiten,
- Moduln als Programmteile mit "minimalen" Querbezügen,

- Moduln als logisch in sich abgeschlossene und zusammenhängende Programmeinheiten (funktionale Modularität)

Diese Definitionen betrachten entweder nur die äußere Form oder stellen auch inhaltliche Anforderungen. In BALG gehen wir von der dritten Möglichkeit, der funktionalen Modularität aus. Das schließt natürlich nicht aus, daß die resultierenden Moduln zugleich getrennt übersetzbar sind oder Schnittstellen geringen Umfangs besitzen (vgl. auch /Go74a/ für weitere Bemerkungen zum Modulbegriff).

Sprachlich betrachtet ist ein Programmmodul entweder eine Prozedur, welche einen einzelnen Algorithmus wiedergibt, oder eine Datenstruktur, eventuell zusammen mit weiteren Prozeduren, welche die Operationen mit dieser Datenstruktur definieren. Diese Klasse von Moduln nennen wir in BALG Strukturmoduln. Prozeduren in BALG haben die in ALGOL-ähnlichen Sprachen üblichen Eigenschaften; zusätzlich kann verlangt werden, daß eine Prozedur offen anstelle ihres Aufrufs eingebaut wird.

Als Beispiel geben wir die Definition eines Kellers an:

```

external situation: stack_overflow, stack_underflow;
                                (*externe Parameter*)
public proc (type) : push ; (*von außen zugängliche Operationen und Objekte*)

    proc : pop ;
    proc type : value ;
    const int : depth #

(*bis hierher geht die Schnittstellenbeschreibung. Jetzt folgt der Modul:*)

module : stack (int : max_depth) type =
    begin
        var array 1 : max_depth type: a
            & int : depth ;
        proc : push (const type : x) =

```

```

    if depth  $\geq$  max_depth then stack overflow
    else depth := 1 ; a [depth] := x fi;
proc : pop =
    if depth  $\geq$  1 then depth :- 1
    else stack_underflow fi;
proc : value type =
    if depth  $\geq$  1 then a [depth],
    else stack_underflow fi;

```

initialize:

```

    depth := 0
    end (*module stack*)

```

Aufrufe von Strukturmoduln sind nur in Form von Strukturvereinbarungen möglich, z.B.

```

    struct stack (27) real : s

```

Strukturvereinbarungen rufen den angegebenen Strukturmodul auf und hinterlassen die Parameter und die im Rumpf vereinbarten lokalen Prozeduren und Datenobjekte als Datenstruktur. Diese wird durch den definierten Bezeichner benannt und existiert bis zum Verlassen des Blocks, der die Strukturvereinbarung enthält. In diesem Zeitraum können Operationen mit der Datenstruktur ausgeführt werden.

Strukturmoduln können mit einem Datentyp parametrisiert werden; im vorliegenden Beispiel ist "type" ein formaler Datentyp, im Aufruf ersetzt durch den aktuellen Typ real. Mit dieser Parametrisierung wird dem Bedürfnis nach "Struktur-Konstruktoren" Rechnung getragen, das sind Strukturmoduln, deren Aufbau weitgehend unabhängig ist vom Typ der Elemente der resultierenden Datenstrukturen. Der Elementtyp wird daher erst bei der aktuellen Bildung der Struktur ergänzt. Wie aus dem Beispiel ersichtlich, kann man in einem Modul Objekte eines formalen Datentyps bilden, zuweisen oder als Prozedurergebnisse abgeben; weitere Operationen sind nicht zulässig.

Monitore im Sinne von /Ho74/ sind Strukturen in einer nicht-

sequentiellen Programmumgebung. Sie lassen zu jeder Zeit nur die Ausführung einer Operation zu. Auch andere Typen von Programmmoduln wie etwa Koroutinen lassen sich auf Strukturmoduln zurückführen.

4. Speicherorganisation

Bei der Konzeption von BALG wurde davon ausgegangen, daß nur in beschränktem Maße Speicher absolut adressiert zugeteilt wird. Überwiegend soll Speicher im Rahmen einer einheitlichen Kellerorganisation vergeben werden. Dieses Prinzip sollte auch aufrechterhalten werden, wenn für kollaterale Prozesse oder für haldenartige Organisationen Speicher bereitgestellt werden soll. Die Begrenzung der Lebensdauer von Strukturen auf den Block, in dem sie geschaffen wurden, und die Festlegung, daß Strukturen nur durch Vereinbarungen geschaffen werden, ist Ausfluß des Kellerprinzips.

Um haldenähnliche Organisationsformen im Rahmen der Kellerorganisation bereitzustellen, fassen wir die Halde als eine Datenstruktur auf, die mit Hilfe eines entsprechenden Strukturmoduls gebildet wird. (Es könnte mehrere Halden geben!) Globale Generatoren im Sinne von ALGOL68 werden zusätzlich mit der Benennung der Halde gekennzeichnet, von der der Speicher bezogen wird. Diese bestimmt auch die Lebensdauer des generierten Objekts. Auch der Typ von Verweisen auf Objekte in einer solchen Halde wird zusätzlich mit der Haldenbenennung gekennzeichnet. Diese Regelung erlaubt bei Verweisen eine sehr einfache Prüfung der Lebensdauerregeln zur Übersetzungszeit statt - wie in ALGOL 68 vielfach nötig - zur Laufzeit. Die Kennzeichnung globaler Generatoren mit der entsprechenden Halde erlaubt es, den Generator zur Übersetzungszeit umzusetzen in einen Prozeduraufruf einer Speicherzuteilungsprozedur allocate, welche der Haldenmodul zur Verfügung stellen muß. Der Aufruf wird intern mit der Anzahl von Speichereinheiten (Zellen) parametrisiert, welche für das unterzubringende Objekt benötigt werden. Das Schema eines Haldenmoduls ist demnach:

```

public proc (int) ref cell : allocate #
unsafe module : haldenmodul (int : länge) =
begin
  var array [1:länge] cell Haldenspeicher ; (*der eigent-
                                             liche Haldenspeicher*)
  proc : allocate (int : umfang) ref cell =
    (*bringe <umfang> Zellen im Speicher, vorzugs-
     weise im Haldenspeicher, unter. Das Ergebnis
     des Typs ref cell wird extern als Referenz auf-
     gefaßt.*);
    .
    .
    .
end

```

Der Aufruf des Haldenspeichers kann durch

```
struct Haldenmodul (n) : Halde
```

erfolgen. Danach ist im Kontext

```
var ref halde int:xx
```

die Zuweisung

```
xx := new halde int
```

mit der Bedeutung

```
xx := halde.allocate (<Länge von int>)
```

möglich. Wie ersichtlich ist der Modul Haldenspeicher primär für die Speicherzuteilungsprozedur zuständig. Den benötigten Speicher kann er aus einer eigenen einstufigen Reihung beziehen, er kann aber die Objekte auch in anderen Speicherbereichen unterbringen, zu denen er Zugang besitzt.

Das Verfahren demonstriert ferner eine Spracheigenschaft, die ich als semantische Erweiterbarkeit bezeichnen möchte: Für Sprachkonzepte wie globale Generatoren und Referenzen ist der syntaktische Rahmen vorgegeben. Die semantische Interpretation dieser syntaktischen Konzepte wird durch die Prozedur allocate und deren Ergebnis festgelegt und wird damit durch den Programmierer, nicht durch die Sprache, bestimmt. Dabei müssen aller-

dings gewisse, eventuell sogar implementierungsbedingte Beschränkungen eingehalten werden; z.B. muß der Inhaltsoperator cont auf die resultierenden Referenzen anwendbar sein.

5. Schnittstellenbeschreibungen

Das Prinzip üblicher blockstrukturierter Sprachen, den Gültigkeitsbereich einer Größe, die in einem äußeren Block vereinbart ist, automatisch auf innenliegende Blöcke zu erweitern, sofern dort keine gleichbenannte Größe definiert ist, wird in BALG bei der Schachtelung von Moduln nicht angewandt: Globale Größen sind in innenliegenden Moduln im allgemeinen unzugänglich. Sie werden nur zugänglich, wenn sie explizit als externe Größen in einer Schnittstellenbeschreibung für den Modul aufgeführt sind. Umgekehrt können Bezeichner lokaler Größen eines Strukturmoduls nur dann außerhalb des Moduls als Selektoren zusammen mit der Benennung einer Struktur benutzt werden, wenn sie als Eingänge des Moduls in der Schnittstellenbeschreibung genannt sind. Die Forderung nach expliziter Kennzeichnung der Eingänge und externen Größen ist aus dem Prinzip des "information hiding" von Parnas abgeleitet: In der Kommunikation zwischen einem Modul und seiner Umgebung sollen nur Größen vorkommen, die relevant für die Funktion des Moduls sind; Größen, die implementierungsbezogene Einzelheiten des Moduls oder seiner Umgebung betreffen, sollen nicht zur Schnittstelle gehören. Daher taucht die Reihung `a` unseres Moduls `stack` nicht in der Schnittstelle auf. Die Zugriffsbeschränkung für externe Größen ist außerdem nützlich, weil der Übergang Modulumgebung → Modul häufig den Übergang zwischen einer tieferliegenden und einer höherliegenden Schicht in einer Programmhierarchie entspricht. Die Zugriffsbeschränkung ist dann Ausdruck der Abschirmung zwischen den Programmschichten.

Aus Gründen der Ausgewogenheit (eine Schnittstellenbeschreibung soll nicht länger als der zugehörige Modul sein) kann bei Moduln, die im Kontext ihrer Umgebung übersetzt werden, die Spezifikation der externen Größen weggelassen werden. Davon haben wir bei den Prozeduren unseres Moduls `stack` Ge-

brauch gemacht. Hingegen muß bei getrennter Übersetzung die Schnittstellenbeschreibung sogar zweimal angegeben sein: Im Text der Umgebung des Moduls werden die externen Größen aufgeführt, welche dem Modul zur Verfügung gestellt werden, und die Eingänge, welche in der Umgebung benötigt werden. Als Vorspann des Moduls erscheinen die externen Größen, welche der Modul benötigt, und die Eingänge, welche der Modul bereitstellt. Diese Doppelspezifikation erlaubt dem Übersetzer die Prüfung der Konsistenz der beiden Angaben (benötigte Größen bilden Teilmenge der bereitgestellten Größen). Dies erscheint angemessen, da die Modulschnittstelle im Programmiererteam häufig auch den Wechsel der Zuständigkeit der Programmierer kennzeichnet. Der Übersetzer prüft also zugleich die richtige Verzahnung der Produkte mehrerer Programmierer, zumindest was Bezeichnung und Typ der Größen angeht.

Schließlich kann durch eine Schnittstellenbeschreibung auch das Zugriffsrecht auf Variable eingeschränkt werden. In unserem Modul stack ist beispielsweise die Tiefe des Kellers von außen nur als unveränderliche Größe zugänglich, während sie im Modul natürlich eine Variable darstellt. Bei zusammengesetzten Objekten wie zum Beispiel Verbunden lassen sich die Zugriffsrechte "variabel", "unveränderlich", "unzugänglich" nicht nur für das Objekt als Ganzes, sondern sogar für die Glieder einzeln bestimmen.

6. Zusammenfassung

Wir haben in dieser Arbeit einige Eigenschaften der in Entwicklung befindlichen Programmiersprache BALG dargestellt. Bevor die noch ausstehenden Ergänzungen und Glättungen vorgenommen werden, soll die Sprache - überwiegend mit Hilfe von Diplomarbeiten - implementiert werden.

Für zahlreiche Diskussionen und wertvolle Hinweise bei der Entwicklung von BALG danke ich den Herren U. Kastens, H. Neugebauer, H. Rohlfing und H. Santo.

Literatur

- /Bu72/ Burroughs Corporation Burroughs B6700 ESPOL Language Information Manual
Burroughs Corporation Form 5000094, Detroit, 1972
- /Cl74/ Clark, B.L., Ham, F.J.B. The project SUE System Language Reference Manual.
Computer Systems research group, University of Toronto, report no. 42, 1974
- /Da67/ Dahl, O.J., Myhrhaug, B., Nygaard, U.
SIMULA 67, Common Base Language
Norwegian Computing Center, Oslo, 1967
- /Go70/ Goos, G., Lagally, K., Sapper, G.
PS440 - Eine niedere Programmiersprache
Rechenzentrum der Technischen Hochschule München
Bericht 7002, 1970
- /Go74a/ Goos, G.
Systemprogrammiersprachen und Strukturiertes Programmieren
LNCS 23, 203 - 224, 1974
- /Go74b/ Goos, G.
Some Thoughts on Variables
Fakultät für Informatik, Universität Karlsruhe
Bericht 19/74, 1974
- /Go75/ Goos, G.
Die Programmiersprache BALG - vorläufige Fassung
Fakultät für Informatik, Universität Karlsruhe
Bericht 6/75, 1975

- /Ho74/ Hoare, C.A.R. Monitors: An Operating System Structuring Concept
Comm. ACM 17, 549 - 557 (1974)
- /Kn74/ Knuth, D.E. Structured Programming with goto Statements
Computing Surveys 6, 261 - 302, 1974
- /Sa71/ Sammet, H.E. A brief survey of languages used in systems implementation
SIGPLAN Notices 6, no. 9, 1971
- /Wi68/ Wirth, N. PL360: A Programming Language for the 360 Computers
Journal ACM 15, 37 - 74, 1968
- /Wi74/ Wirth, N. The Programming Language PASCAL (Revised Report)
LNCS 18, Springer: Berlin-Heidelberg-New York, 1974
- /Wi69/ Wijngaarden, A.v. (ed.) Report on the Algorithmic Language ALGOL 68
Num. Math. 14, 79 - 218, 1969
- /Za74/ Zahn, C.T. A Control Statement for natural top-down structured programming LNCS 19, 170 - 180. Springer: Berlin-Heidelberg-New York, 1974

eingegangen am: 23. 4. 1976

Anschrift des Verfassers:

Prof. Dr. G. Goos
Fakultät für Informatik
Universität Karlsruhe
D 75 Karlsruhe 1
Postfach 6380

Programmierung von Dialogcompilern

1. Einführung

Im Zusammenhang mit der Programmierung eines Dialogcompilers für das Mehrfachzugriffssystem MS /1/, /2/ traten einige Fragen bezüglich der effektiven Gestaltung der Dialogarbeit auf. Im besonderen galt es zu klären, welche Compilerstruktur dem Dialogbetrieb am angemessensten ist. Dabei galt es einige wichtige Nebenbedingungen zu beachten, von denen die Generierung eines effektiven Objektcodes und ein geringer Programmieraufwand als die wichtigsten anzusehen sind. In diesem Beitrag wird von den üblicherweise an den Dialogbetrieb zu stellenden Nutzerforderungen ausgegangen. Darauf aufbauend werden mögliche Realisierungen von Dialogcompilern betrachtet. Im Ergebnis der Untersuchungen zeigt es sich, daß die Generierung eines effektiven Objektcodes nicht den übrigen Forderungen an einen Dialogcompiler widersprechen muß und daß auch der zusätzliche Aufwand zur Realisierung der Dialogmöglichkeiten in Grenzen gehalten werden kann.

2. Allgemeines über Aufgaben und Struktur von Compilern

Zunächst sind einige allgemeine Bemerkungen über Aufgaben und Struktur höherer Programmiersprachen und über ihre Compiler notwendig. Es sollen Hilfsmittel bereitgestellt werden, mit denen sich die Struktur von Dialogcompilern gut charakterisieren und darstellen läßt.

Programme höherer Programmiersprachen beschreiben letztlich die Manipulation von Daten. Zu diesem Zweck werden in den Programmiersprachen Größen definiert, über die man sich auf die Daten bezieht bzw. die zur Steuerung des Programmablaufs dienen. Durch die Definition werden den Größen Angaben zugeordnet, die bestimmen, welche Art von Daten bzw. Informationen zur Programmsteuerung zu den Größen gehören. Ein konkreter Compiler ergänzt diese Angaben (z.B. Speicheradresse, Speicher-

platzbedarf), so daß in den Objektprogrammen die den Größen zugeordneten Daten (Informationen) verarbeitet werden können (z.T. werden diese Angaben auch erst zur Laufzeit vervollständigt). Diese eben beschriebenen, den Größen zugeordneten Angaben, sollen im folgenden Verarbeitungsattribute heißen. Unter Verarbeitungsattributen einer Programmgröße sind also alle diejenigen Angaben zu verstehen, die erforderlich sind, um die den Größen zugeordneten Daten bzw. Informationen zur Steuerung des Programmablaufs im Sinne des Programms in der "richtigen Weise" auf einem konkreten Automaten benutzen zu können.

Der Begriff Verarbeitungsattribut spielt eine wesentliche Rolle im Zusammenhang mit Korrekturen an einem Programm. Erfordert jede Korrektur eine Neuübersetzung des korrigierten Programms, so wird der Prozeß der Dialogprogrammierung äußerst ineffektiv. Es ist anzustreben, daß nur eine gewisse Umgebung der Korrektur neu übersetzt wird und der Objektcode für die übrigen Programmteile gültig bleibt, d.h., es dürfen die Verarbeitungsattribute außerhalb dieser Umgebung nicht verändert werden bzw. der Objektcode darf nicht von eventuell veränderten Verarbeitungsattributen abhängen.

Bestimmte Verarbeitungsattribute von Programmgrößen sind lokaler Natur. Sie können schon bei der lexikalischen Analyse erkannt werden (z.B. Typ von Konstanten). Sie ändern sich also nicht bei Korrekturen an anderen Programmteilen. Bei der Realisierung eines Korrekturkonzepts können sie also im wesentlichen unbeachtet bleiben. Das gleiche gilt für diejenigen Verarbeitungsattribute, die normalerweise erst zur Laufzeit bestimmt werden, wie z.B. Speicherplatz und Adresse dynamischer Felder.

Die Aufgaben von Compilern lassen sich aus unterschiedlichen Blickrichtungen systematisieren. Für die weiteren Betrachtungen ist es günstig, die folgenden Aufgaben zu unterscheiden:

1. Syntaxanalyse (S)

Hierbei geht es um die Analyse der in BNF-Notation beschreib-

baren Programmstruktur.

2. Bestimmung von Verarbeitungsattributen (V)

Hierzu wäre die Auswertung der Definitionen von Programmgrößen und Festlegungen über den Speicherplatz usw. zu zählen.

3. Codegenerierung (C)

Normalerweise wird es sich um Maschinencode handeln. Der erzeugte Code kann aber durchaus auch zu interpretierende Elemente enthalten.

4. Laufzeitsystem (L)

Der Compiler muß im Zusammenhang mit der Programmabarbeitung gesehen werden. Forderungen an das Laufzeitsystem wirken auf die ersten drei Punkte zurück.

Die hier hervorgehobenen Aufgaben werden z.T. in selbständigen Compilerpässen bearbeitet. Im Normalfall erledigen aber die einzelnen Pässe Beiträge zu verschiedenen Aufgaben des Compilers.

Bild 1 zeigt die Struktur eines 2 Paß Compilers für den Stapelverarbeitungsbetrieb. Der ↓ Pfeil zeigt die zeitliche Reihenfolge der Pässe an. S und C können jeweils einem Paß zugeordnet werden. der → Pfeil weist auf einen Beitrag zu einer Aufgabe hin. Hier erfolgt sowohl bei der Syntaxanalyse, bei der Codegenerierung und zur Laufzeit eine Festlegung von Verarbeitungsattributen.

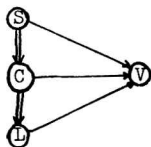


Bild 1

Struktur eines 2 Paß
Compilers für den
Stapelverarbeitungsbetrieb

Bild 2 zeigt die typische Struktur von Interpretern für die Dialogprogrammierung. Die Syntaxanalyse erfolgt lokal für die eingegebenen Programmstücke. Eine Einordnung in den Programm-

zusammenhang erfolgt nicht. Eventuelle Generierung von Code-
stücken, Bestimmung der Verarbeitungsattribute und Laufzeit-
system sind als Einheit zu betrachten.

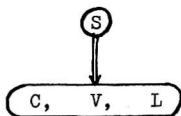


Bild 2

Struktur eines Interpreters
für die Dialogprogrammierung

Verwendet man die übliche Struktur der sehr effektiven batch-
Compiler für die Dialogprogrammierung, so erfordern die rela-
tiv häufig auftretenden Programmänderungen jeweils eine Neu-
übersetzung des vollständigen Programms.

Bei der Verwendung eines Interpreters mit der im Bild 2 ange-
gebenen Struktur läßt sich eine Änderung sehr leicht reali-
sieren. Dagegen wird die Programmabarbeitung uneffektiv. Zwi-
schen diesen beiden Compilermodellen ist ein Kompromiß zu
suchen. Um zwischen den verschiedenen in der Literatur bereits
beschriebenen Formen von Dialogcompilern eine günstige Form
auswählen zu können, sollen zunächst wichtige Forderungen an
Dialogcompiler zusammengestellt werden. Danach erfolgt im Ab-
schnitt 5 eine Diskussion dreier charakteristischer Varianten.

3. Kriterien für den Entwurf von Dialogcompilern

Dialogcompiler sollen den Programmwurf und die Programmverifi-
kation möglichst weitgehend unterstützen. Es geht dabei so-
wohl um Hinweise auf syntaktische Fehler in bereits fertigen
Programmstücken, um Änderungen im Programm und die Programm-
testung im Dialog.

Auf das Problem der Antwortzeit nach einer erfolgten Eingabe
durch den Nutzer wird schon in /3/ hingewiesen.

Es gibt Arbeitsperioden des Programmierers, in denen eine Un-
terbrechung äußerst ungünstig ist, z.B. Formulierung eines
komplizierten Programmteils (ein Stück des Programmteils exi-
stiert nur im Kopf des Programmierers). In derartigen Arbeits-

perioden ist eine kurze Antwortzeit zu gewährleisten. Nach Abschluß derartiger Arbeitsperioden stören etwas längere Wartezeiten nicht.

Der Programmierer sollte möglichst viel Information auf seine Eingaben (Anfragen) erhalten. Das kann bei der Programmeingabe durch umfangreiche syntaktische und semantische Kontrollen erreicht werden oder beim Programmtest durch gezielte Bereitstellung von Information über den Fortgang der Abarbeitung. Die Effektivität des Dialogcompilers ist weiterhin von großem Interesse. Effektivitätssteigerungen sollten durch konzeptionelle Betrachtungen erreicht werden und nicht zu Lasten des Nutzers gehen, denn für ihn soll der Dialogcompiler ein gutes Hilfsmittel sein.

Es ergeben sich also drei wesentliche Forderungen an einen Dialogcompiler:

1. Günstige Antwortzeit
2. Große Information
3. Effektivität des Dialogcompilers

Aus diesen Forderungen können weitere Schlußfolgerungen gezogen werden:

- Es ist günstig, wenn der Compiler bei Eingabe einer Programmzeile soweit wie möglich die Syntax kontrolliert und auch Hinweise auf semantische Fehler gibt.
- Nach Abschluß der Programmeingabe bzw. der Korrektur eines Programms stören etwas größere Wartezeiten nicht. Hier können Vorbereitungen (z.B. Objektcodegenerierung) für eine schnelle Antwort auf weitere Anfragen (z.B. Testanweisungen) durchgeführt werden.
- Die Effektivität des Dialogcompilers verlangt die Lokalisierung des Einflusses der Korrekturen.
- Die Einschränkung auf lokale Information bei der Eingabe einer Programmzeile (keine Beachtung anderer schon eingegebener Programmteile) können eine weitere Effektivitätsverbesserung bringen.

4. Syntax- und Semantiktest bei der Programmeingabe

Die Forderung nach umfangreicher Information für den Programmierer und die Beschränkung des Syntax- und Semantiktests auf eine eingegebene Programmzeile widersprechen einander. Beide Wege sind aber gangbar. Im zweiten Fall kann die globale Information beim Programmtest nachgeliefert werden. Hier ist der Programmierer wieder aufnahmefähig für diese Information. Andererseits lenkt aber derartige Information von der Programmlogik ab, die es beim Testen in erster Linie zu kontrollieren gilt.

Ein globaler Fehlertest bei der Programmeingabe führt zu einer Compilerstruktur, wie sie im Bild 1 dargestellt ist. Die Bestimmung der Verarbeitungsattribute ist soweit wie möglich vorzuziehen, um auch semantische Fehler erkennen zu können. Es gilt zu prüfen, ob mit vertretbaren Effektivitätsverlusten diese Compilerstruktur auch für den Dialogbetrieb nutzbar gemacht werden kann. Vor allem wird es um eine Lokalisierung des Einflusses von Korrekturen gehen.

5. Realisierung von Dialogcompilern, Lokalisierung des Einflusses von Korrekturen

5.1. Dynamische Bestimmung von Verarbeitungsattributen

Will man die Korrektur an Programmen möglichst einfach gestalten, so bietet sich die im Bild 3 dargestellte Compilerstruktur an.

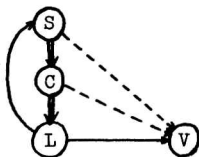


Bild 3

Dialogcompiler mit dynamischer Bestimmung der Verarbeitungsattribute (Die gestrichelten Pfeile deuten an, daß von C und S nur auf Grund lokaler Kontrollen Beiträge zu V erfolgen.)

Alle Verarbeitungsattribute, die nicht als lokal anzusehen sind, werden zur Laufzeit bestimmt. Gleichzeitig erfolgt zur Laufzeit eine Kontrolle der globalen Syntax.

Da in C nicht auf Verarbeitungsattribute Bezug genommen wird,

die außerhalb der bearbeiteten Zeile definiert worden sind, bleibt der Code der Zeile unverändert gültig, bis die Zeile selbst korrigiert wird. Nach einer Korrektur sind damit immer nur die veränderten Programmzeilen zu bearbeiten, eine sehr leichte und mit geringem Aufwand realisierbare Aufgabe. Zur Laufzeit ist eine dynamische Bereitstellung der nicht lokalen Verarbeitungsattribute erforderlich. Weiter muß der Programmablauf interpretiert werden. Bei der Programmiersprache BASIC sind viele Verarbeitungsattribute lokal (Variable, Felder, Funktionen können eindeutig erkannt werden. Es gibt nur einen Datentyp). Hier lassen sich diese Aufgaben relativ effektiv realisieren /4/. Da es in BASIC kein Blockkonzept gibt, ist es auch denkbar, die Verarbeitungsattribute vor Programmabarbeitung einmalig zu bestimmen und in dafür vorgesehenen Speicherplätzen bereitzustellen (im wesentlichen Adressen von Größen), um so die Programmabarbeitung insgesamt noch effektiver zu gestalten, wie es im Abschnitt 9.4 der Arbeit /5/ vorgeschlagen wird.

In Programmiersprachen wie ALGOL 60 ist die Realisierung des Laufzeitsystems nicht so effektiv möglich. Das Blockkonzept erfordert zur Laufzeit den Aufbau einer Identifizierliste, wie sie sonst bei Übersetzern zur Zeit der Codegenerierung erforderlich ist. Weiteren Aufwand bereitet die Kontrolle des Programmablaufs, insbesondere die Bestimmung der Adressen von Sprungzielen. Die Abarbeitung ist etwa so schnell wie die Codegenerierung im Übersetzer für die Stapelverarbeitung /6/.

5.2. Objektcodegenerierung so spät wie möglich

Die im vorigen Abschnitt skizzierte Compilerstruktur wird dann besonders ineffektiv, wenn Programmteile häufig abzuarbeiten sind, was auch im Testbetrieb der Fall sein kann.

Es liegt daher nahe, zwar weiterhin dynamisch die nichtlokalen Verarbeitungsattribute zur Laufzeit zu bestimmen und dynamisch den Programmablauf zu kontrollieren, aber vor der ersten Abarbeitung einer Zeile abarbeitsfähigen Maschinencode zu generieren, wie es in /6/ von H. Schmid vorgeschlagen wird. Bild 4

zeigt die entsprechende Compilerstruktur.

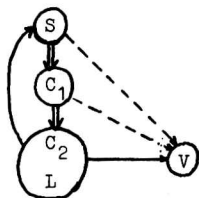


Bild 4

Objektcodegenerierung
zur Laufzeit

Nach Korrekturen müßte man entweder den bis dahin erzeugten Maschinencode als ungültig ansehen oder den Einfluß der Korrektur auf die Verarbeitungsattribute lokalisieren und nur den betroffenen Code streichen. Die Lokalisierung des Einflusses einer Korrektur wäre wie folgt möglich:

- Zusammenstellen aller Größen, deren Verarbeitungsattribute verändert wurden
- Zusammenstellen aller in einer Zeile benutzten Größen
- Falls eine Zeile abgearbeitet werden soll, prüfen, ob die Verarbeitungsattribute einer Größe in der Zeile verändert wurden und eventuelle erneute Codegenerierung.

Bei dieser Form der Dialogcompiler ist die Laufzeit noch relativ stark durch die dynamische Attributbestimmung und Kontrolle des Programmablaufs belastet. Die Lokalisierung der Korrekturen erfordert schon nicht zu vernachlässigenden Aufwand. Nachteilig für den Dialog während der Testphase wirkt sich weiter aus, daß immer wieder die Programme zur Codegenerierung benötigt werden. Das führt zu einer stärkeren Belastung der Arbeit mit den externen Geräten.

5.3. Frühe Objektcodegenerierung mit Lokalisierung des Einflusses von Korrekturen

Um die Laufzeiteffektivität zu verbessern, bietet es sich an, schon nach Abschluß einer Programmeingabe bzw. nach erfolgter Korrektur ein vollständiges Objektprogramm bereitzustellen. Den Nutzer würde an dieser Stelle eine etwas größere Wartezeit

nicht stören. Weitere Fehler könnten so schon vor der Programm-
abarbeitung erkannt und angezeigt werden. Als Compilerstruktur
bietet sich die im Bild 1 festgehaltene Form an, entsprechend
der schon bei der Programmeingabe umfangreiche Fehlerkontrollen
möglich sind. Dieses Compilermodell wurde dem Entwurf des
MS-Algolcompilers zugrunde gelegt. Um der Forderung einer hohen
Effektivität gerecht zu werden, muß eine einfache und möglichst
gute Lokalisierung des Einflusses von Korrekturen möglich sein.
Daß dies realisierbar ist, gilt es im folgenden aufzuzeigen
und später durch Messungen am MS-Algol zu belegen.

Im folgenden wird von ALGOL 60 ausgegangen. Unter einer Korrektur-
umgebung soll zunächst ein zusammenhängendes Programmstück
verstanden werden, das alle Korrekturen enthält. Alle Korrek-
turen in eine Umgebung einzuschließen, bedeutet für die weite-
ren Betrachtungen keine wesentliche Einschränkung, da zwei
"weit voneinander entfernte" Korrekturen als zwei selbständige
Korrekturen behandelt werden könnten.

Um den Begriff Korrekturumgebung weiter präzisieren zu können,
gilt es festzustellen, welche syntaktischen Strukturen man zur
Bildung einer Korrekturumgebung zulassen sollte. Dabei sind
folgende Bedingungen zu beachten:

1. Die Sprachelemente müssen hinreichend allgemein gewählt
werden, so daß sich aus ihnen immer eine Korrekturumgebung
bilden läßt.
2. Beginn und Ende der Korrekturumgebung müssen sich leicht
lokalisieren lassen.
3. Die Korrekturumgebung sollte ein möglichst kleines Programm-
stück sein.
4. Die selbständige Bearbeitung der Korrekturumgebung durch
den Compiler sollte leicht realisierbar sein.

Man könnte daran denken, gleichzeitig unterschiedliche Sprach-
elemente zur Bildung von Korrekturumgebungen zuzulassen
(Ausdrücke, Anweisungen und Vereinbarungen). Dadurch würde man
sicherlich der ersten und dritten Forderung genügen. Das Be-

stimmen der Korrekturumgebung und deren Bearbeitung würde dagegen Fallunterscheidungen erfordern. Die Bearbeitung müßte für jeden dieser Fälle realisiert werden.

Es liegt nahe, sich bei der Bildung von Korrekturumgebungen auf lückenlose Folgen von konstituenten Anweisungen eines Blocks oder Verbunds zu beschränken. Die folgenden beiden Bemerkungen machen diesen Sachverhalt deutlich.

Bemerkung 1:

Wird in einem Programm eine lückenlose Folge von Anweisungen durch eine andere Folge von Anweisungen ersetzt, so bleiben Typ, Art und Modus aller außerhalb der ersetzten Anweisungsfolge definierten Größen unverändert. Von den Größen, die in der eingesetzten Anweisungsfolge definiert werden, können außerhalb nur Marken benutzt werden.

Bemerkung 2:

Eine lückenlose Folge von konstituenten Anweisungen eines Blocks oder Verbunds läßt sich durch eine beliebige andere Anweisungsfolge ersetzen, ohne daß dadurch gegen die kontextfreie Syntax von ALGOL 60 verstoßen wird.

Bemerkung 1 und 2 zeigen, daß die selbständige Bearbeitung dieser Anweisungsfolgen und ihre Lokalisierung relativ einfach ist. Würde man als Korrekturumgebung beliebige Anweisungsfolgen zulassen, so müßten zusätzlich noch zwei Fälle bezüglich der Korrekturmöglichkeiten beachtet werden. Nach do und else und nach einem Prozedurkopf kann eine Anweisung durch eine beliebige andere ersetzt werden, nach then eine unbedingte Anweisung durch eine andere unbedingte Anweisung.

Diese Überlegungen führen zu einem Korrekturkonzept auf der Grundlage folgender Korrekturumgebungen.

Eine Korrekturumgebung ist ein zusammenhängendes Programmstück, das die folgenden Bedingungen erfüllt:

1. Das Programmstück enthält alle Korrekturen.
2. Vor der Korrektur ist das Programmstück eine konstituente

Folge von Anweisungen eines Blocks oder Verbunds.

3. Nach der Korrektur entsteht aus dem Programmstück wieder eine Folge von Anweisungen.
4. Die Korrekturumgebung ist das kleinste Programmstück, das diese Bedingungen erfüllt.

Die dritte Bedingung muß explizit gefordert werden, da es nicht selbstverständlich ist, daß nach den Korrekturen wieder eine Anweisungsfolge vorliegt. Sind in einer Korrektur syntaktische Fehler enthalten, so existiert nach dieser Definition keine Korrekturumgebung. Es wäre das ganze Programm neu zu bearbeiten, wenn es dem Compiler nicht gelingt, diese Fehler so zu behandeln, daß nach ihrem Auftreten sinnvoll weitergearbeitet werden kann.

Eine Korrektur kann bei einer derartigen Wahl von Korrekturumgebungen in folgenden Schritten bearbeitet werden:

1. Es wird die erste Anweisung der Korrekturumgebung bestimmt.
2. Es wird der Zustand des Syntaxanalysepasses hergestellt, den er vor Verarbeitung der ersten Anweisung der Korrekturumgebung haben muß.
3. Nach Analyse der letzten Anweisung der Korrekturumgebung wird der Syntaxanalysepaß beendet. Die übrigen Anweisungen müssen nicht syntaktisch kontrolliert werden.
4. Der Paß zur Codegenerierung wird gestartet und der Zustand hergestellt, den er vor Verarbeitung der ersten Anweisung der Korrekturumgebung haben muß.
5. Nach der Codegenerierung für die letzte Anweisung der Korrekturumgebung wird die Codegenerierung abgeschlossen. Der Code der übrigen Anweisungen bleibt unverändert.
6. Nach der Codegenerierung werden die Sprungadressen entsprechend einer bei der Codegenerierung zusammengestellten bzw. korrigierten Liste in den gesamten Objektcode eingetragen.

Bei diesem Korrekturkonzept kann eine beim Testen unterbrochene Programmabarbeitung außerhalb von in der Korrekturumgebung enthaltenen Blöcken fortgesetzt werden.

Der erste Schritt läßt sich relativ leicht realisieren, da die Teile der Korrekturumgebung, die vor der ersten Korrektur liegen, nicht neu bearbeitet werden müssen.

Von den sechs Schritten sind 2. und 3. am kompliziertesten zu realisieren. Die Identifizierliste (Hashliste mit einem Überlaufbereich) wird sequentiell abgespeichert. In einem Programmvektor wird festgehalten, wo die Eintragungen in der Identifizierliste, die zu einer bestimmten Zeile gehören, beginnen. So wird es möglich, den Inhalt der Identifizierliste zu rekonstruieren, den sie am Anfang einer Zeile hatte. Nach Bearbeitung der Korrekturumgebung bleibt das Problem eventueller Sprünge in die Korrekturumgebung zu lösen. Bei dem ersten Auftreten einer Marke wird eine explizite Eintragung vorgenommen, wenn es sich um eine Vereinbarung handelt, sonst eine implizite. An einem Blockende werden alle impliziten Markeneintragungen mit entsprechenden expliziten bzw. impliziten Eintragungen des umfassenden Blocks verkettet, falls derartige Eintragungen vorhanden sind, sonst an neu zu erzeugende implizite Eintragungen für die entsprechenden Marken. Weiter wird jedes Auftreten einer Marke bei Erzeugung des Zwischencodes durch einen Verweis zu der expliziten bzw. impliziten Eintragung des gültigen Blocks ersetzt und bleibt damit unabhängig von der Position der Markenvereinbarung, was die Voraussetzung für 5. und 6. ist.

Möglichkeiten, die Korrekturumgebungen weiter einzuengen, sind z.B. in /7/ beschrieben. Bei dem vorgestellten Konzept sind

1. weniger Informationen erforderlich.
2. Der Compiler arbeitet bei der ersten Übersetzung eines Programms genauso wie bei der Bearbeitung einer Korrekturumgebung.
3. Die Objektprogramme sind bis auf die Behandlung von Marken nicht vom Korrekturkonzept beeinflusst.
4. Nach einer Programmunterbrechung und einer Korrektur kann außerhalb von korrigierten Blöcken wie üblich weitergearbeitet werden.
5. Der Aufwand zur Übersetzung einer größeren Korrekturumge-

bung wird durch die sequentielle Arbeitsweise des Übersetzers in vielen Fällen kompensiert werden.

Die Ausführungen zeigen, daß sich mit geringem Aufwand und ohne wesentliche Abweichungen von der Struktur eines sequentiell arbeitenden Übersetzers ein effektiv arbeitender Dialogcompiler programmieren läßt, der schnellen Objektcode erzeugt.

Zusammenfassung

Es werden Hilfsmittel zur Beschreibung der Struktur von Dialogcompilern bereitgestellt. Nach Angabe der wichtigsten Forderungen an Dialogcompiler erfolgt eine Diskussion von drei charakteristischen Modellen für Dialogcompiler. Dabei zeigt es sich, daß es möglich ist, ausgehend von der Struktur der sehr effektiven batch-Compiler mit wenig Aufwand einen effektiv arbeitenden Dialogcompiler mit teilweise Neuübersetzung nach Korrekturen zu programmieren. Die vorgeschlagene Variante gestattet eine Fortsetzung der Testarbeiten nach einer Korrektur.

Literatur:

- /1/ Kleffe, P.; Kosciolowicz, R.
Probleme der Objektcodegenerierung
und Abarbeitung für MS-Algol unter den
Bedingungen des virtuellen Speichers
und des Dialogs
Dissertation, Humboldt-Universität
zu Berlin, 1975
- /2/ Grafik, W.
Syntaxanalyse in einem Paß für ALGOL-60
unter den Bedingungen des Dialogbetriebes
und der on-line Korrektur
Dissertation, Humboldt-Universität zu
Berlin, 1976
- /3/ Miller, R.
Response Time in Man-Computer Conversational
Interaction. Proc. AFIPS 68,
New York (1968)

- /4/ Braden, H.V.; Wulf, W.A. The Implementation of a BASIC-System in a Multiprogramming Environment
Comm. ACM, vol. 11 (1968), No. 10, S. 688 - 692
- /5/ Hohberg, B.; Paulin, G. Compilertechnik
Berlin, Verlag Technik 1975
- /6/ Schmidt, H. The Structure of a Flexible and Efficient Compiler for Algorithmic Conversational Languages. In Lecture Notes in Economics and Mathematical Systems 75
1. Fachtagung über Programmiersprachen
Springer Verlag 1972
- /7/ Early, J.; Caizergues, P. A Method for Incrementally Compiling Languages with Nested Statement Structure
CACM, No. 12, 15 (1972), S. 1040 - 1044

Vortrag gehalten auf der 4. Tagung der DDR-ALGOL-Gruppe in Rostock, Februar 1976 (s. auch Heft 2 dieser Reihe).

eingegangen: 23. 4. 1976

Anschrift des Verfassers:

Dr.rer.nat. Bodo Hohberg
Humboldt-Universität zu Berlin
Sektion Mathematik
Bereich Informationsverarbeitung
DDR 108 Berlin
Unter den Linden 6

PL/1-Spracherweiterung durch Benutzung der Möglichkeiten des Makrointerpreters

In meinem Beitrag möchte ich einige Gedanken zur Anwendung eines Konzeptes von PL/1 äußern.

Beim Entwurf von PL/1 wurden von den Schöpfern dieser Sprache Möglichkeiten vorgesehen, auf dem Quelltextniveau Änderungen vornehmen zu können. In der Literatur werden diese Möglichkeiten als Möglichkeiten zur Übersetzungszeit, Vorübersetzermöglichkeiten oder Makromöglichkeiten bezeichnet. Die Überführung des Quelltextes in das Zielprogramm wird in den PL/1-Implementierungen in zwei Etappen realisiert. In der ersten wird der Quelltext entsprechend der in ihm selbst enthaltenen Makroanweisungen modifiziert. Der Compilerteil, der diese Vorübersetzung verwirklicht, heißt Makrointerpreter. In der zweiten Etappe erfolgt die eigentliche Compilierung. Für die z. Z. zur Verfügung stehenden Betriebssysteme für ESER-Anlagen, ist im OS/ES der PL/1-Makrointerpreter integraler Bestandteil des entsprechenden PL/1-Compilers während für den PL/1-Subset-Compiler des DOS/ES ein PL/1-Makrointerpreter als zusätzliches Programm angeboten wird. Bezüglich der realisierten PL/1-Makromöglichkeiten besteht zwischen beiden Betriebssystemen Kompatibilität.

Infolge der schon im Aufbau von PL/1 liegenden Unabhängigkeit der PL/1-Makromöglichkeiten von den anderen Ausdrucksmitteln von PL/1 und des Verfahrens der Übersetzung kann man die PL/1-Makromöglichkeiten als in sich geschlossene Sprache ansehen. Losgelöst von ihrem eigentlichen Ziel, der Erzeugung eines PL/1-Quellprogrammes, kann man so diese PL/1-Makrosprache als eine Sprache zur Beschreibung spezieller Zeichenkettentransformationen charakterisieren. Das Ausgangsobjekt für die Transformation ist eine Zeichenkette, im speziellen der PL/1-Quelltext, die neben dem zu ändernden Text die die Änderungen beschreibenden (PL/1-Makro-) Anweisungen enthält. Es sind fol-

gende Arten von Textänderungen möglich:

1. Erweiterung des Textes durch Texte aus Bibliotheken
2. Auswahl von Textteilen
3. Ersetzung ausgewählter Bezeichnungen (Namen) im Text durch ihnen zugeordnete Werte

Die unter 3. angegebene Änderungsart geschieht derart, daß eine Durchmusterung des Textes hinsichtlich ausgewählter Bezeichnungen (Namen) erfolgt und eine Ersetzung des Namens durch den zugeordneten Wert vorgenommen wird. Dabei ist wesentlich, daß die Reihenfolge der Durchmusterung gesteuert werden kann und der Ersetzungsprozeß rekursiver Gestalt ist. Die Werte können vom Typ "Zeichenkette" oder "dezimale Festkommazahl" sein. Es steht eine Festkommaarithmetik zur Verfügung. Ausdrücke können gebildet werden. Die PL/1-Makromöglichkeiten enthalten ein Prozedurkonzept. Bei Verwendung dieser Prozeduren, die vom Gebrauch her den PL/1-Funktionsprozeduren ähneln und Argument-Parameter-Beziehungen zulassen, verfügt der Programmierer über ein sehr starkes Ausdrucksmittel, das er insbesondere für komplizierte Werteberechnungen einsetzen kann.

Der PL/1-Makrointerpreter nimmt in der Ausgangszeichenkette die gewünschten Änderungen vor und stellt den geänderten Text, der keine PL/1-Makroanweisungen mehr enthält, in der Ausgabezeichenkette, im speziellen als PL/1-Quellprogramm, zur Verfügung. Danach kann die eigentliche PL/1-Compilierung stattfinden.

Als Anwendungen des Konzeptes der PL/1-Makromöglichkeiten sollen genannt werden:

1. Verwendung standardisierter Quelltextteile, die in Bibliotheken gespeichert sind.
Besonders erscheint das sinnvoll für umfangreiche Vereinbarungen, wie beispielsweise für den Satzaufbau von Dateien.
2. Verwendung von Abkürzungen oder Wahl anderer Bezeichnungen für Schlüsselwörter und Einführung von Kurzformen für wiederholt benötigte Anweisungsteile oder Anweisungen.

3. Schaffung und Verwendung problemnaher Formulierungsmittel, die in ihren Ausdrucksmöglichkeiten denjenigen von Fachsprachen nahekommen. Durch Programmierung geeigneter PL/1-Makro-Prozeduren kann eine Menge von "Makros" bereitgestellt werden, mittels derer dann die Programmierung eines Problems leicht auch durch einen PL/1 nichtkennenden Programmierer vorgenommen werden kann. In diesem Sinne kann eine Erweiterung der Sprachmöglichkeiten von PL/1 erfolgen.

In der "PL/1-Sprachbeschreibung" für das OS/ES und in der "Anwendungsbeschreibung für den Makrointerpreter für das DOS/ES" wurde versucht, für einige elementare Aufgaben der Analytischen Geometrie der Ebene eine solche Makrosprache zu konstruieren.

Ein in dieser Makrosprache geschriebenes Programm könnte beispielsweise wie folgt aussehen:

```
% INCLUDE ANALYT;
BEGINN(GEOM)
  VER(PUNKT,(A,B))
  VER(SKALAR,(ABST))
  LIES((A,B))
  ABSTPP(ABST,A,B)
  DRUCKE((ABST))
ENDE
```

Durch den PL/1-Makrointerpreter wird daraus ein zulässiges PL/1-Quellprogramm generiert. Im Programm werden zwei Variablen vom Typ "PUNKT" und eine Variable vom Typ "SKALAR" vereinbart. Die Abszissen- und Ordinatenwerte der beiden Punkte A und B werden eingelesen. Es wird der Abstand ABST zwischen diesen beiden Punkten berechnet und gedruckt.

Die Makros BEGINN (...) und ENDE dienen zur Begrenzung des Programmes. Durch die Anweisung % INCLUDE ANALYT; werden die in der Bibliothek unter dem Namen ANALYT gespeicherten vorprogrammierten Makros dem Makrointerpreter zur Verfügung gestellt.

Ich hoffe, durch meinen Beitrag einige Anregungen gegeben zu haben, wie durch Nutzung der PL/1-Makromöglichkeiten die Pro-

grammierung erleichtert und der Programmieraufwand gesenkt werden kann, was zu einer Erhöhung der Produktivität in der Programmierung führt.

Vortrag, gehalten auf der 4. Tagung der DDR-ALGOL-Gruppe in Rostock, Februar 1976 (s. auch Heft 2 dieser Reihe)

eingegangen: 23. 4. 1976

Anschrift des Verfassers:

Dipl.-Math. Dietmar Schier
VEB Robotron
ZFT Dresden
DDR 801 Dresden
Leningrader Str. 15

Zur Auswahl wesentlichen mathematischen Wissens hinsichtlich der inhaltlichen Abstimmung mit den Schulfächern Physik, Chemie und Biologie

Im Rahmen der Weiterentwicklung des sozialistischen Bildungssystems nehmen Fragen der Erhöhung des Ausbildungsniveaus im Fachunterricht der allgemeinbildenden Schule breiten Raum ein. Neben fachspezifischen Anliegen der einzelnen Unterrichtsfächer sind in letzter Zeit Erörterungen über fachübergreifende Ziel- und Aufgabenstellungen unter dem Gesichtspunkt der allseitigen Entwicklung sozialistischer Persönlichkeiten mehr und mehr in den Mittelpunkt gerückt.

Von Interesse sind hierbei Überlegungen und Maßnahmen hinsichtlich inhaltlicher und zeitlicher Abstimmung zwischen verschiedenen Unterrichtsfächern. Über einzelne grundlegende Ergebnisse hierzu zwischen dem Mathematikunterricht und den Schulfächern Physik, Chemie und Biologie wurde bereits berichtet (/1/, /2/). Konsequenzen für den Unterricht in Mathematik wurden ebenso dargestellt (/3/, /4/, /5/) wie Hinweise zur Festigung und Nutzung mathematischen Wissens und Könnens im naturwissenschaftlichen Unterricht (/6/). Grundlage dieser praxisorientierten Darlegungen waren die gültigen Lehrpläne für den Unterricht in Mathematik, Physik, Chemie und Biologie sowie die hierzu entwickelten Nachfolgematerialien (Lehrbücher, Unterrichtshilfen, Einzeldarstellungen fachlicher und methodischer Art).

Die dynamische Entwicklung unserer Gesellschaft bringt in allen Bereichen ständig neue Probleme und Aufgaben hervor. Als eine solche ist innerhalb der Schulausbildung die Frage nach notwendigen Veränderungen des Bildungsinhalts der Schulfächer anzusehen. Hier stehen wir vor einer langwierigen und komplizierten Aufgabe, da neben Grundsatzfragen pädagogischer Art vor allem auch neue fachwissenschaftliche Ergebnisse in enger Verbindung

mit Weiterentwicklungen von Pädagogik, Psychologie und Fachmethodik berücksichtigt werden müssen.

Untersuchungen zur Auswahl und Anordnung des fachspezifischen Wissens müssen die fachübergreifenden Ziel- und Aufgabenstellungen mit beinhalten. Für das Fach Mathematik ergeben sich aus dieser Sicht besonders vielfältige Aufgaben, da mathematisches Wissen wegen seines großen Umfanges und der vielseitigen Einsatz- und Verwendungsmöglichkeit in der Schulausbildung neben der Muttersprache eine gewisse Schlüsselstellung einnimmt. Ein Ausdruck hierfür ist der hohe Anteil des Mathematikunterrichts im System der Unterrichtsfächer; nahezu jede fünfte Unterrichtsstunde der allgemeinbildenden Schule ist eine Mathematikstunde.

Im Rahmen theoretischer Analysen der Materialien des naturwissenschaftlichen Unterrichts wurde das hier einbezogene mathematische Wissen in Einzelarbeiten unserer Forschungsgruppe zusammengestellt. Durch empirische Untersuchungen in allgemeinbildenden polytechnischen Oberschulen sind wir gegenwärtig bemüht, den Stand der Verwendung mathematischen Wissens in den Fächern Physik und Chemie zu ermitteln.

Die bisher vorliegenden Ergebnisse beider Untersuchungsarten lassen erkennen, daß bei zielgerichteter und zweckmäßiger Nutzung der Vorleistungen des Mathematikunterrichts die Ergebnisse der naturwissenschaftlichen Ausbildung der Schüler verbessert werden können. Von Mathematiklehrern wird aber immer wieder darauf verwiesen, daß eine zielstrebigere Vorbereitung des naturwissenschaftlichen Unterrichts möglich wäre, wenn im Lehrplan des Mathematikunterrichts das von den Schülern zu erwerbende mathematische Wissen, welches für fachübergreifende Aufgabenstellungen als wesentlich angesehen werden muß, explizite ausgewiesen würde. Dieses würde auch den Lehrern der naturwissenschaftlichen Fächer helfen, exaktere Schlußfolgerungen für den Einsatz, die Festigung und gelegentliche Erweiterung mathematischen Wissens zu ziehen und im Unterricht durchzusetzen.

Notwendigkeit und Bedeutung der Auswahl wesentlichen mathema-

tischen Wissens im fachübergreifenden Sinne haben in unserer Forschungsgruppe zu umfangreichen wissenschaftlichen Auseinandersetzungen geführt. Die dabei gewonnenen Aspekte hinsichtlich des Vorgehens hierbei sollen nachfolgend umrissen werden.

1. Grundlegende Aspekte

- Der Begriff "wesentliches mathematisches Wissen im fachübergreifenden Sinne" darf nicht statisch aufgefaßt werden; er unterliegt vielmehr der Gesamtentwicklung von Zielen, Inhalten und Prozessen des Schulunterrichts.
- Umfang und Tiefe des von den Schülern zu erwerbenden mathematischen Wissens fachübergreifender Art sollten in den Lehrplänen der einzelnen Fächer möglichst exakt ausgewiesen werden.
- Den Lehrern der Mathematik und der Naturwissenschaften sind fachliche und methodische Hinweise zu geben für das zielgerichtete und abgestimmte Vorgehen bei der Einführung, Festigung und Nutzung mathematischen Wissens im fachübergreifenden Sinne.

2. Zielaspekte

- Bei der Auswahl des wesentlichen mathematischen Wissens im fachübergreifenden Sinne ist seine Bedeutung für die Persönlichkeitsentwicklung des Schülers von erstrangiger Bedeutung. Das kritisch ausgewählte Wissen dieser Art muß sich organisch in die Zielstellungen der Wissensvermittlung, der Könnensentwicklung und der Erziehung der allgemeinbildenden Schule einfügen lassen.
- Wesentliches mathematisches Wissen im fachübergreifenden Sinne hat besonders enge Beziehungen zu den in den Schulfächern Physik, Chemie und Biologie zu erreichenden Zielen. Deshalb bedürfen diese besonderer Beachtung bei Festlegungen mathematischer Inhalte.
- Das wesentliche mathematische Wissen im fachübergreifenden Sinne muß überwiegend im Mathematikunterricht bereitgestellt

werden. Es muß sich in die fachspezifischen Anliegen dieses Faches so einordnen, daß die Systematik bei der Erreichung mathematischer Ziele und Aufgaben nicht gestört wird.

- Wesentliches mathematisches Wissen, das im naturwissenschaftlichen Unterricht ohne entsprechende Vorleistungen aus dem Mathematikunterricht eingeführt und genutzt wird, bedarf einer zweckmäßigen Abstimmung hinsichtlich der Ziele, Inhalte und Methoden des Mathematikunterrichts.

3. Inhaltliche Aspekte

- Zum wesentlichen mathematischen Wissen im fachübergreifenden Sinne gehören solche Wissens Elemente, die unabdingbare Voraussetzung für die fachspezifische Wissensvermittlung im naturwissenschaftlichen Unterricht sind.
- Zu den Bestandteilen des wesentlichen mathematischen Wissens im fachübergreifenden Sinne zählen solche Elemente, die im naturwissenschaftlichen Unterricht für die Ausprägung fachspezifischer Könnensqualitäten benötigt werden.
- Mathematisches Wissen kann im fachübergreifenden Sinne wesentlich sein, wenn es in hohem Maße beiträgt zur Erreichung fachspezifischer Erziehungsanliegen der einzelnen naturwissenschaftlichen Unterrichtsfächer.

4. Aspekte aus der Sicht des Unterrichtsprozesses

- Grundlegendes mathematisches Wissen kann im Unterricht erfolgreich bei der "Verdichtung" der in verschiedenen Fächern erreichten Einzelkenntnisse von Bedeutung sein. Das gilt insbesondere für Festigungsphasen und -stunden.
- Mathematische Kenntnisse - insbesondere Verfahren und Methoden dieser Wissenschaft - ermöglichen die Aufdeckung von Beziehungen, Verflechtungen und Durchdringungen zwischen den Erkenntnissen verschiedener Wissenschaften. Manche Erkenntnisse solcher Art können einen guten Beitrag zur erfolgrei-

chen Gestaltung des naturwissenschaftlichen Unterrichts leisten; insbesondere hinsichtlich der Gewinnung tiefer theoretischer bzw. praxisorientierter Einsichten (polytechnische Zielsetzungen) der Schule.

- Mathematische Vorleistungen bieten vielfältige Anknüpfungspunkte für Einführungen und Bearbeitungen naturwissenschaftlicher Stoffe. Von besonderer Bedeutung sind dabei solche mathematischen Elemente des Wissens, die für überzeugende Motivierungen und moderne Behandlungs- und Betrachtungsweisen genutzt werden können.
- Der nicht einfach zu handhabende Prozeß der bewußten Erziehung der Schüler im Unterricht bedarf zweckmäßiger Abstimmungen zwischen mathematisch-naturwissenschaftlichen Fächern. Mathematisches Wissen kann wesentlich dazu beitragen, spezielle Anliegen des naturwissenschaftlichen Unterrichts (z.B. kritische Wertung empirisch ermittelter Daten) als auch solche von allgemeinerer Bedeutung (u. a.: Erziehung von Einsichten in die Zusammenhänge der historischen Entwicklung von Mathematik und Naturwissenschaften) zu unterstützen.

Literatur

- /1/ Sietmann, G. Zu Fragen der Koordinierung im mathematisch-naturwissenschaftlichen Unterricht der allgemeinbildenden Schule
Kölbl, I. Wiss. Zeitschrift der Universität Rostock, Math.-nat. Reihe, 23, 8, 677 - 680 (1974).
- /2/ Sietmann, G. Einige Aspekte zur inhaltlichen Abstimmung zwischen dem Mathematikunterricht und den naturwissenschaftlichen Unterrichtsfächern der allgemeinbildenden Oberschule
Kölbl, I. Wiss. Zeitschrift der Universität Rostock, Math.-nat. Reihe, 24, 10, 1283 - 1285 (1975).

- /3/ Sietmann, G. Abstimmungen zwischen dem Mathematikunterricht und dem Unterricht in Physik, Chemie und Biologie
Mathematik in der Schule 14, 2/3, 97 - 100
(1976).
- /4/ Kölbl, I. Zur zeitlichen Koordinierung des Mathematikunterrichts mit anderen Unterrichtsfächern
Mathematik in der Schule 14, 6, 308 - 312
(1976).
- /5/ Leskien, B. Zu mengentheoretischen Betrachtungen im mathematischen und naturwissenschaftlichen Unterricht
Mathematik in der Schule 14, 11, 595 - 600
(1976).
- /6/ Sietmann, G. Zur Festigung mathematischen Wissens und Könnens im naturwissenschaftlichen Unterricht
Rostocker Mathematisches Kolloquium, Heft 1,
97 - 104 (1976).

eingegangen: 28. 3. 1977

Anschrift des Verfassers:

Prof. Dr. paed. habil. Günter Sietmann
Wilhelm-Pieck-Universität Rostock
Sektion Mathematik
DDR-25 Rostock
Universitätsplatz 1

Zur Definition der reellen Zahlen durch Dezimalbrüche

Wenn dieses Thema hier erneut aufgegriffen wird, so muß zunächst durch eine kritische Betrachtung der Literatur der Standpunkt bestimmt werden. Die im Lehrbuch der Klasse 9 /5/ gegebene Definition: Eine reelle Zahl ist ein unendlicher Dezimalbruch ohne Neunerperiode, soll auch für uns Ausgangspunkt sein. Im Lehrbuch /5/ wird jedoch ausdrücklich betont, daß auf die Definition der Rechenoperationen und den Nachweis der Rechengesetze verzichtet werden muß. Somit ergibt sich die Notwendigkeit, Mathematiklehrern und Studenten weitgehend schulgemäße Möglichkeiten zur Ausfüllung dieser Lücke aufzuzeigen.

Die Grundgedanken für den im folgenden eingeschlagenen Weg finden sich in dem Beitrag /2/ von Flachsmeyer und Terpe, allerdings nur in skizzierter Form ausgeführt. Eine Ausarbeitung dieses Ansatzes, die jedoch nicht immer bis in letzte Details ging und zum anderen recht frühzeitig Cauchy-Folgen heranzog, hat Prof. Dr. G. Burosch an der Universität Rostock 1971/72 in Vorlesungen vorgetragen, für die ich Übungen zu betreuen hatte und dadurch zur Beschäftigung mit dem Thema angeregt wurde. Als wesentlich neues Moment gegenüber diesen beiden Darstellungen soll jetzt die Addition sogleich für beliebige Dezimalbrüche definiert, also nicht eine Definition von Addition und Subtraktion zunächst nur für nichtnegative Zahlen und anschließende Übertragung durch Axiome auf die übrigen Fälle vorgenommen werden; dadurch lassen sich die additiven Strukturgesetze geschlossener gewinnen. Wisliceny konstruiert in seinem Buch /6/ den Bereich der nichtnegativen reellen Zahlen \mathbb{R}_+ und begnügt sich mit dem Hinweis, daß danach die Erweiterung zum Bereich der reellen Zahlen \mathbb{R} in der gleichen Weise vorgenommen werden kann wie die vorher behandelte Erweiterung vom Bereich der gebrochenen Zahlen \mathbb{Q}_+ zum Bereich der rationalen Zahlen \mathbb{Q} . Ebendieses Vorgehen findet man in dem Beitrag /4/ von Lemke und Stoye. Auch die Arbeit /3/ von Holland ist eine Begründung des Rech-

nens mit nichtnegativen Dezimalbrüchen.

Ich möchte davon ausgehen, daß vor Einführung der reellen Zahlen den Lernenden der Bereich der rationalen Zahlen bekannt (innerhalb eines systematischen Kurses entwickelt worden) ist, und es scheint mir daher überzeugender (und auch rationeller) zu sein, wenn man eine vorläufige Definition und Untersuchung zunächst nur der nichtnegativen reellen Zahlen mit anschließender abermaliger Erweiterung vermeidet. Zum Abtun dieser abermaligen Erweiterung mit einem Hinweis auf den Übergang von \mathbb{Q}_+ nach \mathbb{Q} sei noch folgendes bemerkt: Konstruiert man nach \mathbb{Q} zunächst \mathbb{R}_+ , so erfolgt dies aus dem Teilbereich \mathbb{Q}_+ von \mathbb{Q} , der sodann als isomorph zu einem Teilbereich von \mathbb{R}_+ nachgewiesen wird. Der anschließende Übergang von \mathbb{R}_+ nach \mathbb{R} enthält eine Überlegung zur isomorphen Einbettung des Bereiches \mathbb{R}_+ in \mathbb{R} , und damit von \mathbb{Q}_+ in \mathbb{R} . Als weitere Frage wäre aber noch die isomorphe Einbettung des gesamten Bereiches \mathbb{Q} in \mathbb{R} zu behandeln. - In den folgenden Ausführungen wird von Anfang an sofort der Bereich aller reellen Zahlen behandelt, und es soll versucht werden, die Definitionen der Rechenoperationen und die Beweise der zugehörigen Strukturgesetze gänzlich auf Dezimalbrüche zu gründen, ja die Definition der Operationen zu orientieren an dem praktischen (vielfach heuristischen) Umgang mit Dezimalbrüchen. ^{1/} Dabei tritt sehr bald der Begriff des Grenzwertes von Zahlenfolgen hervor, jeder Dezimalbruch ist Grenzwert einer speziellen Folge (s. (2.5)), und für die Rechenoperationen gewinnt man Beziehungen (s. (2.6), (3.0), (3.9)), deren Gültigkeit dem Rechnen in der Praxis eine Rechtfertigung gibt. Der Kürze halber werden wir Beweise vielfach nicht ausführen und auf die (unschwer zu gewinnende) isomorphe Einbettung von \mathbb{Q} in \mathbb{R} schon nicht mehr eingehen.

^{1/} In /6/ werden die Dezimalbrüche im wesentlichen lediglich zum Beweis der Stetigkeit (vgl. unten Satz (1.4)) herangezogen, während die Operationen und Strukturgesetze auf dem Begriff der Intervallschachtelung beruhen.

1. Festlegung der Ausgangsposition. Ordnungsrelation in \mathbb{R}

Im folgenden bezeichne \mathbb{N} den Bereich der natürlichen Zahlen, \mathbb{N}^{\neq} die Menge $\mathbb{N} \setminus \{0\}$, Ziffern Elemente der Menge $\{0, 1, \dots, 9\}$, \mathbb{Z} den Bereich der ganzen Zahlen, \mathbb{Q}_+ den Bereich der gebrochenen Zahlen, \mathbb{Q} den Bereich der rationalen Zahlen, $\tilde{\mathbb{R}}$ die Menge aller (unendlichen) Dezimalbrüche (d.h. $\tilde{\mathbb{R}} = \{a_0, a_1 a_2 \dots : a_0 \in \mathbb{Z} \wedge \bigwedge_{i \geq 1} a_i \text{ Ziffer}, \text{ wobei } a_0, a_1 a_2 \dots = b_0, b_1 b_2 \dots \Leftrightarrow \bigwedge_i a_i = b_i\}$), $\mathbb{R}(p)$ die Menge aller periodischen Dezimalbrüche ohne 9er-Periode, $\mathbb{R}(e)$ die Menge aller endlichen (abbrechenden) Dezimalbrüche (Periode 0), \mathbb{R} die Menge aller Dezimalbrüche ohne 9er-Periode (reelle Zahlen). Setzt man $\mathbb{Q} = \left\{ \frac{r}{s} : r \in \mathbb{N} \wedge s \in \mathbb{N}^{\neq} \right\}$, dann ist folgender Divisionsalgorithmus DA bekannt: Zu gegebenen r und s existieren eindeutig bestimmte $a_0, a_1, \dots, t_0, t_1, \dots$, so daß gilt

$$r = a_0 \cdot s + t_0, \quad a_0 \in \mathbb{N}, \quad 0 \leq t_0 < s,$$

$$10 \cdot t_0 = a_1 \cdot s + t_1, \quad a_1 \text{ Ziffer} \quad 0 \leq t_1 < s,$$

$$10 \cdot t_{n-1} = a_n \cdot s + t_n, \quad a_n \text{ Ziffer}, \quad 0 \leq t_n < s \quad (n \in \mathbb{N}^{\neq}).$$

Aus diesen Gleichungen folgt in \mathbb{Q}_+

$$r = \left(a_0 + \frac{a_1}{10} + \frac{a_2}{10^2} + \dots + \frac{a_n}{10^n} \right) \cdot s + \frac{t_n}{10^n},$$

(1.1)

$$r = a_0, a_1 a_2 \dots a_n \cdot s + \frac{t_n}{10^n}.$$

Wir denken uns DA unendlich fortgeführt, setzen $DA\left(\frac{r}{s}\right) := a_0, a_1 a_2 \dots$ und erhalten damit eine Abbildung DA von

$\left\{ \frac{r}{s} : r \in \mathbb{N} \wedge s \in \mathbb{N}^{\neq} \right\}$ in $\tilde{\mathbb{R}}$. Da man aus $\frac{r}{s} = \frac{u}{v}$ in \mathbb{Q}_+ auch $DA\left(\frac{r}{s}\right) = DA\left(\frac{u}{v}\right)$ folgert, ist DA eine Abbildung von \mathbb{Q}_+ in $\tilde{\mathbb{R}}$.

Wir definieren durch

$$D\left(+\frac{r}{s}\right) := + DA\left(\frac{r}{s}\right), \quad D\left(-\frac{r}{s}\right) := - DA\left(\frac{r}{s}\right)$$

eine für das Weitere wichtige Abbildung D von \mathbb{Q} in $\tilde{\mathbb{R}}$. Mit bekannter Argumentation überlegt man, daß DA stets periodische Dezimalbrüche, allerdings niemals die 9er-Periode liefert, und damit ist D eine Abbildung von \mathbb{Q} in $\mathbb{R}(p)$.

Betrachtet man noch die Teilmenge $\left\{ \frac{a}{10^n} : a \in \mathbb{Z} \wedge n \in \mathbb{N} \right\}$ von \mathbb{Q} , so vermittelt D eine einfache Beziehung zwischen dieser Menge und $\mathbb{R}^{(e)}$: Man schreibt in \mathbb{Q}

$$\frac{a_0 a_1 \dots a_n}{10^n} = a_0, a_1 \dots a_n \quad (a_0 \in \mathbb{Z}; a_1, \dots, a_n \text{ Ziffern}),$$

und andererseits gilt

$$D\left(\frac{a_0 a_1 \dots a_n}{10^n}\right) = a_0, a_1 \dots a_n \bar{0} \in \mathbb{R}^{(e)},$$

so daß man die genannte Teilmenge praktisch mit $\mathbb{R}^{(e)}$ identifizieren kann, da D zwischen diesen Mengen eine 1-1-Abbildung ist. Wir machen uns dies zunutze und setzen

$$(1.2) \quad \mathbb{R}^{(e)} = \left\{ \frac{a}{10^n} : a \in \mathbb{Z} \wedge n \in \mathbb{N} \right\},$$

dadurch auch

$$D\left(\frac{a}{10^n}\right) = \frac{a}{10^n}.$$

In $\mathbb{R}^{(e)}$ sind dann durch \mathbb{Q} eine Ordnungsrelation $<$, eine Addition $+$ sowie eine Multiplikation \cdot definiert, und zwar weiß man über die 'üblichen' Strukturgesetze folgendes:

- $<$ ist eine irreflexive totale Ordnung in $\mathbb{R}^{(e)}$,
- $+$ ist eine Operation in $\mathbb{R}^{(e)}$, die assoziativ, umkehrbar, kommutativ, bez. $<$ monoton ist,
- \cdot ist eine Operation in $\mathbb{R}^{(e)}$, die assoziativ, kommutativ, bez. $+$ distributiv, bez. monoton, jedoch i. allg. nicht umkehrbar ist.

Auf $\mathbb{R}^{(e)}$ gründen wir die Definition von $<$, $+$, \cdot in dem umfassenderen Bereich \mathbb{R} und übertragen die in $\mathbb{R}^{(e)}$ gültigen Strukturgesetze. Zur Behandlung der Umkehrbarkeit der Multiplikation in \mathbb{R} in Abschnitt 3 werden wir D abermals heranzuziehen haben.

Ist a ein Dezimalbruch a_0, a_1, a_2, \dots , so sei $a^{(n)} := a_0, a_1, a_2, \dots, a_n \bar{0}$ gesetzt ($n \in \mathbb{N}$). ($a^{(0)}, a^{(1)}, a^{(2)}, \dots$) heißt Folge der kanonischen Näherungen von a . Mit den $a^{(n)}$ können wir, da sie in $\mathbb{R}^{(e)}$ liegen, rechnen.

Definition: Für Dezimalbrüche a und b aus \mathbb{R} sei

$$a <_R b \Leftrightarrow \bigvee_n a^{(n)} < b^{(n)}.$$

Man sieht unmittelbar, daß für Elemente $a^{(m)}, b^{(m)}$ aus $\mathbb{R}^{(e)}$ genau dann $a^{(m)} <_R b^{(m)}$ gilt, wenn auch in $\mathbb{R}^{(e)}$ schon $a^{(m)} < b^{(m)}$ ist; wir wollen daher die $<$ -Zeichen nicht unterscheiden.

(1.3) Satz: $<$ ist eine irreflexive totale Ordnung in \mathbb{R} .

Den leichten Beweis können wir übergehen. - In dem folgenden Satz, der für den weiteren Aufbau wesentlich ist, wird die wichtigste gegenüber \mathbb{Q} neue Eigenschaft von \mathbb{R} ausgedrückt (die Stetigkeit).

(1.4) Satz: In \mathbb{R} besitzt jede nichtleere nach oben (unten) beschränkte Teilmenge M ein Supremum (Infimum).

Der Beweis für die Existenz von $\sup(M)$ in dem Fall, daß M positive Zahlen enthält, kann aus /6/ übernommen werden. Ganz ähnlich können die Beweise für die anderen Fälle geführt, auch mittels der Beziehungen $\sup(M) = -\inf(-M)$, $\inf(M) = -\sup(-M)$ ($-M := \{-x : x \in M\}$) teilweise vereinfacht werden.

Insbesondere gilt $a = \sup \{a^{(n)}\}$ für $a \geq 0$, dagegen $a = \inf a^{(n)}$ für $a < 0$. Das ständige Gegenwärtigsein der Folge $(a^{(n)})$ als "Annäherungsfolge" (später werden wir schreiben $a = \lim_{n \rightarrow \infty} a^{(n)}$), als Definition von a , ist eine wesentliche Voraussetzung zum Verständnis des Weges, reelle Zahlen als Dezimalbrüche einzuführen.

Für die Untersuchung der Addition in \mathbb{R} stellen wir uns noch ein Hilfsmittel bereit und betrachten für $a \in \mathbb{R}$ in $\mathbb{R}^{(e)}$ neben der Folge $(a^{(n)})$ die Folge $(\tilde{a}^{(n)})$, deren Glieder definiert werden durch

$$\tilde{a}^{(n)} := a^{(n)} - \frac{1}{10^n}, \quad n \in \mathbb{N}.$$

(1.5) Für beliebiges a aus \mathbb{R} gelten die Aussagen

$$\bigwedge_n \tilde{a}^{(n+1)} \geq \tilde{a}^{(n)} \quad (\text{d.h., } (\tilde{a}^{(n)}) \text{ ist monoton wachsend}),$$

es gibt beliebig große n mit $\tilde{a}^{(n+1)} > \tilde{a}^{(n)}$,

$a = \sup \{a^{(n)}\}$ (unabhängig vom Vorzeichen!).

Beweis: Aus den Beziehungen

$$\begin{aligned} \tilde{a}^{(n+1)} &= a^{(n+1)} - \frac{1}{10^{n+1}} = a^{(n)} \pm \frac{a_{n+1}}{10^{n+1}} - \frac{1}{10^{n+1}} \\ &\geq a^{(n)} - \frac{9}{10^{n+1}} - \frac{1}{10^{n+1}} = a^{(n)} - \frac{1}{10^n} = \tilde{a}^{(n)} \end{aligned}$$

folgt, daß $\tilde{a}^{(n+1)} = \tilde{a}^{(n)}$ höchstens für $a_{n+1} = 9$ möglich ist, und somit tritt, da a keine 9er-Periode hat, immer wieder das $>$ -Zeichen ein. Weil stets $\tilde{a}^{(n)} < a$ ist, aber andererseits $\tilde{a}^{(n)}$ und $a^{(n)}$ mit wachsendem n um beliebig wenig differieren, gilt $\sup \{\tilde{a}^{(n)}\} = a$.

2. Addition in \mathbb{R}

Wünschenswert als Ziel der Überlegungen ist, für beliebige a, b aus \mathbb{R} die Summe $a + b$ so zu definieren, daß

$$a + b = \lim_{n \rightarrow \infty} (a^{(n)} + b^{(n)})$$

ist. Doch wir können diese Beziehung nicht zur Definition von $a + b$ verwenden, weil dazu eine präzise Fassung des Grenzwertbegriffes nötig ist; sofern wir nicht noch andere Begriffe heranziehen wollen (etwa den der Intervallschachtelung), lautet eine bekannte Definition: Für Folgen (g_n) gilt

$$(2.1) \quad \lim_{n \rightarrow \infty} g_n = g \iff \bigwedge_{\epsilon > 0} \bigvee_{N \in \mathbb{N}} \bigwedge_{n > N} (|g_n - g| < \epsilon),$$

und hier wird in $g_n - g = g_n + (-g)$ die Addition benutzt, die somit vor der Einführung dieses Grenzwertbegriffes definiert werden muß.

Beschränkt man sich am Anfang auf nichtnegative Dezimalbrüche, so wäre die Definition

$$a + b := \sup \{a^{(n)} + b^{(n)}\} \quad (a \geq 0, b \geq 0)$$

möglich. Die Einbeziehung der negativen Dezimalbrüche, könnte durch zusätzliche Axiome erfolgen, wobei die Beweise der Strukturgesetze allerdings unter den damit notwendigen Fallunterscheidungen leiden. Um dies zu vermeiden, benutzen wir folgende

Definition: Für a und b aus \mathbb{R} sei

$$a + b := \sup_{\mathbb{R}} \{ \tilde{a}^{(n)} + \tilde{b}^{(n)} \}.$$

(Das Supremum existiert nach Satz (1.4), da a_0+1+b_0+1 eine obere Schranke ist.)

Zunächst stellen wir fest, daß die neu definierte Addition $\overset{+}{\mathbb{R}}$ eine Erweiterung der schon in $\mathbb{R}^{(e)}$ bekannten $+$ ist, d.h., für Elemente $a^{(m)}, b^{(m)}$ aus $\mathbb{R}^{(e)}$ gilt $a^{(m)} \overset{+}{\mathbb{R}} b^{(m)} = a^{(m)} + b^{(m)}$, denn es ist $a^{(m)} \overset{+}{\mathbb{R}} b^{(m)} = \sup_{\mathbb{R}} \left\{ a^{(m)}(n) - \frac{1}{10^n} + b^{(m)}(n) - \frac{1}{10^n} \right\}$; nun sind die Glieder $a^{(m)}(n) - \frac{1}{10^n} + b^{(m)}(n) - \frac{1}{10^n}$ für beliebiges n kleiner als $a^{(m)} + b^{(m)}$, jedoch für $n \geq m$ gleich $a^{(m)} + b^{(m)} - \frac{2}{10^n}$, womit $a^{(m)} + b^{(m)}$ das Supremum ist. - Wir brauchen daher im weiteren die $+$ -Zeichen nicht zu unterscheiden.

Erste Struktureigenschaften der Addition in \mathbb{R} sind in dem folgenden, leicht zu beweisenden Satz zusammengefaßt.

- (2.2) Satz:
1. $\bigwedge_{a,b} a + b = b + a$;
 2. $\bigwedge_a a + 0 = a$;
 3. $\bigwedge_a \bigvee_{-a} a + (-a) = 0$; nämlich
 $-a := -a_0, a_1, \dots$ für $a = a_0, a_1, \dots$,
 $-a := a_0, a_1, \dots$ für $a = -a_0, a_1, \dots$, wobei
 $a_0 \in \mathbb{N}$: (die eindeutige Bestimmtheit von $-a$ ist in Satz (2.3) enthalten);
 4. $\bigwedge_{a,b,c} (a \leq b \Rightarrow a + c \leq b + c)$.

Etwas schwieriger ist der Beweis des Assoziativgesetzes, denn man hat etwa $(a + b) + c = \sup \{ \tilde{a}^{(n)} + \tilde{b}^{(n)} \} + c$, aber für die weitere Umformung nach Definition der Addition fehlt die Folge der kanonischen Näherungen des ersten Summanden (jedenfalls ist i. allg. $(a + b)^{(n)} \neq a^{(n)} + b^{(n)}$). Wir beweisen dazu einen Hilfssatz: Seien (g_n) und (h_n) monoton wachsende und beschränkte Folgen in \mathbb{R} . Dann ist

$$\sup \{g_n\} + \sup \{h_n\} = \sup \{g_n + h_n\}.$$

Beweis: Es sei $a = \sup \{\tilde{a}^{(n)}\} := \sup \{g_n\}$,

$$b = \sup \{\tilde{b}^{(n)}\} := \sup \{h_n\}.$$

Nach (1.5) ist stets $\tilde{a}^{(n)} < a$, und wegen $\sup \{g_n\} = a$ folgt daraus $\bigwedge_n \bigvee_N \tilde{a}^{(n)} < g_N$; ebenso gilt $\bigwedge_n \bigvee_M \tilde{b}^{(n)} < h_M$. Hier-

aus erhält man mit dem Gesetz (2.2.4) die Aussage

$$\bigwedge_n \bigvee_K \tilde{a}^{(n)} + \tilde{b}^{(n)} \leq g_K + h_K \leq a + b,$$

wenn man nämlich als K jeweils das Maximum von N und M wählt sowie die Monotonie der Folgen (g_n) , (h_n) heranzieht. Damit gilt

$$\sup \{\tilde{a}^{(n)} + \tilde{b}^{(n)}\} \leq \sup \{g_n + h_n\} \leq a + b,$$

d.h. $a + b \leq \sup \{g_n + h_n\} \leq a + b$,

woraus die Behauptung folgt.

Nunmehr können wir die noch fehlenden additiven Strukturgesetze in \mathbb{R} beweisen, die Assoziativität und die Umkehrbarkeit.

(2.3) **Satz:** 1. $\bigwedge_{a,b,c} (a + b) + c = a + (b + c)$;

2. $\bigwedge_{a,b} \bigvee_x a + x = b$, nämlich $x = (-a) + b$; die Lösung x ist eindeutig bestimmt, und man setzt sodann $b - a := x$.

Beweis: Zu 1. Es ist nach Definition bzw. (1.5)

$$(a + b) + c = \sup \{\tilde{a}^{(n)} + \tilde{b}^{(n)}\} + \sup \{\tilde{c}^{(n)}\}.$$

Weil die Folgen $(\tilde{a}^{(n)} + \tilde{b}^{(n)})$ sowie $(\tilde{c}^{(n)})$ nach (1.5) monoton wachsen, können wir den Hilfssatz anwenden und erhalten, da die Assoziativität der Addition in $\mathbb{R}^{(e)}$ schon benutzt werden kann,

$$(a + b) + c = \sup \{\tilde{a}^{(n)} + \tilde{b}^{(n)} + \tilde{c}^{(n)}\}.$$

Mit gleichen Argumenten folgt schließlich

$$(a + b) + c = \sup \{\tilde{a}^{(n)}\} + \sup \{\tilde{b}^{(n)} + \tilde{c}^{(n)}\} = a + (b + c).$$

Aussage 2. folgt unter Benutzung des soeben bewiesenen Asso-

ziativgesetzes.

Abschließend zu den additiven Strukturgesetzen sei vermerkt, daß sich (2.2.4) mittels Satz (2.3) sofort zum Monotoniegesetz bez. < verschärfen läßt:

$$\bigwedge_{a,b,c} a < b \Rightarrow a + c < b + c.$$

Wie üblich können der Betrag $|a|$ für Elemente aus \mathbb{R} und mit ihm durch (2.1) Grenzwert, und damit Konvergenz von Folgen definiert werden.

- (2.4) Satz: 1. Jede nach oben (unten) beschränkte monoton wachsende (fallende) Folge ist konvergent, und zwar gegen ihre obere (untere) Grenze.
2. Für konvergente Folgen (g_n) , (h_n) ist auch die Folge $(g_n + h_n)$ konvergent, und zwar gilt
- $$\lim_{n \rightarrow \infty} (g_n + h_n) = \lim_{n \rightarrow \infty} g_n + \lim_{n \rightarrow \infty} h_n.$$

Die aus der Analysis bekannten Beweise für diese im folgenden benötigten Aussagen können übertragen werden, denn sie arbeiten mit den bereitgestellten Begriffen und Aussagen.

Weil nun die Folge $(a^{(n)})$ für $a \geq 0$ monoton wächst bzw. für $a < 0$ monoton fällt und im ersten Fall $a = \sup \{a^{(n)}\}$, dagegen im zweiten $a = \inf \{a^{(n)}\}$ gilt, folgt aus (2.4.1) in beiden Fällen

$$(2.5) \quad a = \lim_{n \rightarrow \infty} a^{(n)}$$

Damit ist das eingangs dieses Abschnitts formulierte Ziel erreicht, denn (2.4.2) liefert

$$\lim_{n \rightarrow \infty} a^{(n)} + \lim_{n \rightarrow \infty} b^{(n)} = \lim_{n \rightarrow \infty} (a^{(n)} + b^{(n)}), \text{ d.h.}$$

$$(2.6) \quad a + b = \lim_{n \rightarrow \infty} (a^{(n)} + b^{(n)}).$$

Anmerkung: Das Arbeiten mit Dezimalbrüchen a ist ein Umgang mit konvergenten Folgen, nämlich den Dezimalbrüchen selbst, aufgefaßt als (definiert durch den Grenzwert der) Folge $(a^{(n)})$. Auch andere Folgen, etwa $(\tilde{a}^{(n)})$, haben denselben Grenzwert a ,

und nach (2.6) hat $(a^{(n)} + b^{(n)})$ denselben Grenzwert wie $((a + b)^{(n)})$, aber dieses Aufsuchen des Grenzwertes von Folgen ist im Grunde jeweils nur das Bestimmen einer weiteren Folge von kanonischen Näherungen, die zur gegebenen Folge um eine Nullfolge differiert. Sieht man so in einem Dezimalbruch selbst lediglich eine spezielle konvergente Folge, dann wird eine gewisse Nähe zu anderen Wegen der Einführung reeller Zahlen erkennbar.

3. Multiplikation in \mathbb{R}

Für die Definition der Multiplikation steht uns der im vorigen Abschnitt bereitgestellte Grenzwertbegriff für Folgen zur Verfügung. Man überlegt sich leicht, daß für Dezimalbrüche a und b die in $\mathbb{R}^{(e)}$ zu verstehende Folge $(a^{(n)} \cdot b^{(n)})$ monoton und beschränkt ist, und zwar für nichtnegative Glieder monoton wächst, im anderen Fall monoton fällt, da die Folgen $(a^{(n)})$, $(b^{(n)})$ monoton sind, und eine (passende) Schranke findet man unter den Zahlen $(a_0 \pm 1) \cdot (b_0 \pm 1)$. Damit ermöglicht Satz (2.4.1) folgende

Definition: Für a und b aus \mathbb{R} sei

$$(3.0) \quad a \cdot b := \lim_{n \rightarrow \infty} (a^{(n)} \cdot b^{(n)}).$$

Es ist sofort zu sehen, daß für Elemente $a^{(m)}$, $b^{(m)}$ aus $\mathbb{R}^{(e)}$ hierdurch kein neues Produkt definiert wurde, weil für $n \geq m$ stets $(a^{(m)(n)} \cdot b^{(m)(n)}) = a^{(m)} \cdot b^{(m)}$ ist. Wir werden daher im folgenden nicht zwischen den \cdot -Zeichen in \mathbb{R} bzw. $\mathbb{R}^{(e)}$ unterscheiden. - Erste Eigenschaften der Multiplikation in \mathbb{R} sind zusammengefaßt in folgendem

$$(3.1) \quad \text{Satz: } \begin{array}{l} 1. \bigwedge_{a, b} a \cdot b = b \cdot a; \\ 2. \bigwedge_a 1 \cdot a = a \wedge (-1) \cdot a = -a; \\ 3. \bigwedge_{a, b} (a \cdot b = 0 \Leftrightarrow a = 0 \vee b = 0); \\ 4. \bigwedge_{a, b, c} (a \leq b \wedge c \geq 0 \Rightarrow a \cdot c \leq b \cdot c). \end{array}$$

Zum Beweis weiterer Strukturgesetze benötigen wir wieder einen Hilfssatz: Seien (g_n) und (h_n) monotone und beschränkte Folgen in \mathbb{R} , und zwar entweder wachsend mit nichtnegativem Grenzwert oder fallend mit nichtpositivem Grenzwert. dann ist $\lim_{n \rightarrow \infty} g_n \cdot \lim_{n \rightarrow \infty} h_n = \lim_{n \rightarrow \infty} (g_n \cdot h_n)$.

Beweis: Mit $a := \lim_{n \rightarrow \infty} g_n$, $b := \lim_{n \rightarrow \infty} h_n$ ist für $a = 0 \vee b = 0$ die Behauptung klar (wenn man bedenkt, daß konvergente Folgen beschränkt sind). Sei nun beispielsweise $a > 0 \wedge b > 0$; für alle (hinreichend großen) n gibt es dann M , so daß für $m > M$ gilt

$$0 \leq a^{(n)} - \frac{1}{10^n} < g_m \leq a \wedge 0 \leq b^{(n)} - \frac{1}{10^n} < h_m \leq b.$$

Hieraus erhält man mit (3.1.4) sowie Gesetzen aus $\mathbb{R}^{(e)}$

$$a^{(n)} \cdot b^{(n)} - \frac{1}{10^n} (a^{(n)} + b^{(n)}) + \frac{1}{10^{2n}} \leq g_m \cdot h_m \leq a \cdot b, \text{ und durch}$$

Grenzwertbildung (unter Benutzung von (2.4.2) und $\lim_{n \rightarrow \infty} \left(\frac{1}{10^n}\right) = 0$)

$$a \cdot b \leq \lim_{n \rightarrow \infty} (g_n \cdot h_n) \leq a \cdot b, \text{ woraus die Behauptung folgt. In den}$$

Fällen $a > 0 \wedge b < 0$ sowie $a < 0 \wedge b < 0$ kann man ganz analog vorgehen.

Mit dem Hilfssatz lassen sich das Assoziativgesetz der Multiplikation und das Distributivgesetz in \mathbb{R} beweisen:

$$(3.2) \text{ Satz: 1. } \bigwedge_{a,b,c} (a \cdot b) \cdot c = a \cdot (b \cdot c);$$

$$2. \bigwedge_{a,b,c} (a + b) \cdot c = a \cdot c + b \cdot c.$$

Im Beweis von 2. ist für $a = 0 \vee b = 0 \vee a + b = 0$ (d.h. $b = -a = (-1) \cdot a$) die Behauptung klar, während man in den weiteren Fällen zur Anwendung des Hilfssatzes neben $(c^{(n)})$ jeweils eine geeignete monotone Folge mit dem Grenzwert $a + b$ benötigt.

Wir können nun auch (3.1.4) zum Monotoniegesetz der Multiplikation bez. < verschärfen:

$$(3.3) \bigwedge_{a,b,c} a < b \wedge c > 0 \rightarrow a \cdot c < b \cdot c,$$

denn $a \cdot c = b \cdot c$, d.h. $(b - a) \cdot c = 0$ kann unter den getroffenen Voraussetzungen nicht eintreten.

Eine Verallgemeinerung des Hilfssatzes beinhaltet der

(3.4) Satz: für konvergente Folgen (g_n) und (h_n) ist auch die Folge $(g_n \cdot h_n)$ konvergent, und zwar gilt

$$\lim_{n \rightarrow \infty} (g_n \cdot h_n) = \lim_{n \rightarrow \infty} g_n \cdot \lim_{n \rightarrow \infty} h_n$$

Der geläufige ϵ -Beweis dieser Regel kann aus der Analysis übertragen werden, wobei darauf hingewiesen sei, daß für die Umformungen u.a. das Distributivgesetz benötigt wird.

Durch die bisherigen Überlegungen konnten wir die Ordnungsrelation $<$ sowie die Rechenoperationen $+$ und \cdot von $\mathbb{R}^{(e)}$ auf \mathbb{R} übertragen und die in $\mathbb{R}^{(e)}$ gültigen Strukturgesetze auch für \mathbb{R} beweisen; außerdem haben wir den Satz von der oberen Grenze (1.4) und einige Regeln über das Rechnen mit Grenzwerten von Folgen erhalten. Es fehlt von den Strukturgesetzen in \mathbb{R} noch die Umkehrbarkeit der Multiplikation. Dies Gesetz der Umkehrbarkeit gilt in $\mathbb{R}^{(e)}$ nicht, ist jedoch eine charakteristische Eigenschaft von \mathbb{Q} , und wir werden es daher mittels der Abbildung D auf \mathbb{R} übertragen. Zuvor stellen wir über D folgende Aussagen bereit:

(3.5) Für $\frac{r}{s}$ aus \mathbb{Q} gilt $D(\frac{r}{s}) = r \cdot D(\frac{1}{s})$ in \mathbb{R} .

(3.6) Aus $\frac{r}{s} < \frac{u}{v}$ in \mathbb{Q} folgt $D(\frac{r}{s}) < D(\frac{u}{v})$ in \mathbb{R} .

(Dabei $r, s, u, v \in \mathbb{Z}$)

Beweis: Zu (3.5). Wir beweisen für $r \geq 0$ und $s > 0$

$$DA(\frac{r}{s}) = r \cdot DA(\frac{1}{s}),$$

woraus sich unmittelbar die allgemeinere Behauptung ergibt.

Ist $a := DA(\frac{r}{s})$, so gilt nach (1.1) in $\mathbb{R}^{(e)}$, und damit in \mathbb{R}

$$r = a^{(n)} \cdot s + \frac{t_n}{10^n} \quad \text{mit } 0 \leq t_n < s \quad (n \in \mathbb{N}).$$

Hieraus folgt durch Grenzwertbildung (mittels (2.4) und (3.4))

$$r = a.s, \text{ d.h. } r = DA\left(\frac{r}{s}\right).s.$$

Insbesondere ist $1 = DA\left(\frac{1}{s}\right).s$, und nach Multiplikation der vorigen Gleichung mit $DA\left(\frac{1}{s}\right)$ erhalten wir die gewünschte Aussage. Zu (3.6). Vorausgesetzt sei o. B. d. A. $s = v > 0$, und demnach $r < u$ (dies in \mathbb{Q} , $\mathbb{R}^{(e)}$, \mathbb{R}). Mit $D\left(\frac{1}{s}\right) > 0$ folgt dann aus (3.5) und (3.3) $D\left(\frac{r}{s}\right) = r.D\left(\frac{1}{s}\right) < u.D\left(\frac{1}{s}\right) = D\left(\frac{u}{s}\right)$.

Zur Umkehrbarkeit der Multiplikation werden wir sogleich zeigen, daß für $a \in \mathbb{R} \setminus \{0\}$ die Gleichung $a.a^{-1} = 1$ stets eine Lösung a^{-1} hat, nämlich

$$a^{-1} := \lim_{n \rightarrow \infty} D\left(\frac{10^n}{10^n a^{(n)}}\right). \checkmark$$

Die Schwierigkeit zur Bestimmung des Inversen von a liegt darin, daß die Inversen der abbrechenden Dezimalbrüche $a^{(n)}$ selbst schon im allgemeinen nicht abbrechend sind. Der rechtsstehende Grenzwert existiert nach Satz (2.4.1), denn für

$a > 0$ gilt $0 < a^{(n)} \leq a^{(n+1)}$ in $\mathbb{R}^{(e)}$, und damit in \mathbb{Q}

$$0 < \frac{10^n a^{(n)}}{10^n} \leq \frac{10^{n+1} a^{(n+1)}}{10^{n+1}}, \text{ woraus in } \mathbb{Q}$$

$$\frac{10^n}{10^n a^{(n)}} \geq \frac{10^{n+1}}{10^{n+1} a^{(n+1)}} > 0 \text{ folgt. Hieraus erhält man mit}$$

(3.6), daß die Folge $D\left(\frac{10^n}{10^n a^{(n)}}\right)$ in \mathbb{R} monoton fallend, nach unten beschränkt, und daher konvergent ist. Für $a < 0$ handelt es sich um eine monoton wachsende Folge negativer Glieder, die ebenfalls konvergiert. - Für abbrechende Dezimalbrüche $a^{(m)}$ erhält man übrigens

$$(a^{(m)})^{-1} = D\left(\frac{10^m}{10^m a^{(m)}}\right),$$

da die zu betrachtende Folge für große n konstant ist.

(3.7) Satz: $\bigwedge_{a \neq 0, b} \bigvee_x a.x = b$, nämlich $x = a^{-1}.b$; die Lösung x

ist eindeutig bestimmt, und man setzt sodann

\checkmark n so groß, daß Nenner nicht verschwinden.

$$\frac{b}{a} := x.$$

Die für den Beweis wesentliche Aussage, $a \cdot a^{-1} = 1$, folgt unmittelbar mittels (3.4) und (3.5).

Eine weitere Möglichkeit zur Berechnung von Quotienten ergibt sich aus dem

(3.8) Satz: Für konvergente Folgen (g_n) und (h_n) mit $\lim_{n \rightarrow \infty} g_n \neq 0$

$$\text{gilt } 1. \lim_{n \rightarrow \infty} \frac{1}{g_n} = \frac{1}{\lim_{n \rightarrow \infty} g_n}$$

$$2. \lim_{n \rightarrow \infty} \frac{h_n}{g_n} = \frac{\lim_{n \rightarrow \infty} h_n}{\lim_{n \rightarrow \infty} g_n}$$

Der ε -Beweis für 1. kann übertragen werden, die zweite Aussage folgt sodann in Verbindung mit Satz (3.4).

Insbesondere hat man jetzt

$$\begin{aligned} \frac{b}{a} &= \frac{\lim_{n \rightarrow \infty} b^{(n)}}{\lim_{n \rightarrow \infty} a^{(n)}} = \lim_{n \rightarrow \infty} \frac{b^{(n)}}{a^{(n)}} = \lim_{n \rightarrow \infty} (b^{(n)} \cdot D(\frac{10^n}{10^n \cdot a^{(n)}})) \\ &= \lim_{n \rightarrow \infty} D(\frac{10^n \cdot b^{(n)}}{10^n \cdot a^{(n)}}) \quad ; \text{ dies besagt:} \end{aligned}$$

(3.9) Zu reellen Zahlen a, b ($a \neq 0$) erhält man durch Anwendung des Divisionsalgorithmus auf die rationalen Zahlen

$$\frac{10^n b^{(n)}}{10^n a^{(n)}} \quad (n \in \mathbb{N}) \text{ Glieder einer Folge von Dezimalbrüchen}$$

mit dem Grenzwert $\frac{b}{a}$.

Berechnet man von $D(\frac{10^n b^{(n)}}{10^n a^{(n)}})$ für jeweils z.B. die n -te kanonische Näherung ($n = 0, 1, 2, \dots$), so erhält man eine Folge abbrechender Dezimalbrüche, die - wie nun leicht zu folgern - ebenfalls den Grenzwert $\frac{b}{a}$ hat.

Literatur:

/1/ Burosch, G. Vorlesungen zum Grundkurs Mathematik, Rostock 1971/72 (nicht veröffentlicht)

- /2/ Flachsmeyster, J. und Terpe, E. Über die verschiedenen Wege der Einführung der reellen Zahlen
Mitteilungen der MGDDDR, Heft 1, 1968,
57 - 71
- /3/ Holland, G. Ein Vorschlag zur Einführung der reellen Zahlen als Dezimalbrüche, Math.-Phys.
Semesterber. XVIII (1971), 87 - 110
- /4/ Lemke, H. und Stoye, W. Konstruktion des Körpers der reellen Zahlen
Mathematik in der Schule 13 1975, H. 1,
10 - 19
- /5/ Mathematik, Lehrbuch für Klasse 9
Berlin, Volk und Wissen, 1970
- /6/ Wisliceny, J. Grundbegriffe der Mathematik II (MfL Bd.2)
Berlin, Deutscher Verlag der Wissenschaften, 1974

eingegangen am: 1. 2. 1977

Anschrift des Verfassers:

Dr. paed. Klaus-Dieter Drews
Wilhelm-Pieck-Universität Rostock
Sektion Mathematik
DDR 25 Rostock
Universitätsplatz 1

