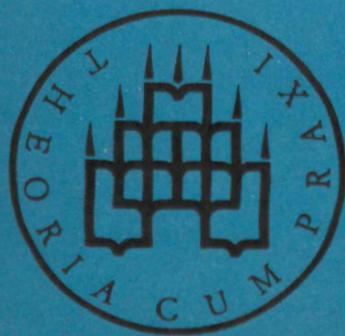


Rostocker

Mathematisches Kolloquium

Heft 8



WILHELM-PIECK-  
UNIVERSITÄT ROSTOCK  
1978



ROSTOCKER MATHEMATISCHES KOLLOQUIUM

Heft 8

1978

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 Sehrage  
Fachredakteur: Doz. Dr. sc. nat. Gerhard Maeß, Sektion  
Mathematik**

**Herausgegeben von der Wilhelm-Pieck-Universität Rostock unter  
Genehmigungs-Nr. C 221/78**

**Druck: Ostsee-Druck Rostock, Werk II**

## Inhalt

	<u>Seite</u>	
Steffen, Günther	Irreduzible zyklische Matrixgruppen über dem Restklassenring mod $p^k$	5
Mumm, Harald; Wildenhain, Günther	Zur Darstellung der Lösung allgemeiner elliptischer Randwertprobleme	31
Clasen, Ute; Wildenhain, Günther	Approximation durch biharmonische Potentiale	47
Albrand, Hans-Jürgen	Diskrete lineare Approximation und zyklische Einzelschrittverfahren	57
Rasch, Dieter; Herrendörfer, Günter	Über die Anzahl elementarer BUB in eingeschränkten $(v,k)$ -Familien	71
Fröhlich, Rolf	Eine Verallgemeinerung der BACKUS- Systeme	83
Gruël, Jürgen; Merbeth, Günter	Eine Betrachtung zur formalen Be- schreibung von Programmiersprachen und Sprachverarbeitungssystemen	91
Riedewald, Günter	A method for construction of compilers and compiler-generating systems based on grammars of syntac- tical functions	109
Deth, Reinhard	Parallele Bearbeitung von Prozessen	123



Irreduzible zyklische Matrixgruppen über dem Restklassenring mod  $p^k$

---

Herrn Prof. G. Pazderski zum 50. Geburtstag  
aus Dankbarkeit gewidmet

Einleitung

Jeder Hauptfaktor einer auflösbaren Gruppe ist elementarabelsch von einer gewissen Ordnung  $p^n$  ( $p$  Primzahl,  $n \geq 1$ ). Additiv geschrieben können wir ihn als  $GF(p)$ -Modul mit  $n$  Basiselementen auffassen. Die Automorphismengruppe dieses Hauptfaktors ist dann isomorph zur  $GL(n, p)$ . Bezeichne  $M|N$  einen Hauptfaktor einer auflösbaren Gruppe  $G$ , dann wird jedem  $g \in G$  gemäß  $mN \rightarrow g^{-1}mgN$  ( $m \in M$ ) ein Automorphismus von  $M|N$  zugeordnet. Diese Zuordnung ist ein Homomorphismus von  $G$  in die Automorphismengruppe von  $M|N$ . Eine auflösbare Gruppe erfährt auf diese Weise auf jedem seiner Hauptfaktoren eine irreduzible Matrixdarstellung über  $GF(p)$ .

Nun können in einer auflösbaren Gruppe auch Faktorgruppen vom Typ  $(p^k, p^k, \dots, p^k)$  ( $p$  Primzahl,  $k \geq 1$ ) vorkommen. Auf solchen Faktorgruppen würde  $G$  dann eine Matrixdarstellung über dem Restklassenring mod  $p^k$  erfahren. Somit sind Aussagen über Matrixdarstellungen über Restklassenringen mod  $p^k$  für Strukturuntersuchungen abstrakter auflösbarer Gruppen von Bedeutung.

In dieser Arbeit wollen wir uns mit gewissen irreduziblen zyklischen Matrixgruppen über dem Restklassenring mod  $p^k$  befassen. Dabei geht es um die Bestimmung bis auf Äquivalenz aller derjenigen irreduziblen zyklischen  $n \times n$ -Matrixgruppen über dem Restklassenring mod  $p^k$ , die bei Reduktion ihrer Koeffizienten mod  $p$  irreduzibel bleiben. Teil 1 hat vorbereitenden Charakter. In ihm wird einiges über Darstellungen über Ringen bereitgestellt. Die wichtigste Aussage in diesem Teil, die dann für den Beweis des Satzes im Teil 2 von Bedeutung ist, wird im

**Lemma 1.3** formuliert. Sie beinhaltet eine Verallgemeinerung der bekannten Tatsache, daß bei einer Divisionsalgebra über einem Körper die einzige irreduzible Darstellung durch die reguläre Darstellung gegeben wird. Die Hauptaussage dieser Arbeit ist die, daß bis auf Äquivalenz genau eine irreduzible zyklische  $n \times n$ -Matrixgruppe der Ordnung  $r$  über dem Restklassenring mod  $p^k$  existiert, die bei Reduktion mod  $p$  ebenfalls irreduzibel von der Ordnung  $r$  ist. Zwischen  $r$  und  $n$  besteht dabei die Beziehung, daß  $n$  die kleinste natürliche Zahl mit  $r \mid p^n - 1$  ist.

An dieser Stelle möchte ich Herrn Prof. G. Pazderaki für seine freundliche Unterstützung bei der Fertigstellung dieser Arbeit recht herzlich danken.

### Bezeichnungen

$\mathbb{Z}$  = Menge der ganzen Zahlen;  $p$  sei Primzahl;  $R_k$  = Restklassenring mod  $p^k$ , wobei  $k \geq 1$  eine natürliche Zahl ist;  $R_k[x]$  = Polynomring in einer Erzeugenden über  $R_k$ ;  $[a]_k$  bezeichne die Restklasse mod  $p^k$ , in der die ganze Zahl  $a$  liegt;  $p^1 \mid f(x)$  bedeutet, daß die Restklasse  $[p^1]_k$  in jedem Koeffizienten von  $f(x) \in R_k[x]$  als Faktor enthalten ist ( $1 \geq 1$  natürliche Zahl);  $p^1 \nmid f(x)$  bedeutet, daß  $p^1 \mid f(x)$  nicht gilt;  $[a(x)]_{f(x)}$  bezeichne diejenige Restklasse von  $R_k[x] \mid (f(x))$ , in der  $a(x) \in R_k[x]$  liegt;  $E_n$  = Einheitsmatrix vom Grade  $n$  über dem jeweils betrachteten Ring; für eine Matrix  $C$  über  $R_k$  ( $k \geq 1$ ) bedeute  $p^1 \mid C$  ( $1 \geq 1$ ), daß  $[p^1]_k$  jeden Koeffizienten von  $C$  teilt.

### 1. Darstellungen über Ringen

Es sei  $F$  ein kommutativer Ring mit Einselement  $1$  und  $R$  eine  $F$ -Algebra, d. h., es sind neben den Ringaxiomen für  $R$  die weiteren Bedingungen erfüllt:

- |   |                               |
|---|-------------------------------|
| 1. Zu $f \in F$ und $r \in R$ existiert ein eindeutig bestimmtes $fr \in R$ | 4. $f(r+r') = fr + fr'$       |
| 2. $(f+f')r = fr + f'r$   | 5. $f(rr') = (fr)r' = r(fr')$ |
| 3. $(ff')r = f(f'r)$  | 6. $\underline{1}r = r$       |

für  $f, f' \in F$  und  $r, r' \in R$ .

Wir wollen nun die Begriffe  $F$ - $R$ -Modul bzw. Darstellung von  $R$  über  $F$  einführen und die Gleichwertigkeit von Matrixdarstellungen und  $F$ - $R$ -Moduln mit endlicher Basis über  $F$  zeigen.

**Definition 1.1:**  $M$  heiÙe  $F$ - $R$ -Modul, wenn  $M$  ein  $F$ -Links- sowie ein  $R$ -Rechtsmodul ist und auÙerdem gilt:

$$\begin{aligned} m(fr) &= f(mr) = (fm)r \\ \underline{1}m &= me = m \end{aligned}$$

für  $f \in F$ ,  $r \in R$ ,  $m \in M$  und  $e$  Einselement von  $R$ .

Offensichtlich bilden die  $F$ -Endomorphismen bezüglich der durch

$$m(\tau + \xi) = m\tau + m\xi, \quad m(\tau\xi) = (m\tau)\xi \quad (m \in M)$$

für beliebige  $F$ -Endomorphismen  $\tau$  und  $\xi$  definierten Addition und Multiplikation einen Ring.

**Definition 1.2:** Ein  $F$ -Operatorhomomorphismus von  $R$  in den Endomorphismenring aller  $F$ -Endomorphismen eines  $F$ - $R$ -Moduls heiÙt Darstellung von  $R$  über  $F$ .

Falls im  $F$ - $R$ -Modul  $M$  eine endliche Basis der Länge  $n$  über  $F$  existiert, so ist der Ring aller  $F$ -Endomorphismen von  $M$  isomorph zur vollen Matrixalgebra aller  $n \times n$ -Matrizen über  $F$ . Darstellungen von  $R$  bezüglich dieser Basis werden dann auch genauer Matrixdarstellungen des Grades  $n$  genannt.

Ist  $\{u_1, \dots, u_n\}$  eine endliche Basis von  $M$  über  $F$ , so wird jedem  $r \in R$  eine  $n \times n$ -Matrix  $\|\alpha_{ij}^{(r)}\| = \sigma(r)$  über  $F$  vermöge

$$u_1 r = \sum_{j=1}^n \alpha_{1j}^{(r)} u_j \text{ zugeordnet. Offensichtlich ist die Zuord-}$$

nung  $r \rightarrow \sigma(r)$  ein  $F$ -Operatorhomomorphismus von  $R$  in die Menge aller  $n \times n$ -Matrizen über  $F$ , also eine Matrixdarstellung. Mit anderen Worten, zu jedem  $F$ - $R$ -Modul mit endlicher Basis gibt es eine zugehörige Matrixdarstellung von  $R$  über  $F$ .

Ist umgekehrt eine Matrixdarstellung  $\sigma(r) = \|\alpha_{ij}(r)\|$  von  $R$  des Grades  $n$  über  $F$  vorgegeben, so können wir aus einem Modul  $M$  über  $F$  mit der Basis  $\{u_1, \dots, u_n\}$  durch die Festsetzung

$$u_i r := \sum_{j=1}^n \alpha_{ij}(r) u_j, \quad \left( \sum_{i=1}^n \gamma_i u_i \right) r := \sum_{i=1}^n \gamma_i (u_i r) \quad \text{mit } \gamma_i \in F \text{ für}$$

$i = 1, \dots, n$  einen  $F$ - $R$ -Modul machen, zu dem die Matrixdarstellung  $\sigma$  gehört.

Wir haben also eine Beziehung zwischen den  $F$ - $R$ -Moduln, in denen eine endliche Basis existiert, und den Matrixdarstellungen von  $R$  über  $F$ .

Im weiteren soll eine Äquivalenzrelation für Matrixdarstellungen von  $R$  über  $F$  eingeführt werden, die mit der Operatorisomorphie der entsprechenden  $F$ - $R$ -Moduln gleichbedeutend ist.

**Definition 1.3:** Wir nennen zwei Matrixdarstellungen  $\mu$  bzw.  $\nu$  des Grades  $m$  bzw.  $n$  von  $R$  über  $F$  äquivalent, wenn eine  $n \times m$ -Matrix  $T$  über  $F$  und eine  $m \times n$ -Matrix  $T'$  über  $F$  mit  $TT' = E_n$  und  $T'T = E_m$  existieren, so daß  $T\mu(r)T' = \nu(r)$  für alle  $r \in R$  gilt.  
Bezeichnung:  $\mu \sim \nu$ .

Es ist sofort zu sehen, daß  $\sim$  eine Äquivalenzrelation ist.

**Lemma 1.1:** Seien  $\mu$  bzw.  $\nu$  Matrixdarstellungen von  $R$  über  $F$  und  $M$  bzw.  $N$  mit den endlichen Basen  $u$  und  $v$  die entsprechenden  $F$ - $R$ -Moduln. Genau dann, wenn ein  $F$ - $R$ -Operatorisomorphismus  $\tau$  von  $M$  auf  $N$  existiert, sind  $\mu$  und  $\nu$  äquivalent.

**Beweis:** Die endlichen Basen  $u$  und  $v$  von  $M$  bzw.  $N$  mögen die Längen  $m$  bzw.  $n$  haben, d. h.

$$u = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \quad \text{und} \quad v = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} .$$

Wir wollen zuerst davon ausgehen, daß  $M$  zu  $N$   $F$ - $R$ -operatorisomorph vermöge  $\tau$  ist. Dann gilt sicher

$$u \tau = Tv$$

mit einer gewissen  $m \times n$ -Matrix  $T$  über  $F$ . Bezeichne  $\tau^{-1}$  die Umkehrabbildung von  $\tau$ . Man kann sich davon überzeugen, daß

$\tau^{-1}$  ein  $F$ - $R$ -Operatorisomorphismus von  $N$  auf  $M$  ist und es gilt  $v \tau^{-1} = T'u$  mit einer gewissen  $n \times m$ -Matrix  $T'$  über  $F$ . Es folgt

$$u = (u \tau) \tau^{-1} = (Tv) \tau^{-1} = T(v \tau^{-1}) = T(T'u) = (TT')u$$

und wegen der eindeutigen Darstellbarkeit der Elemente von  $M$  durch die Basis  $u$  muß  $TT' = E_m$  sein. Analog erkennt man von der Basis  $v$  ausgehend die Gültigkeit von  $T'T = E_n$ . Weiter haben wir

$$\begin{aligned} (T \psi(r))v &= T(\psi(r)v) = T(vr) = (Tv)r = (u \tau)r = (ur)\tau \\ &= (\mu(r)u)\tau = \mu(r)(u \tau) = \mu(r)(Tv) = (\mu(r)T)v \end{aligned}$$

und da  $v$  Basis war, ist  $T \psi(r) = \mu(r)T$  für beliebige  $r \in R$ . Das heißt aber  $T \psi(r)T' = \mu(r)$  und  $\mu$  und  $\psi$  sind äquivalent.

Sei nun umgekehrt  $\mu$  zu  $\psi$  äquivalent und  $\mu(r) = T \psi(r)T'$  für beliebige  $r \in R$  mit einer  $m \times n$ -Matrix  $T$  und einer  $n \times m$ -Matrix  $T'$ , für die  $TT' = E_m$  und  $T'T = E_n$  gilt. Wir definieren die Abbil-

dung  $\tau$  von  $M$  auf  $N$  durch  $x \tau := \sum_{i=1}^m f_i(u_i \tau)$  für beliebige

$x = \sum_{i=1}^m f_i u_i$  aus  $M$  mit  $f_i \in F$ . Dabei ist  $u_i \tau$  aus  $u \tau := Tv$  zu

entnehmen. Wir wollen nun noch die Abbildung  $\tau'$  von  $N$  auf  $M$

durch  $y \tau' := \sum_{i=1}^n g_i(v_i \tau')$  für  $y = \sum_{i=1}^n g_i v_i$  mit  $g_i \in F$  und

$v \tau' := T'u$  definieren.

Unser Ziel ist es, nachzuweisen, daß  $\tau$  ein F-R-Operatorisomorphismus von M auf N ist. Es ist leicht einzusehen, daß  $\tau$  eine

eindeutige Abbildung von M in N ist. Für  $x = \sum_{i=1}^m f_i u_i$

aus M wollen wir kurz  $x = \bar{F}u$  mit  $\bar{F} = (f_1, \dots, f_m)$  schreiben.

Es folgt

$$\begin{aligned} (x\tau)\tau &= ((\bar{F}u)\tau)\tau = (\bar{F}(u\tau))\tau = ((\bar{F}\wedge(r))u)\tau \\ &= (\bar{F}\wedge(r))(u\tau) = (\bar{F}\wedge(r)T)v = (\bar{F}T\varphi(r))v \\ &= \bar{F}T(v\tau) = \bar{F}(T\tau)r = (\bar{F}(u\tau))r = (x\tau)r \end{aligned}$$

Ohne Schwierigkeiten ist zu sehen, daß bei  $\tau$  die Summenübertragung und  $(fx)\tau = f(x\tau)$  für beliebige  $x \in M$  gilt. Damit ist  $\tau$  ein F-R-Operatorhomomorphismus von M in N.

Analog läßt sich zeigen, daß  $\tau'$  auch ein F-R-Operatorhomomorphismus von N in M ist. Weiter gilt

$$\begin{aligned} (x\tau)\tau' &= (\bar{F}(u\tau))\tau' = (\bar{F}T\tau)\tau' = \bar{F}T(\tau\tau') \\ &= \bar{F}T\tau'u = \bar{F}E_m u = \bar{F}u = x \end{aligned}$$

Auf die gleiche Weise zeigt sich, daß  $\tau'\tau$  die identische Abbildung ist. Das heißt,  $\tau'$  ist die Umkehrabbildung zu  $\tau$  und  $\tau$  die zu  $\tau'$ . Daraus und aus der Tatsache, daß  $\tau'$  eine eindeutige Abbildung von N in M ist, folgt nun, daß  $\tau$  ein F-R-Operatorisomorphismus von M auf N ist.

Lemma 1.2: Sei T eine  $n \times m$ -Matrix über dem kommutativen Ring F mit Einselement  $\underline{1}$  und T' eine  $m \times n$ -Matrix über F mit

$TT' = E_n$ . Dann folgt  $n \leq m$ .

Beweis: Wir wollen die Annahme  $n > m$  zu einem Widerspruch führen. Dann besitzt T mindestens  $m+1$  Zeilen und T' besteht aus mindestens  $m+1$  Spalten. Die ersten m Zeilen von T fassen wir zur Matrix A und die ersten m Spalten von T' zur Matrix A' zusammen. Aus der zweiten bis  $(m+1)$ -ten Zeile von T soll die quadratische Matrix B und aus der zweiten bis  $(m+1)$ -ten Spalte von T' die quadratische Matrix B' bestehen. Es folgt nun

$$AA' = E_m \quad ,$$

$$BB' = E_m \quad ,$$

$$AB' = \begin{pmatrix} 0 & \dots & \dots & \dots & 0 \\ \underline{1} & 0 & \dots & \dots & 0 \\ 0 & \underline{1} & 0 & \dots & 0 \\ \vdots & \cdot & & & \\ \vdots & & \cdot & & \\ 0 & 0 & \dots & \dots & \underline{1} & 0 \end{pmatrix} .$$

Da der Determinantenmultiplikationssatz über kommutativen Ringen mit Einselement gilt (siehe etwa Bourbaki /1/, S. 78), haben wir

$$(1.1) \quad \begin{aligned} \det A \det A' &= \det (AA') = \underline{1} \quad , \\ \det B \det B' &= \det (BB') = \underline{1} \end{aligned}$$

und weiterhin

$$\begin{aligned} \det B' &= (\det A \det A') \det B' \\ &= (\det A' \det A) \det B' \\ &= \det A' \det (AB') = 0 \quad , \end{aligned}$$

dabei bezeichne  $\det X$  die Determinante der quadratischen Matrix  $X$  über  $F$ . Diese Beziehung ist aber ein Widerspruch zur Gleichung (1.1).

Die Matrizen  $T$  und  $T'$  aus der Definition 1.3 müssen Lemma 1.2 zufolge also quadratische Matrizen sein, d. h., es können nur Matrixdarstellungen gleichen Grades zueinander äquivalent sein.  $T'$  ist dann die inverse Matrix zu  $T$ . Dieses zieht aber unter Berücksichtigung von Lemma 1.1 nach sich, daß alle endlichen Basen eines  $F$ - $R$ -Moduls die gleiche Länge haben. Für einen  $F$ - $R$ -Modul  $M$ , in dem alle Basen endlich sind, kann man wie üblich die Dimension erklären, die wir mit  $\dim M$  bezeichnen.

Definition 1.4: Eine Matrixdarstellung  $\mathcal{C}$  des Grades  $m$  von  $R$  über  $F$  heie reduzibel, wenn eine zu  $\mathcal{C}$  äquivalente Matrixdarstellung  $\mu$  existiert mit

$$\mu(x) = \begin{pmatrix} \mu_1(x) & 0 \\ \mu_2(x) & \mu_3(x) \end{pmatrix} \quad ,$$

wobei  $\mu_1(r)$  bzw.  $\mu_2(r)$  für alle  $r \in R$  Matrizen eines Formates  $m_1 \times m_1$  bzw.  $(m-m_1) \times (m-m_1)$  mit  $1 \leq m_1 < m$  sind.

Andernfalls heie  $\mathcal{C}$  irreduzibel.

**Definition 1.5:** Sei  $M$  ein  $F$ - $R$ -Modul mit einer endlichen Basis ber  $F$ .  $M$  wollen wir reduzibel nennen, wenn  $M$  einen eigentlichen  $F$ - $R$ -zulssigen Teilmodul hat, der eine Basis ber  $F$  besitzt, die sich zu einer Basis von  $M$  ber  $F$  ergnzen lsst. Ansonsten heie  $M$  irreduzibel.

Man sieht sofort, da Reduzibilitt invariant gegenber Operatorisomorphie ist.

**Lemma 1.3:** Eine Matrixdarstellung  $\mathcal{C}$  des Ringes  $R$  mit endlicher Basis ber  $F$  ist genau dann reduzibel, wenn der zugehrige  $F$ - $R$ -Modul  $M$  reduzibel ist.

**Beweis:** Sei  $M'$  ein eigentlicher  $F$ - $R$ -zulssiger Teilmodul von  $M$ , dessen Basis ber  $F$  sich zu einer Basis von  $M$  ergnzen lsst. Die Matrixdarstellung  $\mu(r)$  bezglich einer solchen Basis hat dann die in der Definition 1.4 angegebene Gestalt. Da  $M'$  zu sich selbst operatorisomorph ist, gilt wegen Lemma 1.1  $\mu \sim \mathcal{C}$  und damit ist  $\mathcal{C}$  reduzibel.

Sei nun  $\mathcal{C}$  reduzibel und  $M$  der zugehrige  $F$ - $R$ -Modul. Dann existiert eine zu  $\mathcal{C}$  quivalente Matrixdarstellung  $\mathcal{C}$  der Gestalt, wie sie in der Definition 1.4 angegeben wurde.  $N$  sei der  $\mathcal{C}$  entsprechende  $F$ - $R$ -Modul und  $b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$  die zu  $\mathcal{C}$  gehrige Basis.

Dann gilt also

$$\begin{pmatrix} b_1 r \\ \vdots \\ b_m r \\ b_{m+1} r \\ \vdots \\ b_n r \end{pmatrix} = \begin{pmatrix} f_{11} & \dots & f_{1m} \\ \vdots & & \vdots \\ f_{m1} & \dots & f_{mm} \\ f_{m+1,1} & \dots & f_{m+1,n} \\ \vdots & & \vdots \\ f_{n1} & \dots & f_{nn} \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_m \\ b_{m+1} \\ \vdots \\ b_n \end{pmatrix},$$

wobei  $1 \leq m < n$  und  $f_{ij} \in F$  ( $i, j = 1, \dots, n$ ).

Offensichtlich spannen die ersten  $m$  Basiselemente einen eigentlichen  $F$ - $R$ -zulässigen Teilmodul von  $N$  auf. Nach Definition 1.5 ist  $N$  und damit auch der zu ihm  $F$ - $R$ -operatorisomorphe Modul  $M$  reduzibel.

**Definition 1.6:**  $R$  sei eine Algebra mit endlicher Basis über  $F$ . Dann ist  $R$  bezüglich der Addition ein  $F$ - $R$ -Modul (durch die Multiplikation in  $R$  sei die Operatoranwendung von rechts gegeben). Eine diesem Modul entsprechende Matrixdarstellung von  $R$  über  $F$  bezüglich einer vorgegebenen Basis möge reguläre Matrixdarstellung heißen.

**Lemma 1.4:**  $F$  bestehe aus endlich vielen Elementen.  $R$  sei Algebra mit Einselement und einer endlichen Basis über  $F$ , deren reguläre Darstellung mit  $\varrho$  bezeichnet werde. Es möge für ein beliebiges  $r \in R$  ( $r \neq 0$ ) ein invertierbares  $r_0 \in R$  und ein  $f \in F$  existieren mit  $r = fr_0$ . Für jede von der Nulldarstellung verschiedene Matrixdarstellung  $\mu$  von  $R$  mit  $\text{grad } \mu \leq \text{grad } \varrho$  folgt dann, daß  $\mu$  irreduzibel und  $\mu \sim \varrho$  ist.

**Beweis:** Sei  $M$  der entsprechende  $F$ - $R$ -Modul für  $\mu$ . Da  $\mu$  von der Nulldarstellung verschieden ist, muß  $M \neq \{0\}$  sein. Die Dimension von  $M$  über  $F$  stimmt mit  $\text{grad } \mu$  überein, also

$$(1.2) \quad \dim M \leq \text{grad } \varrho.$$

Sei  $a \neq 0$  ein Basiselement von  $M$  und  $\{r_1, \dots, r_n\}$  eine Basis von  $R$  über  $F$ , wobei  $n = \text{grad } \varrho$ . Um zu zeigen, daß  $aR = M$  ist, wollen wir den Nachweis erbringen, daß  $\{ar_1, \dots, ar_n\}$  ein linear unabhängiges System von Elementen aus  $M$  über  $F$  ist. Sei

$$\sum_{i=1}^n f_i(ar_i) = 0 \text{ mit } f_i \in F (i = 1, \dots, n). \text{ Dann folgt}$$

$$0 = \sum_{i=1}^n f_i(ar_i) = \sum_{i=1}^n a(f_i r_i) = a \sum_{i=1}^n f_i r_i = ar$$

mit  $r = \sum_{i=1}^n f_i r_i \in R$ . Falls  $r \neq 0$  wäre, würde aus der Voraussetzung dieses Lemmas folgen, daß ein  $f \in F$  und ein invertierbares  $r_0 \in R$  existiert mit  $r = fr_0$ . Das hieße aber

$$0 = ar = a(fr_0) = f(ar_0) ,$$

also gälte

$$0 = ((fa)r_0)r_0^{-1} = (fa)(r_0r_0^{-1}) = fa$$

und folglich  $f = 0$ , da  $a$  Basiselement ist. Daraus ergäbe sich nun  $r = fr_0 = 0$ .

Es folgt jetzt  $\sum_{i=1}^n f_i r_i = 0$ . Da  $\{r_1, \dots, r_n\}$  Basis von  $R$  über  $F$  war, muß  $f_i = 0$  für alle  $i = 1, \dots, n$  gelten. Somit sind die  $\{ar_1, \dots, ar_n\}$  linear unabhängig über  $F$  und es gilt  $|aR| \geq |F|^n$ . Wegen (1.2) können höchstens  $\text{grad } \xi = n$  Basiselemente in  $M$  existieren, d. h.  $|M| \leq |F|^n$ . Infolgedessen ist  $|aR| = |M|$ , also  $aR = M$ . Zudem sieht man

$$(1.3) \quad \dim M = \dim R.$$

Da wir für  $a$  ein beliebiges Basiselement gewählt haben, folgt auch, daß  $M$  und nach Lemma 1.3 auch  $\mathcal{M}$  irreduzibel ist.

Wir wollen nun die Abbildung  $\overline{\pi} : R \rightarrow M$ , die definiert ist durch  $r \overline{\pi} = ar$  ( $r \in R$ ), betrachten. Für diese Abbildung läßt sich ohne weiteres die Operatorhomomorphieeigenschaft von  $R$  als  $F$ - $R$ -Modul auf den  $F$ - $R$ -Modul  $M$  zeigen. Wegen der endlichen Mächtigkeit von  $R$  bzw.  $M$ , denn beides sind Moduln mit endlicher Basis über dem endlichen Ring  $F$ , und der Tatsache, daß nach (1.3)  $M$  und  $R$  die gleichen Dimensionen über  $F$  haben, folgt nun, daß  $\overline{\pi}$  ein  $F$ - $R$ -Operatorisomorphismus und nach Lemma 1.1  $\mathcal{M} \sim \mathcal{M}$  ist.

Insbesondere geht aus Lemma 1.3 hervor, daß die reguläre Matrixdarstellung von  $R$  über  $F$  irreduzibel ist.

## 2. Irreduzible zyklische Matrixgruppen über $R_k$

**Definition 2.1:**  $f(x) \in R_k[x]$  sei ein Polynom, dessen höchster Koeffizient eine prime Restklasse mod  $p^k$  ist.  $f(x)$  soll **reduzibel** heißen, wenn ein  $g(x) \in R_k[x]$  mit einer primen Restklasse als höchstem Koeffizienten und  $0 < \text{grad } g(x) < \text{grad } f(x)$  existiert, so daß  $g(x) | f(x)$ . Ansonsten heiße  $f(x)$  **irreduzibel**.

Die folgenden Tatsachen sollen als bekannt vorausgesetzt werden:

(2.1) Sei  $\Lambda$  ein Ring mit Einselement aus endlich vielen Elementen und  $\eta \in \Lambda$  weder Nullteiler noch Null. Dann ist  $\eta$  invertierbar in  $\Lambda$ .

(2.2) Sei  $\Lambda$  ein Ring mit Einselement und  $f(x), g(x) \in \Lambda[x]$ . Falls der höchste Koeffizient von  $g(x)$  invertierbar ist, dann existieren  $q(x), r(x) \in \Lambda[x]$  mit

$$f(x) = g(x)q(x) + r(x),$$

wobei  $\text{grad } r(x) < \text{grad } g(x)$  oder  $r(x) = 0$  ist (v. d. Waerden /3/, S. 52).

(2.3) Sei  $x^r - 1 \in \text{GF}(p)[x]$  und  $r$  eine natürliche Zahl mit  $p \nmid r$ . Dann ist jedes irreduzible Polynom, welches in der Zerlegung von  $x^r - 1$  auftritt, genau einmal in dieser enthalten (v. d. Waerden /3/, S. 177).

(2.4) Sei  $\langle A \rangle$  eine irreduzible Matrixgruppe der Ordnung  $r$  über  $\text{GF}(p)$ . Dann gilt nach (Pasderaki /2/, S. 14) für die lineare Hülle  $R_1[A]$  von  $\langle A \rangle$  über  $\text{GF}(p)$

$$R_1[A] \cong \text{GF}(p^n),$$

wobei  $n$  die kleinste natürliche Zahl mit  $r \mid p^n - 1$  ist. Außerdem ist  $n$  der Grad von  $A$ .

Nun existiert ein zu  $A \in R_1[A]$  gehöriges Minimalpolynom  $f(x)$  über  $\text{GF}(p)$ , welches eindeutig als irreduzibles normiertes Polynom mit  $A$  als Nullstelle bestimmt ist. Die Körpererweiterung  $R_1[A] \mid R_1$  des Grades  $n$  ist dann isomorph zu  $R_1[x] \mid (f(x))$  und somit  $\text{grad } f(x) = n$ .

Der Ring  $R$  werde bezüglich  $\mathcal{G}$  homomorph auf  $R'$  abgebildet. Dann läßt sich offensichtlich  $\mathcal{G}$  eindeutig zu einem Homomorphismus  $\tau$  von  $R[x]$  auf  $R'[y]$  derart fortsetzen, daß

$$\tau : f(x) = \sum_1 a_i x^i \rightarrow \sum_1 a_i' y^i = \tau(f(x)) = f(x)'$$

mit  $a_i' = a_i^{\mathcal{G}}$ . Wie man unmittelbar sieht, ist die Abbildung

$[a]_k \rightarrow [a]_1$  ( $a \in \mathbb{Z}$ ) ein Homomorphismus von  $R_k$  auf  $R_1$ . Nach dem oben Gesagten wird dann durch

$$\tau : f(x) = \sum_1^r [a_i]_k x^i \rightarrow \sum_1^r [a_i]_1 y^i = \bar{f}(y) = f(x)^*$$

$R_k[x]$  homomorph auf  $R_1[y]$  abgebildet. Man rechnet leicht nach, daß die Abbildung

$$(2.5) \quad [a(x)]_{f(x)} \rightarrow [a(x)^*]_{f(x)^*} =: [a(x)]_{f(x)}^*$$

einen Homomorphismus von  $R_k[x] | (f(x))$  auf  $R_1[y] | (f(y))$  liefert. Entsprechend ist für einen Matrixring  $\Lambda$  über  $R_k$  die Abbildung

$$(2.6) \quad A = \|[a_{ij}]_k\| \rightarrow \|[a_{ij}]_1\| = A^* \quad (A \in \Lambda)$$

ein Ringhomomorphismus von  $\Lambda$  auf einen Matrixring  $\Lambda^*$  über  $R_1 = \text{GF}(p)$ .

Im folgenden sollen mit  $[a(x)]_{f(x)}^*$  bzw.  $A^*$  die Bildelemente von  $[a(x)]_{f(x)}$  bzw.  $A$  bei den Ringhomomorphismen (2.5) bzw. (2.6) und mit  $[R_k[x] | (f(x))]^*$  die Bildmenge von  $R_k[x] | (f(x))$  bezeichnet werden.

Es existieren irreduzible Polynome über  $R_k$ , die beim Übergang  $f(x) \rightarrow f(x)^*$  irreduzibel über  $\text{GF}(p)$  bleiben, und andere, die reduzibel werden. Zum Beispiel ist das Polynom  $x^2 + x + 1$  über dem Restklassenring mod  $3^2$  irreduzibel aber über  $\text{GF}(3)$  reduzibel, denn es gilt  $x^2 + x + 1 = (x-1)^2$  über  $\text{GF}(3)$ . Dagegen ist das Polynom  $x^2 + x + 1$  sowohl über dem Restklassenring mod  $2^2$  als auch über  $\text{GF}(2)$  irreduzibel. Mit Polynomen dieser Art wollen wir uns besonders beschäftigen.

**Definition 2.2:** Ein Polynom  $f(x) \in R_k[x]$ , dessen höchster Koeffizient eine prime Restklasse mod  $p^k$  ist, heie total irreduzibel über  $R_k$ , falls  $f(x)^*$  irreduzibel über  $\text{GF}(p)$  ist.

Offensichtlich ist ein total irreduzibles Polynom  $f(x)$  irreduzibel.

Für ein total irreduzibles Polynom  $f(x)$  wählen wir nun ein  $a(x) \in R_k[x]$  beliebig mit  $[a(x)]_{f(x)}^* = 0$ , d. h.  $f(x)^* | a(x)^*$ . Der höchste Koeffizient von  $f(x)$  ist eine prime Restklasse mod  $p^k$  und demzufolge kann man nach (2.2)  $a(x)$  durch  $f(x)$  dividieren:

$$a(x) = f(x)s(x) + r(x) \quad ,$$

wobei  $r(x)$  verschwindet oder  $\text{grad } r(x) < \text{grad } f(x)$  ist. Es folgt

$$a(x)^* = f(x)^* s(x)^* + r(x)^*$$

und da  $\text{grad } r(x)^* < \text{grad } f(x)^*$  ist, muß  $r(x)^* = 0$  sein, denn es war  $f(x)^* | r(x)^*$ . Das bedeutet aber  $p | r(x)$ . Nun ergibt sich

$$p^{k-1} a(x) = p^{k-1} f(x) s(x)$$

oder anders geschrieben  $f(x) | p^{k-1} a(x)$  und weiter

$$0 = [p^{k-1} a(x)]_{f(x)} = [p^{k-1}]_{f(x)} [a(x)]_{f(x)} \quad .$$

Dabei ist  $[p^{k-1}]_{f(x)} \neq 0$ .

Wir können also feststellen, daß ein beliebiges  $[a(x)]_{f(x)}$  aus  $R_k[x] | (f(x))$  mit  $[a(x)]_{f(x)}^* = 0$  entweder Null oder Nullteiler ist.

**Lemma 2.1:** Sei  $f(x)$  total irreduzibel über  $R_k$ . Dann besitzt  $R_k[x] | (f(x))$  außer denjenigen  $[a(x)]_{f(x)}$ , für die  $[a(x)]_{f(x)}^* = 0$  ist, keine weiteren Nullteiler.

**Beweis:** Angenommen es existieren  $a(x), b(x) \in R_k[x]$  mit

$[a(x)]_{f(x)}^* \neq 0$  und  $[b(x)]_{f(x)} \neq 0$ , so daß

$$0 = [a(x)]_{f(x)} [b(x)]_{f(x)} = [a(x)b(x)]_{f(x)}$$

gilt. O.B.d.A. dürfen wir für die Vertreter  $a(x)$  bzw.  $b(x)$  der Restklassen  $[a(x)]_{f(x)}$  bzw.  $[b(x)]_{f(x)}$   $\text{grad } a(x) < \text{grad } f(x)$  bzw.  $\text{grad } b(x) < \text{grad } f(x)$  annehmen. Sicher existieren solche Vertreter wegen der möglichen Division mit Rest durch  $f(x)$  (siehe (2.2)). Obige Gleichung besagt:

$$a(x)b(x) = f(x)r(x)$$

mit  $r(x) \in R_k[x]$ .

Sei  $l$  maximal mit  $p^l | b(x)$  und  $b(x) = [p^l]_k b_1(x)$  mit  $p \nmid b_1(x)$ .

Da  $[b(x)]_{f(x)} \neq 0$ , ist  $b(x) \neq 0$  und daher  $l < k$ . Offensichtlich gilt  $p^l | f(x)r(x)$ . Mit  $g(x)$  bzw.  $t(x) \in \mathbb{Z}[x]$  wollen wir zwei Polynome des Grades  $\text{grad } f(x)$  bzw.  $\text{grad } r(x)$  bezeichnen, deren Koeffizienten irgendwelche Vertreter der Restklassen der entsprechenden Koeffizienten von  $f(x)$  bzw.  $r(x) \in R_k[x]$  sind. Für einen beliebigen Vertreter  $v$  der Restklasse  $[p^k]_k$  gilt wegen  $l < k$  sicher  $p^l | v$ , denn  $v = p^l + ap^k$  ( $a \in \mathbb{Z}$ ). Dann teilt  $p^l$  alle Koeffizienten von  $g(x)t_1(x)$ . Möge  $d_1$  den g.g.T. der Koeffizienten von  $g(x)$  und  $d_2$  den von  $t(x)$  bezeichnen. Damit haben wir

$$g(x)t(x) = (d_1 g_1(x))(d_2 t_2(x)) = d_1 d_2 (g_1(x)t_2(x)),$$

wobei  $g_1(x), t_2(x) \in \mathbb{Z}[x]$  primitive Polynome über dem Integritätsbereich  $\mathbb{Z}$  sind. Mit Hilfe des Satzes von Gauß über primitive Polynome ergibt sich, daß das Produkt  $g_1(x)t_2(x) \in \mathbb{Z}[x]$  ein primitives Polynom ist. Nach dem vorhin Festgestellten folgt jetzt  $p^l | d_1 d_2$ . Da  $p \nmid f(x)$ , teilt  $p$  nicht alle Koeffizienten von  $g(x)$ . Somit haben wir  $p \nmid d_1$ . Es bleibt damit  $p^l | d_2$  übrig, d. h.  $p^l | t(x)$ . Mithin muß  $p^l | r(x)$  gelten.

Mit  $r(x) = [p^l]_k r_1(x)$  ( $r_1(x) \in R_k[x]$ ) haben wir dann

$$0 = a(x)b(x) - f(x)r(x) = [p^l]_k (a(x)b_1(x) - f(x)r_1(x))$$

und da  $k > l$ , folgt

$$p^l | [a(x)b_1(x) - f(x)r_1(x)],$$

$$\text{d. h. } (a(x)b_1(x) - f(x)r_1(x))^* = 0$$

$$\text{oder } a(x)^* b_1(x)^* = f(x)^* r_1(x)^*.$$

Demnach gilt

$$f(x)^* | a(x)^* b_1(x)^*.$$

Da dies eine Beziehung über dem Körper  $R_1 = \text{GF}(p)$  und  $f(x)^*$  irreduzibel ist, ergibt sich  $f(x)^* | a(x)^*$  oder  $f(x)^* | b_1(x)^*$ , wobei

$a(x)^* \neq 0$  und  $b_1(x)^* \neq 0$ . Folglich müßte  $f(x)^*$  Teiler eines nicht-verschwindenden Polynoms echt kleineren Grades sein, was offensichtlich nicht möglich ist.

(2.7) Wir wollen noch eine Zwischenbetrachtung anstellen, die später benötigt wird. Sei  $q(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  aus  $R_K[x]$  ( $a_n \neq 0$ ) ein Polynom kleinsten Grades mit  $q(C) = 0$  ( $a_n$  braucht keine prime Restklasse zu sein), wobei  $C$  eine Matrix einer gewissen Ordnung  $r$  über  $R_K$  ist. Sicher gibt es so ein Polynom, denn  $x^r - 1$  ist ein solches. Wir zeigen, daß unter allen Koeffizienten, welche  $[p]_K$  in niedrigster Potenz enthalten,  $a_n$  vorkommt. Angenommen das wäre nicht der Fall. Dann gilt, wenn wir unter  $[p^1]_K$  diese niedrigste Potenz von  $[p]_K$  verstehen,

$$\left[ p^{k-1} \right]_K q(x) = \left[ p^{k-1} \right]_K a_n x^n + r(x) = r(x)$$

mit  $\text{grad } r(x) < n$  und  $r(x) \neq 0$ . Nun ergibt sich der Widerspruch  $0 = \left[ p^{k-1} \right]_K q(C) = r(C)$  wegen der Minimalität von  $q(x)$ .

Lemma 2.2: Sei  $C$  eine irreduzible  $m \times m$ -Matrix der Ordnung  $r$  über  $R_K$  mit der Eigenschaft, daß  $C^*$  irreduzibel von der Ordnung  $r$  über  $GF(p)$  ist. Des weiteren soll  $R_K[C]$  die lineare Hülle von  $C$  über  $R_K$  bezeichnen. Dann ist  $m$  die kleinste natürliche Zahl mit  $r \mid [p^m - 1]$ ,  $R_K[C]$  ein Ring, und es existiert ein total irreduzibles normiertes Polynom  $f(x) \in R_K[x]$  des Grades  $m$  mit  $f(C) = 0$ , und das aus  $f(x)$  erzeugte Hauptideal besteht aus genau den  $\phi(x) \in R_K[x]$  mit  $\phi(C) = 0$ . Unter allen Polynomen kleinsten Grades, die  $C$  als Nullstelle haben, ist  $f(x)$  das einzige normierte. Wir nennen  $f(x)$  das zu  $C$  gehörige Minimalpolynom. Weiter gilt  $R_K[C] \cong R_K[x] / (f(x))$ .

Beweis: Man überzeugt sich leicht davon, daß  $R_K[C]$  ein Ring ist.

Es gilt  $C^r - E_m = 0$ , also ist  $u(x) = x^r - 1 \in R_K[x]$  ein normiertes Polynom mit  $u(C) = 0$ . Sei nun  $f(x)$  ein normiertes Polynom kleinsten Grades in  $R_K[x]$  mit  $f(C) = 0$ . Da  $C^*$  irreduzibel vom Grade  $m$  und  $\text{ord } C^* = r$ , ist  $m$  die kleinste natürliche Zahl mit  $r \mid p^m - 1$  nach (2.4). Darum ist  $p \nmid r$ .

(2.8) Wegen (2.3) tritt dann jeder irreduzible Faktor in der Zerlegung von  $(x^r - 1)$  über  $GF(p)$ , die ja bekanntlich eindeutig ist, genau einmal auf.

Als nächstes wollen wir zeigen, daß  $f(x)$  ein total irreduzibles Polynom des Grades  $m$  ist. Wegen (2.2) können wir die Division mit Rest von  $x^r - 1$  durch  $f(x)$  ausführen:

$$x^r - 1 = f(x)s(x) + r(x),$$

wobei  $\text{grad } r(x) < \text{grad } f(x)$  oder  $r(x) = 0$ . Es gilt

$$0 = C^r - E_m = f(C)s(C) + r(C),$$

also  $r(C) = 0$ . Wegen der Minimalität von  $f(x)$  für normierte Polynome folgt, daß der höchste Koeffizient von  $r(x)$  durch  $[p]_K$  teilbar ist, falls  $r(x) \neq 0$ . Aus (2.7) ergibt sich jetzt  $p \mid r(x)$ . Im Falle  $r(x) = 0$  gilt  $p \mid r(x)$  trivialerweise. Somit ist

$$(x^r - 1)^* = f(x)^* s(x)^*$$

und infolge dessen  $f(x)^* \mid (x^r - 1)^*$  in  $R_1 = GF(p)$ .

(2.9) Zusammen mit (2.8) ergibt sich nun, daß  $f(x)^*$  seine irreduziblen Faktoren genau einmal enthält.

Nach (2.4) existiert ein zu  $C^* \in R_1[C^*]$  gehöriges Minimalpolynom vom Grade  $m$ . Dieses können wir in der Form  $g(x)^k$  schreiben mit  $g(x) \in R_K[x]$ . Wegen  $g(C^*)^k = 0$  gilt  $p \mid g(C)$  und deshalb  $p^k \mid g(C)^k$ , also  $g(C)^k = 0$ . Wir können  $g(x)^k$  durch  $f(x)$  dividieren:

$$g(x)^k = f(x)s_1(x) + r_1(x),$$

wobei  $\text{grad } r_1(x) < \text{grad } f(x)$  oder  $r_1(x) = 0$  ist. Es folgt

$$0 = g(C)^k = f(C)s_1(C) + r_1(C).$$

Somit ist  $r_1(C) = 0$ . Genau wie vorhin erhält man wegen der Minimalität von  $f(x)$  für normierte Polynome  $p \mid r_1(x)$ . Daraus ergibt sich nun  $f(x)^* \mid (g(x)^k)^*$ . Aus (2.9) und der Irreduzibilität von  $g(x)^*$  schließen wir  $f(x)^* = g(x)^*$ , d. h. aber nichts anderes, als daß  $f(x)$  total irreduzibel vom Grade  $m$  ist. Nun zeigen wir, daß jedes Polynom, welches  $C$  als Nullstelle hat, durch  $f(x)$  teilbar ist. Sei  $t(x) \in R_k[x]$  ein beliebiges Polynom mit  $t(C) = 0$ . Wir dividieren  $t(x)$  durch  $f(x)$ :

$$t(x) = f(x)s_2(x) + r_2(x),$$

wobei  $\text{grad } r_2(x) < \text{grad } f(x) = m$  oder  $r_2(x) = 0$  ist. Wir führen nun die Annahme  $r_2(x) \neq 0$  zum Widerspruch. Es ist

$$0 = t(C) = f(C)s_2(C) + r_2(C)$$

und demzufolge  $r_2(C) = 0$ . Sei  $[p^1]_k$  die größte  $[p]_k$ -Potenz, die in  $r_2(x)$  enthalten ist, und  $r_2(x) = [p^1]_k r_3(x)$ . Wegen (2.7) ist  $p \nmid b_n$  für den höchsten Koeffizienten  $b_n$  von  $r_3(x)$ . Wir können schreiben:

$$r_2(x) = [p^1]_k b_n r_4(x),$$

wobei  $r_4(x) \in R_k[x]$  normiert ist. Nun erhalten wir

$$0 = r_2(C) = [p^1]_k b_n r_4(C).$$

Auf Grund der Annahme  $r_2(x) \neq 0$  muß  $1 < k$  sein, woraus folgt  $p \mid r_4(C)$ , d. h.  $r_4(C^*) = 0$ . Da  $g(x)^*$  das Minimalpolynom von  $C^* \in R_1[C^*]$  über  $\text{GF}(p)$  ist, ergibt sich der Widerspruch

$$\text{grad } g(x)^* = m > \text{grad } r_4(x)^*.$$

Es folgt  $r_2(x) = 0$  oder anders ausgedrückt  $f(x) \mid t(x)$ . Wir erhalten schließlich, daß das aus  $f(x)$  erzeugte Hauptideal  $(f(x))$  in  $R_k[x]$  genau aus allen denjenigen Polynomen aus  $R_k[x]$

besteht, die  $C$  als Nullstelle haben. Mithin ist das aus  $f(x)$  erzeugte Hauptideal  $(f(x))$  in  $R_k[x]$  der Kern des Homomorphismus  $\varphi(x) \rightarrow \varphi(C)$ ,  $\varphi(x) \in R_k[x]$ , so daß nach dem Homomorphiesatz gilt

$$R_k[x] \mid (f(x)) \cong R_k[G].$$

Man kann weiter sagen, daß sich alle Polynome kleinsten Grades mit der Nullstelle  $C$  von  $f(x)$  nur um einen Faktor aus  $R_k$  unterscheiden können. Unter diesen ist  $f(x)$  einziges normiertes Polynom.

**Lemma 2.3:** Sei  $r$  eine natürliche Zahl und  $n$  die kleinste natürliche Zahl mit  $r \mid p^n - 1$ . Dann existiert eine über  $R_k$  irreduzible zyklische  $n \times n$ -Matrixgruppe  $\langle C \rangle$  der Ordnung  $r$  mit der Eigenschaft, daß  $\langle C^* \rangle$  irreduzibel von der Ordnung  $r$  über  $GF(p)$  ist.

**Beweis:** Bekanntlich existiert eine irreduzible zyklische  $n \times n$ -Matrixgruppe  $\langle A' \rangle$  der Ordnung  $r$  über  $GF(p)$  (Pazderski /2/, S. 14). Sicher gibt es eine  $n \times n$ -Matrix  $A$  über  $R_k$  mit  $A^* = A'$ .

Man setze nämlich für jeden Koeffizienten aus  $A'$  irgendeinen seiner Vertreter aus  $\mathbb{Z}$  und anschließend für diesen seine Restklasse mod  $p^k$  ein. Für  $A$  gilt dann

$$(A^r)^* = (A^*)^r = A'^r = E_n^1.$$

$E_n^1$  bezeichne die Einheitsmatrix des Formates  $n \times n$  über  $R_k$ .

Das heißt  $A^r = E_n^k + [p]_k B$ , wobei  $E_n^k$  die  $n \times n$ -Einheitsmatrix über  $R_k$  und  $B$  eine gewisse  $n \times n$ -Matrix über  $R_k$  bezeichnen. Es folgt

$$(A^r)^p = E_n^k + \binom{p}{1} [p]_k B + \binom{p}{2} [p^2]_k B^2 + \dots + \binom{p}{p} [p^p]_k B^p = E_n^k + [p^2]_k B',$$

wobei  $B'$  eine gewisse  $n \times n$ -Matrix über  $R_k$  ist. Weiter gilt

$$\begin{aligned} (A^r)^{p^2} &= (E_n^k + [p^2]_k B')^p \\ &= E_n^k + \binom{p}{1} [p^2]_k B' + \binom{p}{2} [p^4]_k B'^2 + \dots + \binom{p}{p} [p^{2p}]_k B'^p \\ &= E_n^k + [p^3]_k B'' \quad \text{usw.} \end{aligned}$$

Letzten Endes ergibt sich  $(A^r)^{p^k} = E_n^k$  über  $R_k$  oder anders

$$(A^{p^k})^r = E_n^k, \quad \text{d. h. } \text{ord}(A^{p^k}) \mid r.$$

Wegen  $A^* = A'$  gilt weiter

$$\left(A^{p^k}\right)^* = \left(A^*\right)^{p^k} = A'^{p^k}$$

und somit  $\text{ord}\left(A^{p^k}\right)^* = \text{ord}\left(A'\right)^{p^k}$ . Da nun  $\text{ord} A' = r$  und

$(r, p) = 1$ , folgt  $r = \text{ord}(A') = \text{ord}\left(A'\right)^{p^k}$ .

Damit ist  $r = \text{ord}\left(A^{p^k}\right)^*$ .

Nun ergibt sich  $r = \text{ord}\left(A^{p^k}\right)^* \mid \text{ord} A^{p^k} \mid r$ , da bei einer homomorphen Abbildung die Ordnung des Bildelementes ein Teiler der

Ordnung vom Urbild ist. Das heißt aber,  $r = \text{ord} A^{p^k}$ . Die Matrix

$C = A^{p^k}$  hat nun die im Lemma behaupteten Eigenschaften.

Aus Lemma 2.2 und Lemma 2.3 folgt die Existenz eines total irreduziblen Polynoms  $f(x)$  des Grades  $n$  über  $R_k$  mit  $f(x) \mid x^r - 1$ , wobei  $n = \text{ord } p \text{ mod } r$  ist.

Lemma 2.4: Bezeichne  $r$  eine natürliche Zahl und  $n$  die kleinste natürliche Zahl mit  $r \mid p^n - 1$ . Sei weiter

$x^r - 1 = f_1(x) \dots f_t(x)r(x)$  eine Zerlegung von  $x^r - 1 \in R_k[x]$  in total irreduzible normierte Bestandteile  $f_1(x), f_2(x), \dots, f_t(x)$  des Grades  $n$  und ein weiteres Polynom  $r(x) \in R_k[x]$ , welches keines der obigen Polynome als Faktor mehr enthält. Dann sind die  $f_i(x)$  ( $i = 1, 2, \dots, t$ ) verschieden und es handelt sich um die einzige Zerlegung dieser Art. Weiter ist  $f_i(x)^* \neq f_j(x)^*$ , wenn  $i \neq j$  ( $i, j = 1, 2, \dots, t$ ). Die diesem Lemma vorangehende Bemerkung besagt, daß  $t \geq 1$  ist.

Beweis: Mittels (2.5) ergibt sich

$$\left(x^r - 1\right)^* = \left(f_1(x) \dots f_t(x)r(x)\right)^* = f_1(x)^* \dots f_t(x)^* r(x)^*.$$

Es handelt sich hier um eine Zerlegung von  $\left(x^r - 1\right)^*$  über  $\text{GF}(p)$ . Wegen (2.3) folgt  $f_i(x)^* \neq f_j(x)^*$  und damit  $f_i(x) \neq f_j(x)$ ,

falls  $i \neq j$  für alle  $i, j = 1, 2, \dots, t$ .

Wir wollen indirekt beweisen, daß  $x^r - 1$  kein von  $f_1(x), \dots, f_t(x)$  verschiedenes total irreduzibles normiertes Polynom des Grades  $n$  über  $R_k$  als Faktor enthält.

(2.10) Dazu nehmen wir an, daß ein total irreduzibles normiertes Polynom  $g(x) \in R_k[x]$  des Grades  $n$  mit  $g(x) \mid (x^r - 1)$  existiert, welches nicht unter den  $f_1(x), \dots, f_t(x)$  vorkommt.

Aus (2.5) folgt  $g(x)^* \mid (x^r - 1)^*$ . Es können nur zwei Fälle eintreten, nämlich, daß  $g(x)^*$  unter den  $f_i(x)^*$  ( $i = 1, \dots, t$ ) vorkommt oder nicht.

1. Fall: Es existiert ein  $f_i(x)$  ( $1 \leq i \leq t$ ) mit  $f_i(x)^* = g(x)^*$ .

O.B.d.A. sei  $i = 1$ . Wir wollen nun annehmen, daß

$[f_2(x) \dots f_t(x)] g(x) \in R_k[x] \mid (g(x))$  entweder Null oder Nullteiler ist und diese Annahme zu einem Widerspruch führen. Aus dieser Annahme folgt mittels (2.5) und Lemma 2.1

$$[f_2(x)^* \dots f_t(x)^* r(x)^*] g(x)^* = [f_2(x) \dots f_t(x) r(x)] g(x)^* = 0$$

und daraus  $g(x)^* \mid f_2(x)^* \dots f_t(x)^* r(x)^*$ . Das wäre wegen

$g(x)^* = f_1(x)^*$ , was  $g(x)^* \mid x^r - 1$  nach sich zieht, im Widerspruch zu (2.3).

Aus der Beziehung

$$\begin{aligned} [f_1(x)] g(x) [f_2(x) \dots f_t(x) r(x)] g(x) &= [f_1(x) \dots f_t(x) r(x)] g(x) \\ &= [x^r - 1] g(x) = 0 \end{aligned}$$

folgt nun  $[f_1(x)] g(x) = 0$ , da der zweite Faktor, wie wir soeben gesehen haben, weder Null noch Nullteiler ist. Das heißt aber nichts anderes, als daß  $g(x) \mid f_1(x)$  ist und da die Beziehung  $\text{grad } g(x) = n = \text{grad } f_1(x)$  gilt und außerdem beide Polynome normiert waren, folgt  $g(x) = f_1(x)$ . Das ist aber ein Widerspruch zur Annahme über  $g(x)$ . Also kann der erste Fall nicht eintreten.

2. Fall: Es existiert kein  $f_i(x)$  ( $1 \leq i \leq t$ ) mit  $f_i(x)^* = g(x)^*$ .

Um zu zeigen, daß  $[f_1(x) \dots f_t(x)]_{g(x)} \in R_k[x] \mid (g(x))$  weder Null noch Nullteiler ist, wollen wir das Gegenteil annehmen.

Mit Lemma 2.1 folgt aus dieser Annahme

$$[f_1(x) \dots f_t(x)]_{g(x)}^* = 0 .$$

Nach (2.5) ist dann  $[f_1(x)]_{g(x)}^* \dots [f_t(x)]_{g(x)}^* = 0$ . Da in

$[R_k[x] \mid (g(x))]^* \cong R_k[x] \mid (g(x)^*)$  kein Nullteiler existiert, muß es ein  $i$  ( $1 \leq i \leq t$ ) geben mit  $[f_i(x)]_{g(x)}^* = 0$ , d. h.

$g(x)^* \mid f_i(x)$ . Da  $\text{grad } g(x)^* = n = \text{grad } f_i(x)^*$  und außerdem beide Polynome normiert sind, muß  $g(x)^* = f_i(x)^*$  gelten. Das ist aber

nicht möglich wegen der Voraussetzung im zweiten Fall. Damit ist  $[f_1(x) \dots f_t(x)]_{g(x)}$  weder Null noch Nullteiler. Es gilt in  $R_k[x] \mid (g(x))$

$$[f_1(x) \dots f_t(x)]_{g(x)} [r(x)]_{g(x)} = [x^r - 1]_{g(x)} = 0 ,$$

da  $g(x) \mid x^r - 1$  über  $R_k$  war. Der erste Faktor ist aber weder Null noch Nullteiler, wie wir soeben festgestellt haben.

Es muß also  $[r(x)]_{g(x)} = 0$  sein. Das heißt  $g(x) \mid r(x)$ , was ein Widerspruch zur Voraussetzung dieses Lemmas ist. Der betrachtete zweite Fall tritt damit auch nicht ein.

Darum ist unsere Annahme (2.10) falsch. Die in diesem Lemma behauptete Eindeutigkeit ergibt sich nun ohne Schwierigkeit.

Nun wollen wir das endgültige Resultat, um das es hier im wesentlichen geht, im folgenden Satz formulieren.

Satz: Sei  $\langle A \rangle$  irgendeine zyklische irreduzible  $n \times n$ -Matrixgruppe der Ordnung  $r$  über  $R_k$  mit der Eigenschaft, daß  $\langle A^k \rangle$  eine irreduzible zyklische Matrixgruppe der Ordnung  $r$  über  $GF(p)$  ist. Dann ist  $n$  die kleinste natürliche Zahl mit  $r \mid p^n - 1$ . Sei umgekehrt die natürliche Zahl  $r$  vorgegeben und  $n$  die kleinste natürliche Zahl mit  $r \mid p^n - 1$ . Dann existiert bis auf Äquivalenz genau eine zyklische irreduzible Matrixgruppe  $\langle G \rangle$  der Ordnung  $r$ , für die  $\langle G^k \rangle$  ebenfalls irreduzibel ist und die Ordnung  $r$  hat.

**Beweis:**  $\langle A^* \rangle$  ist eine irreduzible  $n \times n$ -Matrixgruppe der Ordnung  $r$  über  $GF(p)$ . Demzufolge ist nach (2.4)  $n$  die kleinste natürliche Zahl mit  $r \mid p^n - 1$ .

Betrachten wir nun den zweiten Teil des Satzes. Nach Lemma 2.3 existiert sicher eine zyklische Matrixgruppe  $\langle C \rangle$  über  $R_k$  mit den geforderten Eigenschaften. Sei  $\langle B \rangle$  eine weitere solche Matrixgruppe. Unser Ziel ist es nun, nachzuweisen, daß  $\langle C \rangle$  und  $\langle B \rangle$  äquivalent sind.

Möge  $g(x)$  das zu  $B$  gehörige Minimalpolynom bezeichnen.  $g(x)^*$  ist offensichtlich das zu  $B^*$  gehörige Minimalpolynom über dem Körper  $R_1 = GF(p)$ . Nach (Pazderski /2/, S. 14) gibt es bis auf Äquivalenz genau eine irreduzible zyklische  $n \times n$ -Matrixgruppe der Ordnung  $r$  über  $GF(p)$ . Demzufolge existiert eine Potenz  $C^{*v}$  von  $C^*$  mit  $(v, r) = 1$  und eine invertierbare  $n \times n$ -Matrix  $T$  über  $GF(p)$  mit

$$TC^{*v}T^{-1} = B^* .$$

Darum muß nun gelten

$$g(C^{*v})^* = 0 .$$

Es ist klar, daß  $g(x)^*$  das zu  $C^{*v}$  gehörige Minimalpolynom sein muß, denn es ist ein normiertes irreduzibles Polynom mit  $C^{*v}$  als Nullstelle. Sei  $g_1(x) \in R_k[x]$  das zu  $C^v \in R_k[C^v]$  gehörige Minimalpolynom. Dann ist  $g_1(x)^*$  das zu  $C^{v*} = C^{*v}$  aus  $R_1[C^{*v}]$  gehörige Minimalpolynom. Wegen der eindeutigen Bestimmtheit des Minimalpolynoms muß dann

$$(2.11) \quad g(x)^* = g_1(x)^*$$

gelten. Mit  $t(x) := x^r - 1$  gilt  $t(B) = 0$  und  $t(C^v) = 0$ . Aus Lemma 2.2 können wir dann  $g(x) \mid t(x)$  und  $g_1(x) \mid t(x)$  entnehmen.

Lemma 2.4 und (2.11) ergeben nun  $g_1(x) = g(x)$ . Also ist  $g(x)$  das zu  $C^v \in R_k[C^v]$  gehörige Minimalpolynom. Es folgt unter Zuhilfenahme von Lemma 2.2

$$(2.12) \quad R_k[C] = R_k[C^v] \cong R_k[x] \mid (g(x)) \cong R_k[B] .$$

Also sind  $R_k[C]$  und  $R_k[B]$  irreduzible Matrixdarstellungen des Grades  $n$  von  $R_k[x] \mid (g(x))$ . Man kann sich folgendermaßen überlegen, daß alle Voraussetzungen für Lemma 1.4 erfüllt sind.

Sei  $[a(x)]_{g(x)} \neq 0$  ein beliebiges Element aus  $R_k[x] \mid (g(x))$  mit  $a(x) \in R_k[x]$ , wobei o.B.d.A.  $\text{grad } a(x) < \text{grad } g(x)$  angenommen werden kann. Es gelte  $a(x) = [p^1]_k a_1(x)$  mit  $p \nmid a_1(x) \in R_k[x]$ . Dabei ist  $1 < k$  und  $\text{grad } a_1(x) < \text{grad } g(x)$ . Somit gilt nach (2.5)

$$[a_1(x)]_{g(x)}^* = [a_1(x)^*]_{g(x)}^* \neq 0.$$

Lemma 2.1 besagt nun, daß  $[a_1(x)]_{g(x)}$  weder Null noch Nullteiler ist. Daraus folgt mittels (2.1), daß  $[a_1(x)]_{g(x)}$  invertierbar ist. Die Anwendung des Lemma 1.4 ergibt nun, daß sowohl  $R_k[C]$  als auch  $R_k[B]$  zur regulären Matrixdarstellung von  $R_k[x] \mid (g(x))$ , die ebenfalls den Grad  $n$  über  $R_k$  hat, äquivalent sind. Daraus folgt aber

$$(2.13) \quad R_k[B] \text{ äquivalent zu } R_k[C].$$

Für beliebige Polynome aus  $R_k[x]$  ist nach (2.2) die Division mit Rest durch  $g(x)$  möglich. Darum lassen sich die Vertreter sämtlicher Restklassen von  $R_k[x]$  nach  $(g(x))$  in der Form

$$(2.14) \quad \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \dots + \alpha_{n-1} x^{n-1},$$

wobei die  $\alpha_i \in R_k$  ( $i = 1, \dots, n-1$ ) sind, aufschreiben. Um zu sehen, daß zwei verschiedene Polynome, deren Grad kleiner als  $n$  ist, nicht in derselben Restklasse von  $R_k[x]$  nach  $(g(x))$  liegen, nehmen wir das Gegenteil an. Dann wäre ihre Differenz ein von Null verschiedenes Polynom kleineren Grades als  $n$ , welches in  $(g(x))$  liegen müßte, was sicher nicht möglich ist. Also wird in der Form (2.14) jede Restklasse genau einmal vertreten. Man rechnet leicht nach, daß ihre Anzahl  $p^{nk}$  ist. Sei  $a(x)$  ein beliebiges Element aus  $R_k[x]$  und  $a_1(x)$  sein Vertreter von der Form (2.14). Nach Lemma 2.1 ist das Element  $[a(x)]_{g(x)} = [a_1(x)]_{g(x)}$  genau dann Nullteiler oder Null, wenn  $p \mid a_1(x)$  ist. Wir wollen

$H := \left\{ [a_1(x)]_{g(x)} \in R_k[x] \mid (g(x)) \mid \text{grad } a_1(x) < n \text{ und } p \nmid a_1(x) \right\}$ ,  
d. h. die Menge aller Nichtnullteiler von  $R_k[x] \mid (g(x))$ , einfüh-

ren. Bekanntlich ist  $H$  eine Gruppe bezüglich der Multiplikation. Jede in  $R_K[x] \setminus (g(x))$  enthaltene multiplikative Gruppe liegt in  $H$ , denn sie darf keine Nullteiler enthalten. Mit Hilfe der vorherigen Betrachtungen läßt sich die Anzahl der Nullteiler einschließlich Null in  $R_K[x] \setminus (g(x))$  zu  $p^{(k-1)n}$  berechnen. Daraus folgt

$$|H| = p^{kn} - p^{(k-1)n} = p^{(k-1)n}(p^n - 1).$$

Als abelsche Gruppe enthält  $H$  nur Normalteiler als Untergruppen. Man kann jetzt unter Ausnutzung gewisser Syloweigenschaften auf die Existenz genau einer Untergruppe  $U$  der Ordnung  $p^n - 1$  in  $H$  schließen. Des weiteren muß eine Untergruppe von  $H$ , deren Ordnung ein Teiler von  $p^n - 1$  ist, in  $U$  liegen. Es war  $R_K[C] \cong R_K[x] \setminus (g(x))$  nach (2.12). Bezeichne  $U_1$  das zu  $U$  isomorphe Bild in  $R_K[C]$ . Dann ist  $U_1$  die einzige Gruppe der Ordnung  $p^n - 1$  in  $R_K[C]$  bezüglich der Multiplikation und alle Gruppen aus  $R_K[C]$  bezüglich der Multiplikation, deren Ordnung ein Teiler von  $p^n - 1$  ist, liegen in  $U_1$ . Somit ist  $\langle C \rangle \leq U_1$ , denn  $|\langle C \rangle| = r |p^n - 1|$ . Bei dem Ringhomomorphismus (2.6) wird  $R_K[C]$  auf  $R_K[C]^*$ , einem Matrixring über  $GF(p)$ , abgebildet.  $U_1^*$  bezeichne dabei das Bild von  $U_1$ . Die Gruppe  $U_1$  bezüglich der Multiplikation wird dann homomorph auf  $U_1^*$  abgebildet (hierbei soll nur ein Gruppenhomomorphismus bezüglich der Multiplikation gemeint sein).  $K$  bezeichne den Kern dieser homomorphen Abbildung. Dann ist  $U_1/K \cong U_1^*$ . Ein Element aus  $K$  ist offensichtlich eine Matrix der Gestalt  $E_n + [p]_K D$ , wobei  $D$  eine  $n \times n$ -Matrix über  $R_K$  und  $E_n$  die Einheitsmatrix über  $R_K$  sind. Diese Matrix hat bekanntlich  $p$ -Potenzordnung (siehe z. B. Beweis zu Lemma 2.3). Da diese Matrix aber in  $U_1$  liegt und  $|U_1| = p^n - 1$  gilt, muß  $K = \{E_n\}$  sein. Darum ist  $U_1 \cong U_1^*$ . Nach (2.4) ist  $R_K[C]^* \cong R_1[C^*]$  ein endlicher Körper. Die multiplikative Gruppe von  $R_K[C]^*$  ist demzufolge zyklisch, d. h.  $U_1^*$

ist zyklisch.  $U_1$  ist demnach eine zyklische Gruppe.  $\langle C \rangle$  ist also die einzige Untergruppe der Ordnung  $r$  in  $U_1$ . Folglich kann es außer  $\langle C \rangle$  keine weiteren Gruppen der Ordnung  $r$  in  $R_k[C]$  bezüglich der Multiplikation geben. Genauso ist  $\langle B \rangle$  die einzige Gruppe der Ordnung  $r$  in  $R_k[B]$  bezüglich der Multiplikation. Aus (2.13) folgt nun, daß  $\langle B \rangle$  und  $\langle C \rangle$  zueinander äquivalent sind.

### Literatur

- /1/ Bourbaki, N., Algebra, Hermann, Paris 1948
- /2/ Pazderski, G., Über lineare auflösbare Gruppen  
Math. Nachr. 45, 1 - 68 (1970)
- /3/ v. d. Waerden, B. L.,  
Moderne Algebra  
2. verb. Aufl., Berlin 1937

eingegangen: 10. 9. 1977

### Anschrift des Verfassers:

Dipl. Math. Günther Steffen  
Wilhelm-Pieck-Universität Rostock  
Sektion Mathematik  
DDR-25 Rostock  
Universitätsplatz 1



Zur Darstellung der Lösung allgemeiner elliptischer Randwert-  
probleme

---

1. Einleitung

In /5/, /4/ wurde für die Lösung des Dirichlet-Problems elliptischer Gleichungen beliebiger Ordnung eine Darstellung durch Maßstapel mit dem Träger auf dem Rand des Gebietes hergeleitet, die als natürliche Verallgemeinerung der harmonischen Maße in der klassischen Potentialtheorie anzusehen sind. Die in /5/, /4/ entwickelte Theorie besitzt potentialtheoretische und funktionalanalytische Aspekte. Unabhängig von weiterführenden potentialtheoretischen Überlegungen sind zum Beispiel in /7/ die funktionalanalytischen Gesichtspunkte herausgearbeitet. Dabei wird im wesentlichen mit den Räumen  $C$ ,  $C^m$  und mit Räumen sogenannter Whitney-Taylorfelder gearbeitet. Die Methode basiert auf punktwisen a-priori-Abschätzungen, die von C. Miranda /3/ und S. Agmon /1/ für das Dirichlet-Problem und glatte Ränder bewiesen wurden.

Darüber hinaus sind für allgemeinere elliptische Randwertprobleme von S. Agmon, A. Douglis und L. Nirenberg /2/ etwas schwächere Schauder-Abschätzungen bewiesen worden, und es liegt der Gedanke nahe, für solche allgemeinen Randwertprobleme einen zu /4/ analogen funktionalanalytischen Zugang zu entwickeln, der auf diesen Schauder-Abschätzungen und auf der Verwendung von entsprechend angepassten Räumen hölderstetig differenzierbarer Funktionen beruht. Die technischen Hilfsmittel sind im Hinblick darauf bereits in /6/ bereitgestellt worden. In der vorliegenden Arbeit sollen diese in zum Teil modifizierter Form zu Realisierung des genannten Programms eingesetzt werden.

2. Hilfsmittel und Bezeichnungen

Wir beschäftigen uns mit der folgenden Problemstellung. In einem Gebiet  $\Omega \subset \mathbb{R}^n$  wird eine Lösung der elliptischen Diffe-

Differentialgleichung

$$Lu = 0$$

gesucht, die auf dem Rand  $\partial \Omega$  den Bedingungen

$$B_j u|_{\partial \Omega} = \varepsilon_j \quad (j = 1, \dots, m)$$

genügt.  $L$  und  $B_j$  seien lineare Differentialoperatoren der Ordnung  $2m$  bzw.  $m_j$ ,  $q = \max(2m, m_j)$ , und  $\sigma \in (0, 1)$  sei fest gewählt.

$C^{1+\sigma}(\bar{\Omega})$  ( $1 \geq 0$ , ganz) bezeichne den Vektorraum aller im abgeschlossenen Gebiet  $\bar{\Omega}$  1-mal stetig differenzierbaren Funktionen, deren Ableitungen der Ordnung 1 einer Hölderbedingung mit dem Exponenten  $\sigma$  genügen. Mit der Norm

$$\|u\|_{C^{1+\sigma}(\bar{\Omega})} = \sum_{|\alpha| \leq 1} \sup_{x \in \bar{\Omega}} |D^\alpha u(x)| + \sum_{|\alpha|=1} \sup_{\substack{x, y \in \bar{\Omega} \\ x \neq y}} \frac{|D^\alpha u(x) - D^\alpha u(y)|}{|x - y|^\sigma}$$

$$(\alpha = (\alpha_1, \dots, \alpha_n), \alpha_i \geq 0, \text{ ganz}, |\alpha| = \sum_{i=1}^n \alpha_i,$$

$$D^\alpha = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}) \text{ wird } C^{1+\sigma}(\bar{\Omega}) \text{ ein Banachraum.}$$

$C^{1+\sigma}(\partial \Omega)$  wird analog in lokalen Koordinaten auf dem Rand erklärt, falls  $\partial \Omega$  genügend glatt ist.

In /2/ spielen folgende Voraussetzungen eine Rolle:

(i) Der Rand  $\partial \Omega$  sei hinreichend glatt.

Da wir die Glattheitsbedingungen hier nicht explizit benötigen, verzichten wir auf ihre genaue Formulierung und verweisen auf /2/.

(ii) Die Koeffizienten des Operators  $L$  seien in  $C^{q-2m_j+\sigma}(\Omega)$  enthalten und in der Norm dieses Raumes durch eine Konstante  $k$  beschränkt.

(iii) Die Koeffizienten der Operatoren  $B_j$  seien Elemente des Raumes  $C^{q-m_j+\epsilon}(\partial\Omega)$  und ihre Norm sei ebenfalls durch  $k$  beschränkt. Ferner sei bzgl. der Operatoren  $B_j$  die sogenannte Lopatinskij-Bedingung, vgl. /2/, erfüllt.

Als grundlegende Annahme für das Folgende formulieren wir Bedingung I: Die im Raum  $C^{q+\epsilon}(\bar{\Omega})$  enthaltenen Lösungen der Gleichung  $Lu = 0$  genügen Abschätzungen der Gestalt

$$\|u\|_{C^{q+\epsilon}(\bar{\Omega})} \leq C \sum_{j=1}^m \|B_j u\|_{C^{q-m_j+\epsilon}(\partial\Omega)},$$

wobei die Konstante  $C$  von  $u$  unabhängig sein soll.

In /2/ ist gezeigt, daß die Bedingung I zum Beispiel dann erfüllt ist, wenn die Voraussetzungen (i)-(iii) gelten und das obige Randwertproblem eindeutig lösbar ist.

$K \subset \mathbb{R}^n$  bezeichne eine kompakte Menge und  $\mathcal{W}^1(K)$  den Vektorraum aller auf  $K$  definierten Taylorfelder der Ordnung 1, d. h. aller Systeme  $f = (f_\alpha)_{|\alpha| \leq 1}$  von auf  $K$  definierten stetigen Funktionen. Für festes  $x \in K$  und  $f \in \mathcal{W}^1(K)$  sei  $R_x^1 f$  das Taylorfeld mit den Komponenten

$$(R_x^1 f)_\alpha(y) = f_\alpha(y) - \sum_{|\beta| \leq 1-|\alpha|} \frac{1}{\beta!} f_{\alpha+\beta}(x)(y-x)^\beta \quad (|\alpha| \leq 1)$$

$$(\beta = (\beta_1, \dots, \beta_n), \beta! = \beta_1! \dots \beta_n!),$$

$$\alpha + \beta = (\alpha_1 + \beta_1, \dots, \alpha_n + \beta_n),$$

$$(y-x)^\beta = (y_1-x_1)^{\beta_1} \dots (y_n-x_n)^{\beta_n}.$$

Das Taylorfeld  $f \in \mathcal{W}^1(K)$  werde ein Whitney'sches Taylorfeld ( $W$ -Taylorfeld) der Ordnung  $1+\epsilon$  genannt, falls eine Konstante  $C > 0$  derart existiert, daß für alle  $x, y \in K$  und  $|\alpha| \leq 1$  die Ungleichung

$$|(R_x^1 f)_\alpha(y)| \leq C |x-y|^{1+\epsilon-|\alpha|}$$

gilt.  $W^{1+\delta}(K)$  sei der Raum aller dieser  $W$ -Taylorfelder mit der Norm

$$\|f\|_{W^{1+\delta}(K)} = \sum_{|\alpha| \leq 1} \sup_{x \in K} |f_\alpha(x)| + \sum_{|\alpha|=1} \sup_{\substack{x, y \in K \\ x \neq y}} \frac{|f_\alpha(x) - f_\alpha(y)|}{|x-y|^\delta}$$

Die kompakte Menge  $K \subset \mathbb{R}^n$  heißt  $(1+\delta)$ -regulär, falls  $W^{1+\delta}(K)$  bezüglich dieser Norm vollständig ist.

Ein Taylorfeld  $f \in \mathcal{W}^1(K)$  läßt sich genau dann zu einer Funktion  $\tilde{f} \in C^{1+\delta}(\mathbb{R}^n)$  fortsetzen (mit  $D^\alpha \tilde{f}(x) = f_\alpha(x)$  für alle  $x \in K$  und  $|\alpha| \leq 1$ ), falls  $f \in W^{1+\delta}(K)$ .

Betrachtet man speziell  $K = \bar{\Omega}$ , so gilt im Sinne einer naheliegenden Identifizierung stets  $W^{1+\delta}(\bar{\Omega}) \subset C^{1+\delta}(\bar{\Omega})$ , wobei der Fall  $W^{1+\delta}(\bar{\Omega}) \neq C^{1+\delta}(\bar{\Omega})$  nicht auszuschließen ist. Im folgenden identifizieren wir  $u \in C^{1+\delta}(\bar{\Omega})$  stets mit dem gemäß  $(D^\alpha u)_{|\alpha| \leq 1}$  induzierten Taylorfeld.

Zu den voranstehenden Aussagen vergleiche man /6/.

Es sei jetzt  $l=q$ . Für  $f = (f_\alpha)_{|\alpha| \leq q} \in W^{q+\delta}(\bar{\Omega})$  setzen wir

$$f(0, \dots, 0) = u$$

und schreiben gelegentlich  $u = f$ .

Es sei

$D(\bar{\Omega})$  ein Teilraum der Menge

$$\left\{ f = (f_\alpha)_{|\alpha| \leq q} = (D^\alpha u)_{|\alpha| \leq q} : u \in W^{q+\delta}(\bar{\Omega}), Lu=0 \text{ in } \Omega \right\}.$$

$D(\partial\Omega) = r_{\partial\Omega} D(\bar{\Omega})$  sei die Menge aller Einschränkungen von  $u \in D(\bar{\Omega})$  auf den Rand ( $r_{\partial\Omega}$  Einschränkungoperator).

**Lemma 2.1:** Die Bedingung I und Voraussetzung (iii) seien erfüllt. Dann sind die Normen

$$\|u\|_{W^{q+\delta}(\partial\Omega)} \quad \text{und} \quad \sum_{j=1}^m \|B_j u\|_{C^{q-m_j+\delta}(\partial\Omega)}$$

in  $D(\partial\Omega)$  äquivalent.

Beweis: Aus Bedingung I folgt

$$\|u\|_{W^{q+\epsilon}(\partial\Omega)} \leq \|u\|_{C^{q+\epsilon}(\bar{\Omega})} \leq C \sum_{j=1}^m \|B_j u\|_{C^{q-m_j+\epsilon}(\partial\Omega)}$$

Umgekehrt ergibt sich mit Hilfe von (iii) durch direkte Abschätzung

$$\sum_{j=1}^m \|B_j u\|_{C^{q-m_j+\epsilon}(\partial\Omega)} \leq C_1 \|u\|_{C^{q+\epsilon}(\partial\Omega)}$$

Offensichtlich kann man  $\|B_j u\|_{C^{q-m_j+\epsilon}(\partial\Omega)}$  durch

$\|B_j u\|_{W^{q-m_j+\epsilon}(\partial\Omega)}$  ersetzen. Davon werden wir häufig Gebrauch

machen.

Wir wollen nun in dem oben formulierten Randwertproblem die Randbedingungen wie folgt präzisieren.

Es sei ein Taylorfeld  $f = (f_\alpha)_{|\alpha| \leq q} \in W^{q+\epsilon}(\partial\Omega)$  vorgegeben.

Wir konstruieren uns daraus die Funktionen

$$\varphi_{j,\alpha} = D^\alpha(B_j f) \quad (j=1, \dots, m; |\alpha| \leq q-m_j).$$

Hierbei stehen in den Ausdrücken  $\varphi_{j,\alpha}$  an Stelle der partiellen Ableitungen die entsprechenden Komponenten des  $W$ -Taylorfeldes  $f$ .

Das Randwertproblem kann damit in folgender Form behandelt werden.

Gesucht ist eine Funktion  $u \in C^{q+\epsilon}(\bar{\Omega})$ , die in  $\Omega$  der Differentialgleichung  $Lu=0$  genügt und für  $y \in \partial\Omega$  die Randwerte

$$D^\alpha(B_j u)(y) = \varphi_{j,\alpha}(y) \quad (j=1, \dots, m; |\alpha| \leq q-m_j)$$

stetig annimmt.

Das folgende Lemma beweist man analog zu Satz 3.1 in /6/.

Lemma 2.2:  $K = \partial\Omega$  sei  $(q+\epsilon)$ -regulär. Dann existieren zu jeder

stetigen Linearform  $L$  auf  $W^{q+\epsilon}(K)$  Maßstapel

$$v_j = (v_j^\alpha)_{|\alpha| \leq q-m_j} \quad \text{und} \quad \mathcal{K}_j = (\mathcal{K}_j^\alpha)_{|\alpha| = q-m_j}$$

mit  $\text{supp } v_j \subset K$ ,  $\text{supp } \mathcal{K}_j \subset K \times K$  ( $j=1, \dots, m$ ), so daß

$$L(f) = \sum_{j=1}^m \left\{ \sum_{|\alpha| \leq q-m_j} \varphi_{j,\alpha}(x) d v_j^\alpha(x) + \sum_{|\alpha|=q-m_j} \left( \frac{\varphi_{j,\alpha}(x) - \varphi_{j,\alpha}(y)}{|x-y|^\epsilon} d \mathcal{K}_j^\alpha(x,y) \right) \right\}.$$

Dabei ist  $f=(f_\alpha)_{|\alpha| \leq q} \in W^{q+\epsilon}(K)$ ,  $\varphi_{j,\alpha} = D^\alpha(B_j f)$ ,

$$\text{supp } v_j = \bigcup_{\alpha} \text{supp } v_j^\alpha, \quad \text{supp } \mathcal{K}_j = \bigcup_{\alpha} \text{supp } \mathcal{K}_j^\alpha.$$

Es sei betont, daß die angegebene Darstellung nicht eindeutig ist.

### 3. Balayage-Prinzip

Wir beziehen uns jetzt auf das in 2. formulierte Randwertproblem und beweisen eine der Bedingung I und dem Raum  $D(\bar{\Omega})$  angepaßte sogenannte Balayage-Beziehung.

Satz 3.1: Die Bedingung I sei erfüllt und  $\partial\Omega$  sei  $(q+\epsilon)$ -regulär.  $\mu = (\mu^\alpha)_{|\alpha| \leq q}$  und  $\lambda = (\lambda^\alpha)_{|\alpha|=q}$  seien Maßstapel mit

$$\text{supp } \mu \subset \bar{\Omega} \text{ bzw. } \text{supp } \lambda \subset \bar{\Omega} \times \bar{\Omega}.$$

Dann existieren weitere Maßstapel  $v_j = (v_j^\alpha)_{|\alpha| \leq q-m_j}$  und

$$\mathcal{K}_j = (\mathcal{K}_j^\alpha)_{|\alpha|=q-m_j} \quad (j = 1, \dots, m) \text{ mit } \text{supp } v_j \subset \partial\Omega \text{ bzw.}$$

$\text{supp } \mathcal{K}_j \subset \partial\Omega \times \partial\Omega$  ( $j = 1, \dots, m$ ), so daß gilt

$$\begin{aligned}
& \sum_{|\alpha| \leq q} \int D^\alpha u(x) \, d\mu^\alpha(x) + \sum_{|\alpha| = q} \int \frac{D^\alpha u(x) - D^\alpha u(y)}{|x-y|^6} \, d\lambda^\alpha(x, y) \\
&= \sum_{j=1}^m \left\{ \sum_{|\alpha| \leq q-m_j} \int D^\alpha (B_j u)(x) \, d\nu_j^\alpha(x) \right. \\
& \left. + \sum_{|\alpha| = q-m_j} \int \frac{D^\alpha (B_j u)(x) - D^\alpha (B_j u)(y)}{|x-y|^6} \, d\mathfrak{K}_j^\alpha(x, y) \right\} \quad (1)
\end{aligned}$$

für alle  $u \in D(\bar{\Omega})$ .

Wir schreiben zur Abkürzung

$$(\mu, \lambda)(u) = \sum_{j=1}^m (\nu_j, \mathfrak{K}_j)(u) \text{ für alle } u \in D(\bar{\Omega})$$

und setzen

$$\sum_{j=1}^m (\nu_j, \mathfrak{K}_j) = \Pi(\mu, \lambda).$$

Beweis: Wegen Bedingung I und Lemma 2.1 bildet  $r_{\partial\Omega}$  den Raum  $D(\bar{\Omega})$  linear und homöomorph auf  $D(\partial\Omega)$  ab. Durch

$$(\mu, \lambda)(u) = (\mu, \lambda) r_{\partial\Omega}^{-1}(r_{\partial\Omega} u)$$

ist daher ein stetiges lineares Funktional auf  $D(\partial\Omega)$  erklärt, das wir uns nach dem Satz von Hahn-Banach auf  $W^{q+6}(\partial\Omega)$  fortgesetzt denken. Nach Lemma 2.2 existieren Maße

$$\nu_j = (\nu_j^\alpha)_{|\alpha| \leq q-m_j} \text{ und } \mathfrak{K}_j = (\mathfrak{K}_j^\alpha)_{|\alpha| = q-m_j} \quad (j=1, \dots, m)$$

auf  $\partial\Omega$  bzw.  $\partial\Omega \times \partial\Omega$  mit

$$(\mu, \lambda) r_{\partial\Omega}^{-1}(r_{\partial\Omega} u) = \sum_{j=1}^m (\nu_j, \mathfrak{K}_j)(r_{\partial\Omega} u) = \sum_{j=1}^m (\nu_j, \mathfrak{K}_j)(u),$$

so daß

$$(\mu, \lambda)(u) = \sum_{j=1}^m (\nu_j, \mathfrak{K}_j)(u) \text{ für alle } u \in D(\bar{\Omega}),$$

w. s. z. w.

$\mathcal{M}$  heißt Balayage-Operator. Die Maße  $\nu_j$  und  $\lambda_j$  sind im allgemeinen nicht eindeutig bestimmt.

Durch Spezialisierung erhalten wir aus Satz 3.1 wichtige Darstellungsformeln für die Funktionen  $u \in D(\bar{\Omega})$  und ihre Ableitungen.

Satz 3.2: Bedingung I sei erfüllt und  $\partial\Omega$  sei  $(q+\epsilon)$ -regulär. Zu beliebigem  $x \in \Omega$  und  $|\alpha| \leq q$  existieren Maßstapel

$$\mu_{x, \alpha, j} = (\mu_{x, \alpha, j}^\beta)_{|\beta| \leq q - m_j} \quad (\text{supp } \mu_{x, \alpha, j} \subset \partial\Omega) \text{ und}$$

$$\lambda_{x, \alpha, j} = (\lambda_{x, \alpha, j}^\beta)_{|\beta| = q - m_j} \quad (\text{supp } \lambda_{x, \alpha, j} \subset \partial\Omega \times \partial\Omega),$$

so daß gilt

$$\begin{aligned} D^\alpha u(x) = & \sum_{j=1}^m \left\{ \sum_{|\beta| \leq q - m_j} \int D^\beta (B_j u)(y) d\mu_{x, \alpha, j}^\beta(y) + \right. \\ & \left. + \sum_{|\beta| = q - m_j} \int \frac{D^\beta (B_j u)(y) - D^\beta (B_j u)(z)}{|y - z|^\epsilon} d\lambda_{x, \alpha, j}^\beta(y, z) \right\} \end{aligned} \quad (2)$$

für alle  $u \in D(\bar{\Omega})$ .

Ferner existieren zu beliebigem  $x \in \bar{\Omega}$ ,  $y \in \bar{\Omega}$  ( $x \neq y$ ) und  $|\alpha| = q$  Maßstapel

$$\mu_{x, y, \alpha, j} = (\mu_{x, y, \alpha, j}^\beta)_{|\beta| \leq q - m_j} \quad (\text{supp } \mu_{x, y, \alpha, j} \subset \partial\Omega)$$

$$\text{und } \lambda_{x, y, \alpha, j} = (\lambda_{x, y, \alpha, j}^\beta)_{|\beta| = q - m_j} \quad (\text{supp } \lambda_{x, y, \alpha, j} \subset \partial\Omega \times \partial\Omega),$$

so daß gilt

$$\begin{aligned} \frac{D^\alpha u(x) - D^\alpha u(y)}{|x - y|^\epsilon} = & \sum_{j=1}^m \left\{ \sum_{|\beta| \leq q - m_j} \int D^\beta (B_j u)(z) d\mu_{x, y, \alpha}^\beta(z) + \right. \\ & \left. + \sum_{|\beta| = q - m_j} \int \frac{D^\beta (B_j u)(z) - D^\beta (B_j u)(s)}{|z - s|^\epsilon} d\lambda_{x, y, \alpha, j}^\beta(z, s) \right\} \end{aligned} \quad (3)$$

für alle  $u \in D(\bar{\Omega})$ .

**Beweis:** Wir setzen in Satz 3.1  $(\mu, \lambda) = (\delta_x, \alpha, 0) =: L_{x, \alpha}$ ,

wobei  $\delta_x, \alpha = (\mu^\beta)_{|\beta| \leq q}$  mit

$$\mu^\beta = \begin{cases} 0 & \text{für } \beta \neq \alpha \\ \delta_x & \text{für } \beta = \alpha \end{cases} \quad (\delta_x \text{ Dirac-Maß in } x \in \bar{\Omega}).$$

Nach Satz 3.1 existieren dann Maßstapel  $\mu_{x, \alpha, j}$  und  $\lambda_{x, \alpha, j}$  auf  $\partial\Omega$  bzw.  $\partial\Omega \times \partial\Omega$  mit

$$\Pi(\delta_x, \alpha, 0) = \sum_{j=1}^m (\mu_{x, \alpha, j}, \lambda_{x, \alpha, j}).$$

Setzt man diese Maße in (1) ein, so ergibt sich (2).

Setzt man entsprechend  $(\mu, \lambda) = (0, \delta_{x, y}, \alpha) =: L_{x, y, \alpha}$  mit

$\delta_{x, y}, \alpha = (\lambda^\beta)_{|\beta| = q}$  und

$$\lambda^\beta = \begin{cases} 0 & \text{für } \beta \neq \alpha \\ \delta_{x, y} & \text{für } \beta = \alpha \end{cases}$$

$(\delta_{x, y}$  Dirac-Maß in  $(x, y) \in \bar{\Omega} \times \bar{\Omega})$ ,

so liefert  $\Pi(0, \delta_{x, y}, \alpha) = \sum_{j=1}^m (\mu_{x, y, \alpha, j}, \lambda_{x, y, \alpha, j})$

und Einsetzen in (1) die Beziehung (3).

Die Maßstapel  $\Lambda_{x, \alpha} := (\mu_{x, \alpha, j}, \lambda_{x, \alpha, j})$  sowie

$$\Lambda_{x, y, \alpha} := (\mu_{x, y, \alpha, j}, \lambda_{x, y, \alpha, j})$$

wollen wir dem betrachteten Randwertproblem zugeordnete harmonische Maße nennen.

4. Lösung des Problems für Randwerte, die durch ein beliebiges Taylorfeld  $f \in W^{q+\epsilon}(\partial\Omega)$  erzeugt werden.

Mit  $\|\Lambda_{x,\alpha}\|_{\partial\Omega}$  bzw.  $\|\Lambda_{x,y,\alpha}\|_{\partial\Omega}$  bezeichnen wir die Norm der durch  $\Lambda_{x,\alpha}$  bzw.  $\Lambda_{x,y,\alpha}$  gemäß Lemma 2.2 auf  $W^{q+\epsilon}(\partial\Omega)$  gegebenen stetigen Linearformen.

**Lemma 4.1:**  $\partial\Omega$  sei  $(q+\epsilon)$ -regulär und  $\Omega$  erfülle die Bedingung I sowie (iii). Durchläuft  $x$  ganz  $\Omega$ , so existieren für alle  $\alpha$  mit  $|\alpha| \leq q$  ausgezeichnete Scharen harmonischer Maße  $\Lambda_{x,\alpha}$ , für welche die Normen  $\|\Lambda_{x,\alpha}\|_{\partial\Omega}$  bezüglich  $x$  gleichmäßig beschränkt sind.

Entsprechendes gilt für  $\|\Lambda_{x,y,\alpha}\|_{\partial\Omega}$ , falls  $x$  und  $y$  ( $x \neq y$ )  $\Omega$  durchläuft.

**Beweis:** Wegen Bedingung I und Lemma 2.1 gilt für  $u \in D(\bar{\Omega})$  die Ungleichung

$$\begin{aligned} |L_{x,\alpha}(u)| &\leq \|L_{x,\alpha}\| \|u\|_{W^{q+\epsilon}(\bar{\Omega})} \\ &\leq \|L_{x,\alpha}\| C \sum_{j=1}^m \|B_j u\|_{C^{q-m_j+\epsilon}(\partial\Omega)} \quad (4) \\ &\leq \|L_{x,\alpha}\| C_1 \|u\|_{W^{q+\epsilon}(\partial\Omega)}. \end{aligned}$$

$\hat{L}_{x,\alpha}(r_{\partial\Omega} u) := L_{x,\alpha}(u)$  definiert also ein stetiges lineares Funktional auf  $D(\partial\Omega)$ .

Nach dem Satz von Hahn-Banach kann  $\hat{L}_{x,\alpha}$  unter Erhaltung der Norm zu einem stetigen linearen Funktional  $\Lambda_{x,\alpha}$  auf  $W^{q+\epsilon}(\partial\Omega)$  fortgesetzt werden.

Wegen  $\Lambda_{x,\alpha}(u) = L_{x,\alpha}(u)$  für alle  $u \in D(\bar{\Omega})$  ist  $\Lambda_{x,\alpha}$  harmonisches Maß, und es gilt  $\|\Lambda_{x,\alpha}\|_{\partial\Omega} = \|\hat{L}_{x,\alpha}\|$ .

Ferner ist  $|\Lambda_{x,\alpha}(u)| \leq \|\Lambda_{x,\alpha}\|_{\partial\Omega} \|u\|_{W^{q+\epsilon}(\partial\Omega)}$  für alle  $u \in W^{q+\epsilon}(\partial\Omega)$ . Der Vergleich mit (4) liefert

$$\|\hat{L}_{x,\alpha}\| = \|\Lambda_{x,\alpha}\|_{\partial\Omega} \leq C_1 \|L_{x,\alpha}\|.$$

Wegen  $|L_{x,\alpha}(u)| = |D^\alpha u(x)| \leq \|u\|_{W^{q+\epsilon}(\bar{\Omega})}$  erhalten wir weiter

$$\|L_{x,\alpha}\| \leq 1 \text{ und damit } \|\Lambda_{x,\alpha}\|_{\partial\Omega} \leq C_1.$$

Der Beweis des zweiten Teiles der Aussage verlauft analog.

$\|L_{x,y,\alpha}\| \leq 1$  folgt aus

$$|L_{x,y,\alpha}(u)| = \frac{|D^\alpha u(x) - D^\alpha u(y)|}{|x-y|^\epsilon} \leq \|u\|_{W^{q+\epsilon}(\bar{\Omega})}.$$

Es sei  $D_B(\partial\Omega) = \sum_{j=1}^m \left\{ (D^\alpha B_j u)_{|\alpha| \leq q-m_j} : u \in D(\partial\Omega) \right\}$  und

$$W_B(\partial\Omega) = \sum_{j=1}^m \left\{ (D^\alpha B_j f)_{|\alpha| \leq q-m_j} : f \in W^{q+\epsilon}(\partial\Omega) \right\}.$$

#### Bedingung II

$D_B(\partial\Omega)$  sei bzgl.  $\sum_{j=1}^m \|B_j u\|_{C^{q-m_j+\epsilon}}$  dicht in  $W_B(\partial\Omega)$

Bemerkung: Die stetige Linearform  $\sum_{j=1}^m (\psi_j, \chi_j)$  in Satz 3.1

(Balayage-Prinzip) ist genau dann eindeutig bestimmt, wenn das Gebiet  $\Omega$  der Bedingung II genugt. Insbesondere sind in diesem Falle die harmonischen Mae  $\Lambda_{x,\alpha}$  ( $|\alpha| \leq q$ ,  $x \in \Omega$ ) und

$\Lambda_{x,y,\alpha}$  ( $|\alpha| = q$ ,  $x \in \Omega$ ,  $y \in \Omega$ ,  $x \neq y$ ) eindeutig bestimmt und

Lemma 4.1 sagt aus, da ihre Normen gleichmaig beschrankt sind. Der Beweis dieser Bemerkung ist unmittelbar aus dem Beweis von Satz 3.1 ersichtlich.

Satz 4.1: Der Rand  $\partial\Omega$  des Gebietes  $\Omega$  sei  $(q+\epsilon)$ -regular und die Bedingungen I und II seien erfullt. Dann ist das obige Randwertproblem eindeutig losbar und die Losung  $u$  sowie die Ableitungen  $D^\alpha u$  besitzen fur  $x \in \Omega$  die Darstellung

$$D^\alpha u(x) = \sum_{j=1}^m \left\{ \sum_{|\beta| \leq q-m_j} \int \varphi_{j,\beta}(y) d\mu_{x,\alpha,j}^\beta(y) \right. \\ \left. + \sum_{|\beta| = q-m_j} \int \frac{\varphi_{j,\beta}(y) - \varphi_{j,\beta}(z)}{|y-z|^\beta} d\lambda_{x,\alpha,j}^\beta(y,z) \right\} \quad (5)$$

$\Lambda_{x,\alpha} = (\mu_{x,\alpha,j}, \lambda_{x,\alpha,j})$  bezeichnet dabei die nach

der voranstehenden Bemerkung eindeutig bestimmte Schar harmonischer Maße.

**Beweis:** Wir betrachten zunächst ein Taylorfeld

$f = (f_\alpha)_{|\alpha| \leq q} \in D(\partial\Omega)$ . Dann existiert per def. eine Funktion

$u \in D(\bar{\Omega})$  mit  $D^\alpha(B_j u|_{\partial\Omega}) = \varphi_{j,\alpha}$  ( $j=1, \dots, m; |\alpha| \leq q-m_j$ ),

die die eindeutige Lösung des Problems darstellt. Die Gültigkeit von (5) folgt aus der Tatsache, daß  $\Lambda_{x,\alpha}$  durch Balayage von  $L_{x,\alpha}$  erzeugt ist (Satz 3.2).

Nunmehr sei  $f = (f_\alpha)_{|\alpha| \leq q}$  ein beliebiges Element aus  $W^{q+\epsilon}(\partial\Omega)$ .

Dann existiert wegen Bedingung II eine Folge

$$u^{(n)} = (u_\alpha^{(n)})_{|\alpha| \leq q} \in D(\partial\Omega)$$

mit

$$\sum_{j=1}^m \|B_j(f - u^{(n)})\|_{W^{q-m_j+\epsilon}(\partial\Omega)} \rightarrow 0 \text{ für } n \rightarrow \infty \quad (6)$$

Offenbar ist  $u_\alpha^{(n)}(x) = D^\alpha u^{(n)}(x)$  für  $x \in \bar{\Omega}$ , wenn wir die entsprechenden Funktionen aus  $D(\bar{\Omega})$  ebenfalls mit  $u^{(n)}$  bezeichnen. Wir setzen für  $|\alpha| \leq q$

$$u_\alpha(x) = \sum_{j=1}^m \left\{ \sum_{|\beta| \leq q-m_j} \int D^\beta(B_j f)(y) d\mu_{x,\alpha,j}^\beta(y) \right. \\ \left. + \sum_{|\beta| = q-m_j} \int \frac{D^\beta(B_j f)(y) - D^\beta(B_j f)(z)}{|y-z|^\beta} d\lambda_{x,\alpha,j}^\beta(y,z) \right\}$$

und bilden die Differenz

$$u_{\alpha}(x) - D^{\alpha}u^{(n)}(x)$$

$$= \sum_{j=1}^m \left\{ \sum_{|\beta| \leq q - m_j} \int [D^{\beta}(B_j f)(y) - D^{\beta}(B_j u^{(n)})(y)] d\mu_{x, \alpha, j}^{\beta}(y) \right. \\ \left. + \sum_{|\beta| = q - m_j} \int \frac{[(D^{\beta}(B_j f) - D^{\beta}(B_j u^{(n)}))(y) - (D^{\beta}(B_j f) - D^{\beta}(B_j u^{(n)}))(z)]}{|y - z|^{\epsilon}} d\lambda_{x, \alpha, j}^{\beta}(y, z) \right\}$$

Daraus folgt

$$u_{\alpha}(x) - D^{\alpha}u^{(n)}(x) = \Lambda_{x, \alpha}(f - u^{(n)})$$

und aus Lemma 4.1 für  $|\alpha| \leq q$

$$|u_{\alpha}(x) - D^{\alpha}u^{(n)}(x)| \leq C \sum_{j=1}^m \|B_j(f - u^{(n)})\|_{W^{q - m_j + \epsilon}(\partial\Omega)}$$

Diese Abschätzung gilt gleichmäßig bezüglich  $x$ . Wegen (6) konvergieren also die Folgen  $(D^{\alpha}u^{(n)})$  für  $n \rightarrow \infty$  gleichmäßig in  $\Omega$  gegen  $u_{\alpha}$ . Setzt man entsprechend für  $|\alpha| = q$

$$h_{\alpha}(x, y) = \sum_{j=1}^m \left\{ \sum_{|\beta| \leq q - m_j} \int D^{\beta}(B_j f)(z) d\mu_{x, y, \alpha, j}^{\beta}(z) \right. \\ \left. + \sum_{|\beta| = q - m_j} \int \frac{D^{\beta}(B_j f)(z) - D^{\beta}(B_j f)(s)}{|z - s|^{\epsilon}} d\lambda_{x, y, \alpha, j}^{\beta}(z, s) \right\} = \\ = \Lambda_{x, y, \alpha}(f)$$

und berücksichtigt

$$\frac{D^{\alpha}u^{(n)}(x) - D^{\alpha}u^{(n)}(y)}{|x - y|^{\epsilon}} = \sum_{j=1}^m \left\{ \sum_{|\beta| \leq q - m_j} \int D^{\beta}(B_j u^{(n)})(z) d\mu_{x,y,\alpha}^{\beta}(z) \right. \\ \left. + \sum_{|\beta| = q - m_j} \int \frac{D^{\beta}(B_j u^{(n)})(z) - D^{\beta}(B_j u^{(n)})(s)}{|z - s|^{\epsilon}} d\lambda_{x,y,\alpha,j}^{\beta}(z,s) \right\} = \\ = \Lambda_{x,y,\alpha}(u^{(n)}),$$

so ergibt sich aus der gleichmäßigen Beschränktheit von  $\|\Lambda_{x,y,\alpha}\|$  (Lemma 4.1) weiter, daß

$$\frac{D^{\alpha}u^{(n)}(x) - D^{\alpha}u^{(n)}(y)}{|x - y|^{\epsilon}}$$

in  $\bar{R} \times \bar{R}$  gleichmäßig gegen  $h_{\alpha}(x,y)$  konvergiert.

Zusammengefaßt bedeutet dies, daß die Folge der Funktionen  $u^{(n)} \in D(\bar{R})$  im Sinne der Norm von  $W^{q+\epsilon}(\bar{R})$  konvergiert, wobei (mit der abkürzenden Bezeichnung  $u(0, \dots, 0) = u$ ) aus der Vollständigkeit von  $W^{q+\epsilon}(\bar{R})$  folgt:

$$u_{\alpha}(x) = D^{\alpha}u(x) \text{ für } |\alpha| \leq q \text{ und}$$

$$h_{\alpha}(x,y) = \frac{D^{\alpha}u(x) - D^{\alpha}u(y)}{|x - y|^{\epsilon}} \text{ für } |\alpha| = q.$$

Die Funktion  $u$  ist als gleichmäßiger Grenzwert von Lösungen in  $\Omega$  ebenfalls Lösung von  $Lu = 0$ . Die Annahme der vorgegebenen Randwerte folgt unmittelbar aus der Konstruktion, die Eindeutigkeit der Lösung aus Bedingung II.

**Bemerkung:** Die voranstehenden Ausführungen zeigen, daß die konkrete Natur des Raumes  $D(\bar{R})$  nicht entscheidend ist. Das Problem besteht darin, geeignete Teilräume  $D \subset W^{q+\epsilon}(\bar{R})$  von Lösungen der Gleichung  $Lu = 0$  zu finden, die den Bedingungen I und II genügen (vgl. auch die Arbeit /7/). In /4/ geht man im Falle des Dirichlet-Problems unter Verwendung der Agmon-Miranda-Abschätzungen zum Beispiel von geeigneten Räumen von Potentialen aus. Es liegt nahe, dies auch im vorliegenden Fal-

le zu versuchen.

Ferner sei darauf hingewiesen, daß die  $(q+\epsilon)$ -Regularität lediglich zur Darstellung der Funktionale  $\Lambda_{x,\alpha}$  usw. benötigt wird. Ohne diese Voraussetzung lautet z. B. die Darstellungsformel (5)

$$D^\alpha u(x) = \Lambda_{x,\alpha}(f) .$$

### Literatur

- /1/ Agmon, S., Maximum theorems for solutions of higher order elliptic equations.  
Bull. Am. math. Soc. 66, 77-80 (1960)
- /2/ Agmon, S.; Douglis, A.; Nirenberg, L., Estimates near the boundary for solutions of elliptic differential equations satisfying general boundary conditions.  
Comm. pure applied math. 12, 623-727 (1959)
- /3/ Miranda, G., Teorema del massimo modulo e teorema di esistenza e di unicità per il problema di Dirichlet relative alle equazioni ellittiche in due variabili.  
Ann. Math. Pura Appl. 46, 265-311 (1958)
- /4/ Schulze, B. W.; Wildenhain, G., Methoden der Potentialtheorie für elliptische Differentialgleichungen beliebiger Ordnung.  
Berlin, 1977
- /5/ Wildenhain, G., Potentialtheorie linearer elliptischer Differentialgleichungen beliebiger Ordnung.  
Berlin 1968

- /6/ Wildenhain, G., Der Raum der  $H$ -stetig differenzierbaren Funktionen.  
Math. Nachr. 50, 217-228 (1971)
- /7/ Wildenhain, G., Ein funktionalanalytischer Zugang zum Dirichlet-Problem bei elliptischen Differentialgleichungen mit variablen Koeffizienten.  
Math. Nachr. 56, 247-259 (1973)

eingegangen: 14. 7. 1977

Anschrift der Verfasser:

Dipl. Math. Harald Mumm  
Prof. Dr. rer. nat. habil. Günter Wildenhain  
Wilhelm-Pieck-Universität Rostock  
Sektion Mathematik  
DDR-25 Rostock  
Universitätsplatz 1

### Approximation durch biharmonische Potentiale

Bei der potentialtheoretischen Behandlung des Dirichlet-Problems der Gleichung  $\Delta \Delta u = 0$  spielen Fragen der Dichtigkeit biharmonischer Potentiale in gewissen Räumen stetig differenzierbarer Funktionen eine wichtige Rolle (vgl. /3/, /4/). Derartige Dichtheitsaussagen kann man in verschiedener Weise gewinnen. In /2/ war für den Fall der Ebene ein konstruktives Verfahren zur Erzeugung dichter Mengen von Potentialen angegeben worden. Dieses Verfahren soll in der vorliegenden Mitteilung auf den  $n$ -dimensionalen Fall verallgemeinert werden. Wir stellen zunächst die benutzten Begriffsbildungen und Bezeichnungen zusammen.

Es seien  $x = (x_1, \dots, x_n)$ ,  $z = (z_1, \dots, z_n)$  Punkte des  $\mathbb{R}^n$ ,  $|x - z|$  der euklidische Abstand von  $x$  und  $z$  und

$$\phi(x, z) = |x - z|^{4-n} \text{ für } n = 3 \text{ und } n > 4,$$

$$\phi(x, z) = |x - z|^{4-n} \ln|x - z| \text{ für } n = 2 \text{ und } n = 4$$

die Fundamentallösung des biharmonischen Operators (abgesehen von einem konstanten Faktor).  $\phi_i(x, z)$  ( $i=1, \dots, n$ ) bezeichne die sogenannten Ableitungskerne, die sich durch Differentiation von  $\phi$  nach der ersten Variablen ergeben.

Wir betrachten Tupel  $\mu = (\mu_0, \mu_1, \dots, \mu_n)$  Radonscher Maße mit kompaktem Träger (d. h. Distributionen der Ordnung 1) und definieren damit biharmonische Potentiale

$$\phi_\mu(x) = \int \phi(x, z) d\mu_0(z) + \sum_{i=1}^n \int \phi_i(x, z) d\mu_i(z). \quad (1)$$

Außerhalb  $\text{supp } \mu := \bigcup_{i=0}^n \text{supp } \mu_i$  ist  $\phi_\mu$  Lösung der Gleichung

$$\Delta \Delta u = 0. \quad (2)$$

$C_0^1(\mathbb{R}^n)$  bezeichne den Raum der stetig differenzierbaren Funktionen mit kompaktem Träger. Bei der Anwendung der biharmonischen Potentiale (1) zum Studium des Dirichlet-Problems der Gleichung (2) ist der folgende Dichtheitsatz von Interesse.

**Satz:** Es sei  $f \in C_0^1(\mathbb{R}^n)$  mit  $\text{supp } f =: K$  gegeben. Dann existiert zu beliebigem  $\varepsilon > 0$  und beliebig vorgegebener (abgeschlossener) Umgebung  $V \supset K$  ein biharmonisches Potential  $\phi_\mu(x)$  mit  $\text{supp } \mu \subset V$ , welches stetig nach allen Variablen differenzierbar ist, außerhalb  $V$  identisch verschwindet und folgender Bedingung genügt:

$$\|f - \phi_\mu\| := \sup_{x \in V} |f(x) - \phi_\mu(x)| + \sum_{i=1}^n \sup_{x \in V} \left| \frac{\partial f(x)}{\partial x_i} - \frac{\partial \phi_\mu(x)}{\partial x_i} \right| < \varepsilon.$$

Für die im folgenden dargelegte spezielle Beweis konstruktion benötigen wir zwei Lemmata, die wir ohne Beweis angeben.

**Lemma 1:**  $H$  sei eine Teilmenge des Raumes  $C_0^1(\mathbb{R}^n)$  mit folgenden Eigenschaften:

1. Es ist  $\varphi \geq 0$  für alle  $\varphi \in H$ .
2. Zu jeder Nullumgebung  $U$  existiert eine Funktion  $\varphi \neq 0$ ,  $\varphi \in H$  mit  $\text{supp } \varphi \subset U$ .
3.  $H$  ist translationsinvariant, d. h., mit  $\varphi(x) \in H$  ist auch  $\varphi(x-x_0) \in H$  für beliebiges  $x_0 \in \mathbb{R}^n$ .

Dann existiert zu gegebenem  $f \in C_0^1(\mathbb{R}^n)$  mit  $\text{supp } f =: K$ , zu einer beliebigen (abgeschlossenen) Umgebung  $V \supset K$  und zu beliebigem  $\varepsilon > 0$  eine endliche Linearkombination  $g$  von Funktionen aus  $H$  mit  $\text{supp } g \subset V$  und

$$\|f - g\| := \sup_{x \in V} |f(x) - g(x)| + \sum_{i=1}^n \sup_{x \in V} \left| \frac{\partial f}{\partial x_i} - \frac{\partial g}{\partial x_i} \right| < \varepsilon.$$

Zum Beweis dieses Lemmas verweisen wir auf /2/. Für den Beweis des folgenden Lemmas vergleiche man /1/.

**Lemma 2:** u sei eine innerhalb einer Kugel  $K(R, x_0)$  mit dem Radius  $R$  und dem Mittelpunkt  $x_0$  biharmonische Funktion, die einschließlich der ersten partiellen Ableitungen am Rande stetig ist. Dann besitzt u für  $x \in K(R, x_0)$  die folgende Darstellung:

$$u(x) = \frac{-(R^2 - r^2)^2}{2R^2 \omega_n} \int_{S(R, x_0)} \frac{h(y)}{|x-y|^n} dO(y) + \\ + \frac{(R^2 - r^2)^2}{2R^2 \omega_n} \int_{S(R, x_0)} \frac{2R^2 - (n-2)r^2 + (n-4)Rr \cos t_1}{|x-y|^{n+2}} g(y) dO(y).$$

Hierbei bezeichnet  $S(R, x_0)$  die Kugeloberfläche,  $dO(y)$  das Flächenelement bezüglich der Polarkoordinaten  $(R, t_1, \dots, t_{n-1})$ ,  $r = |x - x_0|$ ,  $\omega_n$  den Inhalt der n-dimensionalen Einheitskugeloberfläche und  $g(y)$  bzw.  $h(y)$  die Randwerte bzw. die Werte der Normalableitung von u bei nach außen orientierter Normalen.

**Beweis des Satzes:** Wir konstruieren eine aus Potentialen bestehende Teilmenge  $H \subset C_0^1(R^n)$  mit den in Lemma 1 angegebenen

Eigenschaften, woraus die Behauptung unmittelbar folgt. Es sei  $U$  eine beliebige Umgebung des Koordinatenursprungs und  $K_1 = K(r_1, 0)$ ,  $K_2 = K(r_2, 0)$  ( $r_2 < r_1$ ) seien in  $U$  enthaltene, abgeschlossene, konzentrische Kugeln mit den Rändern  $S_1$  bzw.  $S_2$ .

Für festes  $z \in R^n \setminus K_1$  ist dann  $\phi(x, z)$  innerhalb  $K_1$  bezüglich  $x$  eine am Rande  $S_1$  stetig differenzierbare biharmonische Funktion, die sich folglich für  $x \in K_1$  in der in Lemma 2 angegebenen Gestalt darstellen läßt.

Setzt man speziell  $x = 0$ , so ergibt sich

$$\begin{aligned} \phi(0, z) = & - \frac{r_1^{2-n}}{2\omega_n} \sum_{i=1}^n \int_{S_1} \phi_i(y, z) \cos(n, x_i) dO(y) \\ & + \frac{1}{\omega_n} \int_{S_1} \phi(y, z) dO(y). \end{aligned} \quad (3)$$

$\cos(n, x_i)$  ( $i=1, \dots, n$ ) bezeichnet die Richtungskosinus zwischen der Normalen an die Kugel im Punkt  $y$  und der  $x_i$ -Achse. Die rechte Seite von (3) definiert bezüglich der Variablen  $z$  ein biharmonisches Potential  $\phi_{\mu^1}$  mit  $\text{supp } \mu^1 = S_1$ . Führt man dieselbe Betrachtung für die Kugel  $K_2$  durch, erhält man entsprechend ein Potential  $\phi_{\mu^2}$  mit  $\text{supp } \mu^2 = S_2$ . Nach Konstruktion gilt  $\phi_{\mu^1} \equiv \phi_{\mu^2}$  in  $R^n \setminus K_1$ .

$$\text{Setzt man } \phi_{\mu} := \phi_{\mu^2} - \phi_{\mu^1} \quad (4)$$

$$\text{oder } \phi_{\mu} := \phi_{\mu^1} - \phi_{\mu^2}, \quad (5)$$

so ist  $\text{supp } \mu = S_1 \cup S_2 \subset U$  und  $\phi_{\mu} = 0$  in  $R^n \setminus K_1$ .

Wie in /2/ kann man sich davon überzeugen, daß  $\phi_{\mu}$  überall stetig differenzierbar ist. Im nächsten Schritt zeigen wir nacheinander in den Fällen  $n = 3$ ,  $n = 4$  und  $n \geq 5$ , daß entweder die Potentiale (4) oder (5) für alle  $z \in R^n$  nicht negativ sind, wenn man gegebenenfalls noch geeignet über die Wahl des Verhältnisses  $r_2/r_1$  verfügt.

Es sei zunächst  $n = 3$ . Für  $z \in K_1$  können wir dann  $\phi_{\mu^1}$  gemäß Lemma 2 durch Integrale über die Randwerte darstellen. Nach Konstruktion gilt

$$\phi_{\mu^1} \Big|_{S_1} = r_1, \quad \frac{\partial \phi_{\mu^1}}{\partial n} \Big|_{S_1} = 1.$$

Mit  $|z| = r$  und unter Berücksichtigung von

$$\omega_n = \frac{2(2\pi)^{\frac{n-1}{2}}}{1 \cdot 3 \cdot 5 \cdots (n-2)} \quad \text{für } n \text{ ungerade} ,$$

$$\omega_n = \frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})} \quad \text{für } n \text{ gerade}$$

erhalten wir daher

$$\begin{aligned} \phi_{\mu^1}(z) &= \frac{(r_1^2 - r^2)^2}{8\pi r_1^3} \int_{S_1} \frac{r_1(2r_1^2 - r^2 - rr_1 \cos t_1)}{(r_1^2 + r^2 - 2rr_1 \cos t_1)^{5/2}} dO(y) \\ &\quad - \frac{(r_1^2 - r^2)^2}{8\pi r_1^2} \int_{S_1} \frac{dO(y)}{(r_1^2 + r^2 - 2rr_1 \cos t_1)^{3/2}} . \end{aligned} \quad (6)$$

Hierbei ist  $dO(y) = r_1^2 \sin t_1 dt_1 dt_2$ . für beliebiges  $n$  gilt

entsprechend  $dO(y) = r_1^{n-1} (\sin t_1)^{n-2} dt_1 dt_2 \cdots dt_{n-1}$ .

Da  $u \equiv 1$  die eindeutig bestimmte Lösung des Dirichlet-Problems in  $K_1$  der Gleichung  $\Delta u = 0$  mit den Randwerten

$$u|_{S_1} = 1 \quad , \quad \frac{\partial u}{\partial n}|_{S_1} = 0$$

ist, ergibt sich für den ersten Summanden in (6) sofort der Wert  $r_1$ . Die Berechnung des zweiten Integrals liefert

$$\phi_{\mu^1}(z) = \frac{r_1}{2} + \frac{r^2}{2r_1} = \phi_{\mu^1}(r) \quad (r \leq r_1) .$$

Analog erhält man

$$\phi_{\mu^2}(z) = \phi_{\mu^2}(r) = \frac{r_2}{2} + \frac{r^2}{2r_1} \quad \text{für } r \leq r_2 .$$

Wir betrachten die Differenz (5).

Zunächst sei  $z \in K_1 \setminus K_2$ . Für diese  $z$  gilt  $\phi_{\mu^2} = r$ , und damit

$$\phi_{\mu^1}(r) - \phi_{\mu^2}(r) = \frac{r_1}{2} + \frac{r^2}{2r_1} - r = f(r) .$$

Aus  $f(r_1) = 0$ ,  $f(0) = \frac{r_1}{2} > 0$ ,  $f'(r) = \frac{r}{r_1} - 1 < 0$  für  $0 \leq r < r_1$

erhalten wir  $f(r) \geq 0$ . Für  $z \in K_2$  folgt

$$\begin{aligned} \phi_{\mu^1}(r) - \phi_{\mu^2}(r) &= \frac{r_1}{2} + \frac{r^2}{2r_1} - \frac{r_2}{2} - \frac{r^2}{2r_2} \\ &= \frac{1}{2}(r_1 - r_2) \left(1 - \frac{r^2}{r_1 r_2}\right) > 0 \quad \text{für } r \leq r_2 . \end{aligned}$$

Für  $n=4$  liefert eine analoge Rechnung unter Beachtung von

$$\phi_{\mu^1} \Big|_{S_1} = \ln r_1 \quad , \quad \frac{\partial \phi_{\mu^1}}{\partial n} \Big|_{S_1} = \frac{1}{r_1}$$

die Darstellung

$$\phi_{\mu^1}(r) = \ln r_1 - \frac{r_1^2 - r^2}{2r_1^2} \quad \text{für } r \leq r_1 \quad ,$$

$$\phi_{\mu^2}(r) = \ln r_2 - \frac{r_2^2 - r^2}{2r_2^2} \quad \text{für } r \leq r_2 \quad .$$

Für  $z \in K_1 \setminus K_2$  ist  $\phi_{\mu^2}(r) = \ln r$  und

$$\phi_{\mu^1}(r) - \phi_{\mu^2}(r) = \ln \frac{r_1}{r} + \frac{r_1^2 - r^2}{2r_1^2} = f(r) \geq 0$$

wegen  $f(r_1) = 0$ ,  $\lim_{r \rightarrow 0} f(r) = +\infty$  und  $f'(r) = -\frac{1}{r} - \frac{r}{r_1^2} < 0$

für  $0 < r \leq r_1$ .

Im Falle  $z \in K_2$  folgt

$$\phi \mu^1(r) - \phi \mu^2(r) = \ln \frac{r_1}{r_2} + \frac{r^2(r_2^2 - r_1^2)}{2r_1^2 r_2^2}.$$

Dieser Ausdruck besitzt eine Nullstelle bei

$$r_0 = \left[ \frac{2r_2^2}{\left(\frac{r_2}{r_1}\right)^2 - 1} \ln \frac{r_2}{r_1} \right]^{1/2}.$$

Wählt man etwa  $\frac{r_2}{r_1} = \frac{1}{3}$ , so wird  $r_0 > r_2$ , und da

$$\phi \mu(0) = \ln \frac{r_1}{r_2} > 0 \text{ ist, bedeutet dies } \phi \mu(z) > 0 \text{ für}$$

$z \in K_2$ .

Nunmehr sei  $n \geq 5$ .

$$\phi \mu^1|_{S_1} = \frac{1}{r_1^{n-4}}, \quad \frac{\partial \phi \mu^1}{\partial n} \Big|_{S_1} = \frac{4-n}{r_1^{n-3}} \quad \text{und Berechnung der}$$

(6) entsprechenden Integrale ergibt

$$\phi \mu^1(r) = \frac{1}{r_1^{n-4}} - \frac{(r_1^2 - r^2)(4-n)}{2r_1^{n-2}} \quad (r \leq r_1),$$

$$\phi \mu^2(r) = \frac{1}{r_2^{n-4}} - \frac{(r_2^2 - r^2)(4-n)}{2r_2^{n-2}} \quad (r \leq r_2).$$

In diesem Falle bilden wir  $\phi \mu = \phi \mu^2 - \phi \mu^1$  und erhalten für  $z \in K_1 \setminus K_2$

$$\phi \mu = \frac{1}{r^{n-4}} - \frac{1}{r_1^{n-4}} + \frac{(r_1^2 - r^2)(4-n)}{2r_1^{n-2}} = f(r) \geq 0$$

(wegen  $f(r_1) = 0$ ,  $\lim_{r \rightarrow 0} f(r) = +\infty$ ,  $f'(r) = (n-4)\left(\frac{r}{r_1^{n-2}} - \frac{1}{r^{n-3}}\right)$   
 $< 0$  für  $r < r_1$ ).

Innerhalb  $K_2$  gilt

$$\phi_{\mu}^2 - \phi_{\mu}^1 = \frac{1}{r_2^{n-4}} - \frac{1}{r_1^{n-4}} - \frac{(r_2^2 - r^2)(4-n)}{2r_2^{n-2}} + \frac{(r_1^2 - r^2)(4-n)}{2r_1^{n-2}}.$$

Dieser Ausdruck besitzt eine Nullstelle bei

$$r_0 = \left[ \frac{n-2}{n-4} r_2^2 \left\{ \left( \frac{r_2}{r_1} \right)^{n-4} - 1 \right\} \right]^{\frac{1}{2}}.$$

Offenbar läßt sich für jedes  $n$  durch genügend kleine Wahl des Verhältnisses  $r_2/r_1$  erreichen, daß der Faktor

$$\frac{n-2}{n-4} \left\{ \dots \right\}$$

größer als 1, und damit  $r_0 > r_2$  ausfällt. Wegen

$$\phi_{\mu}(0) = \frac{n-2}{2} \left[ \frac{1}{r_2^{n-4}} - \frac{1}{r_1^{n-4}} \right] > 0$$

bedeutet dies  $\phi_{\mu} > 0$  in  $K_2$ .

Mit  $H$  werde nun die Menge der oben konstruierten Potentiale  $\phi_{\mu}$ , einschließlich aller daraus durch Verschiebung des Arguments entstehenden Potentiale der Gestalt

$$\phi_{\mu_{z_0}}(z) = \phi_{\mu}(z - z_0)$$

mit  $\text{supp } \mu_{z_0} = S(r_1, z_0) \cup S(r_2, z_0)$  bezeichnet.  $H$  erfüllt die

Voraussetzungen von Lemma 1. Die daraus folgende Behauptung ist gerade die Aussage des Satzes, da die Linearkombinationen  $g$  von Funktionen aus  $H$  ein Potential mit dem Träger in  $V$  definieren.

## Literatur

- /1/ Edenhofer, J. Integraldarstellung einer  $m$ -polyharmonischen Funktion, deren Funktionswerte und erste  $m-1$  Normalableitungen auf einer Hypersphäre gegeben sind,  
Math. Nachr. 68, 105 - 113 (1975)
- /2/ Wildenhain, G. Anwendung eines Approximationsatzes für stetig differenzierbare Funktionen auf die Bipotentialgleichung,  
Math. Nachr. 31, 137 - 152 (1966)
- /3/ Wildenhain, G. Potentialtheorie linearer elliptischer Differentialgleichungen beliebiger Ordnung,  
Berlin 1968
- /4/ Schulze, B.-W.,; Wildenhain, G.,  
Methoden der Potentialtheorie für elliptische Differentialgleichungen beliebiger Ordnung,  
Berlin 1977

eingegangen: 15. 9. 1977

## Anschrift der Verfasser:

Dipl.-Math. Ute Clasen  
Prof. Dr. rer. nat. habil. Günther Wildenhain  
Wilhelm-Pieck-Universität Rostock  
Sektion Mathematik  
DDR-25 Rostock  
Universitätsplatz 1



Diskrete lineare Approximation und zyklische Einzelschrittverfahren

---

Die direkte Anwendung des Gedankens der zyklischen Einzelschrittiteration auf diskrete lineare Approximationsaufgaben ist im allgemeinen mit einigen Schwierigkeiten verbunden, insbesondere mit der Problematik des vorzeitigen Festlaufens. In /1/ wurde diese Problematik hinsichtlich der diskreten Tschebyscheff-Approximation untersucht. Es wurde eine Repräsentantenmenge definiert und gezeigt, daß die zyklische Minimierung bezüglich dieser Repräsentantenmenge ein konvergentes Verfahren darstellt. Dieses Verfahren kann aber zunächst nur von theoretischem Interesse sein, da die Repräsentantenmenge schon bei kleineren Problemen relativ viele Elemente enthält. In gewissen gutartigen Räumen dagegen, z. B. in flach und strikt konvexen Banachräumen (hierzu gehören die Räume  $l_p$ ,  $L_p$  ( $1 < p < \infty$ )), konvergieren die Verfahren der direkten zyklischen Minimierung (/2/, /3/, /4/, /5/, /6/). Um den Gedanken der zyklischen Einzelschrittiteration aufrechtzuerhalten und der Festlaufproblematik aus dem Wege zu gehen, wird in dieser Arbeit für die  $l_\infty$ ,  $l_1$ -Approximation ein anderes Vorgehen gewählt gegenüber /1/. Die Approximationsaufgaben werden in lineare Optimierungsaufgaben übergeführt und diese mittels eines in /7/ dargestellten Einzelschrittverfahrens gelöst. Dabei mag die recht einfache Struktur des Verfahrens (eingeschränktes Gauß-Seidel-Verfahren) eine Entschädigung sein für die eventuell langsame Konvergenz des Verfahrens. Die vorliegende Arbeit soll von der Möglichkeit der Darstellung eines relativ einfach strukturierten Einzelschrittverfahrens zur Lösung linearer  $l_\infty$  und  $l_1$ -Approximationsaufgaben berichten. Ergebnisse über numerische Tests müssen an anderer Stelle vermittelt werden.

## 1. Die diskrete lineare Tschebyscheff-Approximation

Mit  $\underline{b} = (b_1, \dots, b_m)^T$ ,  $A = (a_{ij})$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ ,  
 $\underline{x} = (x_1, \dots, x_n)^T$ ;  $b_i$ ,  $x_i$ ,  $a_{ij}$  sämtlich reell, betrachten wir  
die Aufgabe

$$E_\infty := \min_{\underline{x}} \|\underline{b} - A \underline{x}\|_\infty \quad (1)$$

oder ausführlich

$$\min_{x_1, \dots, x_n} \max_{1 \leq r \leq m} \left| b_r - \sum_{k=1}^n x_k a_{rk} \right|. \quad (1')$$

Die lineare Optimierungsaufgabe

$$\begin{aligned} -\underline{e} M &\leq \underline{b} - A \underline{x} \leq \underline{e} M \\ M &\longrightarrow \min \end{aligned} \quad (2)$$

mit  $\underline{e} = (1, 1, \dots, 1)^T \in \mathbb{R}^m$ , ist äquivalent mit (1). Der Optimalwert der Zielfunktion ist  $M = E_\infty$ .

Die zu (2) duale lineare Optimierungsaufgabe ist

$$\begin{aligned} \underline{e}^T \underline{u}^1 + \underline{e}^T \underline{u}^2 &= 1 \\ A^T \underline{u}^1 - A^T \underline{u}^2 &= \underline{0} \\ \underline{u}^1, \underline{u}^2 &\geq \underline{0} \\ \underline{b}^T \underline{u}^1 - \underline{b}^T \underline{u}^2 &\longrightarrow \max \end{aligned} \quad (3)$$

oder mit  $\underline{v} = \underline{u}^1 - \underline{u}^2$

$$\begin{aligned}
\underline{e}^T \underline{u}^1 + \underline{e}^T \underline{u}^2 &= 1 \\
\Lambda^T \underline{v} &= \underline{0} \\
\underline{u}^1, \underline{u}^2 &\geq \underline{0} \\
\underline{b}^T \underline{v} &\rightarrow \max .
\end{aligned}
\tag{3'}$$

Nun ist aus der Theorie der linearen Optimierung bekannt, daß zulässige Lösungen von (1) und (3') genau dann optimal sind, wenn gilt

$$\underline{b}^T \underline{v} - M = 0 .
\tag{4}$$

Jede beste Approximation  $M, x_1, \dots, x_n$  von (1) genügt also dem linearen Gleichungssystem

$$\begin{aligned}
\underline{e}M + \Lambda \underline{x} - \underline{y}^1 &= \underline{b} \\
2\underline{e}M - \underline{y}^1 - \underline{y}^2 &= \underline{0} \\
M - \underline{b}^T \underline{v} &= 0 \\
\Lambda^T \underline{v} &= \underline{0} \\
\underline{v} - \underline{u}^1 + \underline{u}^2 &= \underline{0} \\
\underline{e}^T \underline{u}^1 + \underline{e}^T \underline{u}^2 &= 1
\end{aligned}
\tag{5}$$

mit den Nebenbedingungen

$$\underline{y}^1, \underline{y}^2, \underline{u}^1, \underline{u}^2 \geq \underline{0} .
\tag{5'}$$

Und umgekehrt kann man aus jeder Lösung von (5) mit (5') die Koeffizienten  $x_1, \dots, x_n$  einer besten Approximation von (1) ablesen. Der Abkürzung wegen führen wir an dieser Stelle folgende Bezeichnungen ein:

$$\underline{z} := (\underline{m}, \underline{x}^T, \underline{y}^{1T}, \underline{y}^{2T}, \underline{y}^T, \underline{u}^{1T}, \underline{u}^{2T})^T$$

$$\underline{d} := (\underline{b}^T, \underline{q}^T, 0, \underline{q}^T, \underline{q}^T, 1)^T, \underline{q} \in \mathbb{R}^m,$$

$$C := \begin{pmatrix} \underline{e}^T A & -I & 0 & 0 & 0 & 0 \\ 2\underline{e}^T 0 & -I & -I & 0 & 0 & 0 \\ 1 \underline{q}^T & \underline{q}^T & \underline{q}^T & -\underline{b}^T & \underline{q}^T & \underline{q}^T \\ \underline{q}^T 0 & 0 & 0 & A^T & 0 & 0 \\ \underline{q}^T 0 & 0 & 0 & I & -I & I \\ 0 \underline{q}^T & \underline{q}^T & \underline{q}^T & \underline{q}^T & \underline{e}^T & \underline{e}^T \end{pmatrix},$$

wobei  $I$  die  $m \times m$  Einheitsmatrix bezeichnet. Mit diesen Bezeichnungen geht (5) über in

$$C \underline{z} = \underline{d}. \quad (6)$$

Die Lösung von (6) unter den Nebenbedingungen (5') ist äquivalent der Lösung von

$$F(\underline{z}) := (C\underline{z} - \underline{d})^T (C\underline{z} - \underline{d}) \rightarrow \min \quad (7)$$

unter den Nebenbedingungen (5').  $F(\underline{z})$  ist ein konvexes und differenzierbares Funktional mit  $F(\underline{z}) \geq 0$  für alle  $\underline{z}$  und  $F(\underline{z}) = 0$  genau dann, wenn (6) gilt. Da (1) eine Lösung besitzt, besitzt auch, wie aus der Theorie der linearen Optimierung bekannt, (3') eine Lösung und es gilt (4). Folglich gibt es ein  $\underline{z}$  mit  $F(\underline{z}) = 0$  und (5').

Das in //7// beschriebene Einzelschrittverfahren läßt sich auf die Optimierungsaufgabe (7), (5') anwenden:

$$\min (F(\underline{z}), \underline{z} \in Z), \quad (7')$$

wobei  $Z$  das folgende konvexe kartesische Produkt bezeichnet

$$Z := \mathbb{R}^1 \times \mathbb{R}^n \times \mathbb{R}_+^{2m} \times \mathbb{R}^m \times \mathbb{R}_+^{2m}$$

mit

$$R_+^B := (\underline{w} \in R^B : \underline{w} \geq \underline{0}),$$

sofern noch die Kompaktheit von  $S^0 := (\underline{z} \in Z : F(\underline{z}) \leq F(\underline{z}^0))$  nachgewiesen werden kann.

Wenn  $F(\underline{z})$  konvex, stetig und die Menge der Minimallösungen von (1) beschränkt ist (und damit auch von (7'), da  $\underline{0} \leq \underline{u}^1 \leq \underline{e}$ ,  $i = 1, 2$ ), muß auch  $S^0$  notwendig beschränkt sein. Dies ergibt sich leicht aus den nachstehenden Überlegungen.

$L$  bezeichne die Menge der Minimallösungen von (7'). Dann existiert eine Konstante  $K$  derart, daß  $\|\underline{z}^1 - \underline{z}^2\| \leq K$  ist für alle  $\underline{z}^1, \underline{z}^2 \in L$ . Wir wählen nun eine Zahl  $K_0 > 0$  und nehmen an, daß  $S^0$  unbeschränkt ist. Dann existieren  $\underline{w}^r \in S^0$ ,  $\underline{z} \in L$  mit  $\|\underline{z} - \underline{w}^r\| \rightarrow \infty$  für  $r \rightarrow \infty$ . Es sei  $\underline{z}^r := g_r \underline{z} + (1-g_r) \underline{w}^r$ ,  $0 \leq g_r \leq 1$ , und

$$K + K_0 = \|\underline{z}^r - \underline{z}\| = (1 - g_r) \|\underline{w}^r - \underline{z}\| \quad (8)$$

für alle  $r \geq r_0$ , wobei  $r_0$  definiert ist durch

$$K + K_0 < \|\underline{w}^r - \underline{z}\|$$

für alle  $r \geq r_0$ . Wegen der Beschränktheit von  $(\underline{z}^r)$  existiert ein Häufungspunkt  $\underline{z}^{\infty}$  mit  $\|\underline{z}^{\infty} - \underline{z}\| = K + K_0$ . Mit der Konvexität von  $F$  wird

$$F(\underline{z}^r) \leq g_r F(\underline{z}) + (1-g_r) F(\underline{w}^r) \leq g_r F(\underline{z}) + (1-g_r) F(\underline{z}^0).$$

Folglich ist  $F(\underline{z}^{\infty}) = F(\underline{z})$ , da  $F$  stetig und  $g_r \rightarrow 1$  für  $r \rightarrow \infty$ , wie man aus (8) sieht. Damit ist  $\underline{z}^{\infty}$  aus  $L$  und wegen

$$\|\underline{z}^{\infty} - \underline{z}\| = K + K_0$$

haben wir einen Widerspruch zur Bedeutung von  $K$  und  $K_0$ . Die Menge  $S^0$  ist also kompakt, wenn die Menge der Minimallösungen von (1) beschränkt ist. Dies wiederum ist genau dann der Fall, wenn die Spaltenvektoren von  $A$  linear unabhängig sind.

Um das Einzelschrittverfahren in seiner Allgemeinheit übersichtlich darzustellen, werden mengenwertige Abbildungen eingeführt.

Jedem  $\underline{z} = (\underline{z}_1^T, \dots, \underline{z}_r^T)^T$  aus  $Z$ , wobei  $1 \leq r \leq 5m + n + 1$ ,

$\underline{z}_i$  aus  $Z_i$  so, daß  $Z = Z_1 \times Z_2 \times \dots \times Z_r$ , wird ein  $N_1(\underline{z}) \subset Z_1$  ( $i = 1, 2, \dots, r$ ) zugeordnet. Dabei sollen die  $N_1(\underline{z})$  abgeschlossen im Sinne von a) und monoton im Sinne von b) sein:

a) Aus  $\underline{z}^S \in Z$ ,  $\underline{z}^S \rightarrow \underline{z}^{OO} \in Z$ ,  $\underline{w}_1^S \in N_1(\underline{z}^S)$ ,  $\underline{w}_1^S \rightarrow \underline{w}_1^{OO}$  folgt

$$\underline{w}_1^{OO} \in N_1(\underline{z}^{OO}); \quad i = 1, 2, \dots, r.$$

b)  $F(\underline{z}_{1-}, \underline{w}_1, \underline{z}_{1+}) = F(\underline{z})$  für alle  $\underline{w}_1 \in N_1(\underline{z})$  (wobei

$$F(\underline{z}_{1-}, \underline{w}_1, \underline{z}_{1+}) = F(\underline{z}_1, \dots, \underline{z}_{i-1}, \underline{w}_1, \underline{z}_{i+1}, \dots, \underline{z}_r))$$

genau dann, wenn  $\underline{z}$   $i$ -minimal ist (d. h.  $F(\underline{z}) \leq F(\underline{z}_{1-}, \underline{w}_1, \underline{z}_{1+})$ )

für alle  $\underline{w}_1 \in Z_1$ ).  $F(\underline{z}_{1-}, \underline{w}_1, \underline{z}_{1+}) < F(\underline{z})$  für alle  $\underline{w}_1 \in N_1(\underline{z})$

genau dann, wenn  $\underline{z}$  nicht  $i$ -minimal ist.

Nach diesen Vorbereitungen kann das Verfahren in übersichtlicher Weise dargestellt werden:

$$\underline{z}_1^{k+1} \in N_1(\underline{z}_{1-}^{k+1}, \underline{z}_1^k, \underline{z}_{1+}^k), \quad (9)$$

$i = 1, 2, \dots, r$ . Es werden also für alle Komponenten die zuletzt bekannten Werte zugrunde gelegt. Nach Beendigung eines Zyklus, bestehend aus  $r$  Einzelschritten, liegt der neue Iterationspunkt  $\underline{z}^{k+1}$  vor. In /7/ wurde gezeigt, daß jeder Häufungspunkt  $\underline{z}^{OO}$  von  $(\underline{z}^k)$  eine optimale Lösung von (7') darstellt. Im folgenden betrachten wir eine spezielle Wahl von  $N_1(\underline{z})$ , welche sich aus der zyklischen Minimierung (/7/, /8/) ergibt. Wir setzen

$$\begin{aligned} N_1(\underline{z}) &:= (\underline{w}_1 \in Z_1 : \min (F(\underline{z}_{1-}, \underline{z}_1, \underline{z}_{1+}), \underline{z}_1 \in Z_1) = \\ &= F(\underline{z}_{1-}, \underline{w}_1, \underline{z}_{1+})), \end{aligned} \quad (10)$$

d. h.  $\underline{w}_1$  minimiert  $F(\underline{z}_{1-}, \underline{z}_1, \underline{z}_{1+})$  über  $Z_1$ .

Die Eigenschaften a), b) sind erfüllt:

a)  $\underline{w}_1^s \in N_1(\underline{z}^s)$ . Dann gilt  $F(\underline{z}_{1-}^s, \underline{w}_1^s, \underline{z}_{1+}^s) \leq F(\underline{z}_{1-}^s, \underline{z}_1, \underline{z}_{1+}^s)$   
für alle  $\underline{z}_1$  aus  $Z_1$  und wegen der Stetigkeit von  $F$  wird  
 $F(\underline{z}_{1-}, \underline{w}_1, \underline{z}_{1+}) \leq F(\underline{z}_{1-}, \underline{z}_1, \underline{z}_{1+})$  für alle  $\underline{z}_1$  aus  $Z_1$ , d. h.  
 $\underline{w}_1^{\infty}$  aus  $N_1(\underline{z}^{\infty})$ .

b) Offenbar gilt  $F(\underline{z}_{1-}, \underline{w}_1, \underline{z}_{1+}) \leq F(\underline{z})$  für alle  $\underline{w}_1$  aus  $N_1(\underline{z})$   
und das Gleichheitszeichen genau dann, wenn  $\underline{z}$  i-minimal ist.

Setzen wir  $r = 5m + n + 1$ , d. h.  $z_1 = M$ ,  $z_2 = x_1, \dots, z_{n+1} = x_n$ ,  
 $z_{n+2} = y_1, \dots, z_{5m+n+1} = u_m^2$ , dann erhalten wir mit (9) und  
(10), wegen  $\text{grad } F(\underline{z}) = 2 C^T C \underline{z} - 2 C^T \underline{d}$ , die folgende einfache  
Iterationsvorschrift zur Lösung von (1) bzw. (7):

$$(C^T C)_{ii} z_i^{k+1} := (C^T \underline{d})_i - \sum_{j=1}^{i-1} (C^T C)_{ij} z_j^{k+1} - \sum_{j=i+1}^{5m+n+1} (C^T C)_{ij} z_j^k$$

$$z_i^{k+1} := \tilde{z}_i^{k+1} \text{ für } i = 1, \dots, n+1, 2m+n+2, \dots, 3m+n+1 \quad (11)$$

$$z_i^{k+1} := \max(\tilde{z}_i^{k+1}, 0) \text{ für } i = n+2, \dots, 2m+n+1, \dots, 5m+n+1.$$

Dabei bedeutet  $(B)_{ij}$  das Element der Matrix  $B$ , welches in der  
i-ten Zeile und in der j-ten Spalte steht. Entsprechend bedeutet  
 $(\underline{v})_i$  die i-te Komponente des Vektors  $\underline{v}$ .

Berücksichtigen wir die konkrete Darstellung von  $C^T C$ ,  $C^T \underline{d}$  und  
 $\underline{z}$ , dann erhalten wir die folgenden Iterationsformeln:

$$(5m+1)M^{k+1} := -\underline{e}^T A \underline{x}^k + 3\underline{e}^T \underline{y}^{1,k} + 2\underline{e}^T \underline{y}^{2,k} + \underline{b}^T \underline{v}^k + \underline{e}^T \underline{b}, \quad (12)$$

$$\begin{aligned} (A^T A)_{ii} x_i^{k+1} &:= (A^T \underline{b})_i - (A^T \underline{e})_i M^{k+1} - \sum_{j=1}^{i-1} (A^T A)_{ij} x_j^{k+1} \\ &\quad - \sum_{j=i+1}^n (A^T A)_{ij} x_j^k + \sum_{j=1}^m (A^T)_{ij} y_j^{1,k} \\ &\quad (i=1, 2, \dots, n), \end{aligned}$$

$$2 \tilde{y}_i^{1,k+1} := -b_i + 3 M^{k+1} + \sum_{j=1}^n (A)_{ij} x_j^{k+1} - y_i^{2,k}$$

$$y_i^{1,k+1} = \max(\tilde{y}_i^{1,k+1}, 0) \text{ für } i = 1, 2, \dots, m,$$

$$\tilde{y}_i^{2,k+1} := 2 M^{k+1} - y_i^{1,k+1}$$

$$y_i^{2,k+1} = \max(\tilde{y}_i^{2,k+1}, 0) \text{ für } i = 1, 2, \dots, m,$$

$$\begin{aligned} (\underline{b}\underline{b}^T + AA^T + I)_{ii} v_i^{k+1} &:= b_i M^{k+1} - \sum_{j=1}^{i-1} (\underline{b}\underline{b}^T + AA^T + I)_{ij} v_j^{k+1} \\ &\quad - \sum_{j=i+1}^m (\underline{b}\underline{b}^T + AA^T + I)_{ij} v_j^k + u_i^{1,k} - u_i^{2,k} \\ &\quad (i = 1, 2, \dots, m), \end{aligned}$$

$$2 \tilde{u}_i^{1,k+1} := 1 + v_i^{k+1} - \sum_{j=1}^{i-1} u_j^{1,k+1} - \sum_{j=i+1}^m u_j^{1,k} - \sum_{j=1}^m u_j^{2,k}$$

$$u_i^{1,k+1} = \max(\tilde{u}_i^{1,k+1}, 0) \text{ für } i = 1, 2, \dots, m,$$

$$2 \tilde{u}_i^{2,k+1} := 1 - v_i^{k+1} - \sum_{\substack{j=1 \\ j \neq i}}^m u_j^{1,k+1} - \sum_{j=1}^{i-1} u_j^{2,k+1} - \sum_{j=i+1}^m u_j^{2,k}$$

$$u_i^{2,k+1} = \max(\tilde{u}_i^{2,k+1}, 0) \text{ für } i = 1, 2, \dots, m.$$

## 2. Die $l_1$ -Approximation

Ein entsprechendes Vorgehen für die  $l_1$ -Approximation

$$E_1 := \min_{\underline{x}} \| \underline{b} - A \underline{x} \|_1, \quad (13)$$

oder ausführlich

$$\min_{x_1, \dots, x_n} \sum_{r=1}^m \left| b_r - \sum_{j=1}^n x_j a_{rj} \right|,$$

liefert uns zu (12) analoge Iterationsformeln zur Bestimmung einer besten  $l_1$ -Approximation. Der Übergang von (13) zu einer äquivalenten linearen Optimierungsaufgabe ergibt

$$\begin{aligned} A \underline{x} + \underline{y} &\geq \underline{b} \\ -A \underline{x} + \underline{y} &\geq -\underline{b} \\ \underline{e}^T \underline{y} &\rightarrow \min \end{aligned} \quad (14)$$

Die zu (14) duale lineare Optimierungsaufgabe ist

$$\begin{aligned} A^T \underline{y} &= \underline{0} \\ \underline{y} - \underline{u}^1 + \underline{u}^2 &= \underline{0} \\ \underline{u}^1 + \underline{u}^2 &= \underline{e} \\ \underline{u}^1, \underline{u}^2 &\geq \underline{0} \\ \underline{b}^T \underline{y} &\rightarrow \max \end{aligned} \quad (15)$$

Aus (14) und (15) erhält man hier das zu (5) analoge lineare Gleichungssystem

$$C \underline{z} = \underline{d}, \quad (16)$$

mit den Bezeichnungen

$$\underline{z} := (\underline{y}^T, \underline{x}^T, \underline{y}^{1T}, \underline{y}^{2T}, \underline{y}^T, \underline{u}^{1T}, \underline{u}^{1T}, \underline{u}^{2T})^T,$$

$$\underline{d} := (\underline{b}^T, \underline{0}^T, \underline{0}, \underline{0}^T, \underline{0}^T, \underline{e}^T)^T,$$

$$C := \begin{pmatrix} I & A & -I & 0 & 0 & 0 & 0 \\ 2I & 0 & -I & -I & 0 & 0 & 0 \\ \underline{g}^T & \underline{0}^T & \underline{0}^T & \underline{0}^T & -\underline{b}^T & \underline{0}^T & \underline{0}^T \\ 0 & 0 & 0 & 0 & A^T & 0 & 0 \\ 0 & 0 & 0 & 0 & I & -I & I \\ 0 & 0 & 0 & 0 & 0 & I & I \end{pmatrix}$$

und den Nebenbedingungen

$$\underline{y}^1, \underline{y}^2, \underline{u}^1, \underline{u}^2 \geq \underline{0}. \quad (17)$$

Die Lösung der zu (7') analogen Aufgabe mit Hilfe der zyklischen Minimierung führt auf eine zu (11) analoge Iterationsvorschrift oder ausführlich auf die zu (12) analogen Beziehungen:

$$6 y_i^{k+1} := b_i - \sum_{j=1}^{i-1} y_j^{k+1} - \sum_{j=i+1}^m y_j^k - \sum_{j=1}^n (A)_{ij} x_j^k \quad (18)$$

$$+ 3 y_1^{1,k} + 2 y_1^{2,k} + \sum_{j=1}^m b_j v_j^k \quad (i = 1, 2, \dots, m),$$

$$(\Delta^T A)_{1i} x_1^{k+1} := (\Delta^T \underline{b})_i - \sum_{j=1}^m (\Delta^T)_{ij} y_j^{k+1} - \sum_{j=1}^{i-1} (\Delta^T A)_{ij} x_j^{k+1}$$

$$- \sum_{j=i+1}^n (\Delta^T A)_{ij} x_j^k + \sum_{j=1}^m (\Delta^T)_{ij} y_j^{1,k}$$

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

$$2 \tilde{y}_1^{1,k+1} := -b_1 + 3 y_1^{k+1} + \sum_{j=1}^n (A)_{1j} x_j^{k+1} - y_1^{2,k}$$

$$\tilde{y}_1^{1,k+1} := \max(\tilde{y}_1^{1,k+1}, 0)$$

$$y_1^{2,k+1} := 2 y_1^{k+1} - y_1^{1,k+1}$$

$$y_1^{2,k+1} := \max(y_1^{2,k+1}, 0) \quad (i = 1, 2, \dots, m),$$

$$\begin{aligned} (\underline{b} \underline{b}^T + A A^T + I)_{ii} v_1^{k+1} &:= b_i \sum_{j=1}^m y_j^{k+1} - \sum_{j=1}^{i-1} (\underline{b} \underline{b}^T + A A^T + I)_{ij} v_j^{k+1} \\ &- \sum_{j=i+1}^m (\underline{b} \underline{b}^T + A A^T + I)_{ij} v_j^k + u_1^{1,k} - u_1^{2,k} \\ &(i = 1, 2, \dots, m), \end{aligned}$$

$$2 \tilde{u}_1^{1,k+1} := 1 + v_1^{k+1}, \quad u_1^{1,k+1} := \max(\tilde{u}_1^{1,k+1}, 0),$$

$$2 \tilde{u}_1^{2,k+1} := 1 - v_1^{k+1}, \quad u_1^{2,k+1} := \max(\tilde{u}_1^{2,k+1}, 0),$$

$$(i = 1, 2, \dots, m).$$

Es soll noch erwähnt werden, daß die Folge  $(\underline{x}^k)$  linear konvergiert, falls die Aufgabe (7') eine eindeutig bestimmte Lösung hat. Einen entsprechenden Konvergenzbeweis findet man in [7]. Der Anfangsvektor  $\underline{x}^0$  kann beliebig gewählt werden.

Zusammenfassend haben wir den folgenden

**Satz:** Die Spaltenvektoren der Matrix A seien linear unabhängig. Dann gelten die folgenden Aussagen:

1. Die Komponenten jedes Häufungspunktes  $\underline{x}^\infty$  der nach (12) bestimmten Folge  $(\underline{x}^k)$  stellen die Koeffizienten einer besten Approximation von (1) dar. Die Folge  $(M^k)$  besitzt den einzigen Häufungspunkt  $M^\infty = E_{00}$ .
2. Die Komponenten jedes Häufungspunktes  $\underline{x}^\infty$  der nach (18) bestimmten Folge  $(\underline{x}^k)$  stellen die Koeffizienten

einer besten Approximation von (13) dar. Die Folge  $(\underline{e}^T \underline{y}^k)$  besitzt den einzigen Häufungspunkt  $\underline{e}^T \underline{y}^\infty = E_1$ .

3. Die Iterationsvorschriften (12) bzw. (18) erzeugen linear konvergente Folgen, falls (5) mit (5') bzw. (16) mit (17) eindeutige Lösungen besitzen.

### Literatur

- /1/ Albrand, H.-J., Verfahren zur Lösung der diskreten Tschebyscheff-Approximation, Wiss. Z. d. Universität Rostock, Math.-Nat. Reihe, Heft 8, 639 - 643, (1974)
- /2/ Albrand, H.-J., Über die Lösung linearer Gleichungssysteme durch Spaltenapproximation, Wiss. Z. d. Universität Rostock, Math.-Nat. Reihe, Heft 8, 755 - 757, (1972)
- /3/ Albrand, H.-J., Spaltenapproximation und Fixpunkte, Wiss. Z. d. Universität Rostock, Math.-Nat. Reihe, Heft 10, 1253 - 1256, (1975)
- /4/ Kiesewetter, H.; Maeß, G., Elementare Methoden der numerischen Mathematik, S. 85 - 89, Berlin, 1974
- /5/ Stiles, W. J., A solution to Hirschfeld's problem, Nieuw Archief voor Wiskunde (3), 116 - 119, XIII (1965)
- /6/ Pantelidis, G., Konvergente Iterationsverfahren für flach konvexe Banachräume, Schriften des Rheinisch-Westfälischen Instituts für instrumentelle Mathematik an der Universität Bonn Serie A, Nr. 23, (1968)

- /7/ Oettli, W., Einzelschrittverfahren zur Lösung konvexer und dual-konvexer Minimierungsprobleme, ZAMM 54, 343 - 351, (1974)
- /8/ Warga, J., Minimizing certain convex functions, SIAM J. Appl. Math., 11, 588 - 593, (1963)

eingegangen am: 15. 4. 1977

Anschrift des Verfassers

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



Über die Anzahl elementarer BUB in eingeschränkten  $(v,k)$ -Familien

---

1. Einleitung

Zur optimalen Behandlung von Problemstellungen der mathematischen Statistik werden ausgehend von einem Optimalitätskriterium die Entscheidungsfunktion (das statistische Auswertungsverfahren) und der statistische Versuchsplan einschließlich des Versuchsumfanges gewählt. Um die statistischen Modelle mit ihren Zufallsfehlern auf praktische Probleme anwenden zu können, ist bei der Versuchsdurchführung weitgehend zu randomisieren. Ein wichtiger Bestandteil des Versuchsplanes ist die Versuchsanlage, die sowohl der Ausschaltung von Störfaktoren als auch der Zuordnung der Versuchsobjekte zu den Stufenkombinationen der Prüffaktoren (Behandlungen) dient. Versuchsanlagen wurden allgemein und unabhängig vom Modell der statistischen Auswertung von Herrendörfer und Rasch (1978) definiert. Unter den Versuchsanlagen spielen solche zur Ausschaltung einer Störgröße eine besondere Rolle, man nennt sie Blockanlagen. Ein wichtiger Spezialfall sind die balancierten unvollständigen Blockanlagen BUB, die in der mathematischen Literatur bereits bei Kirkman (1847, 1850) und Steiner (1853) auftreten und seit den Arbeiten von Bose (1939) bzw. Fisher (1940) in mehreren hundert Artikeln behandelt wurden. [Federer und Balaam (1973)]. Wir geben die

Definition 1: Gegeben sei eine Folge  $\mathcal{A}$  von  $v \geq 3$  (verschiedenen) Elementen  $1, 2, \dots, v$ . Eine Menge  $\mathcal{B}$  von  $b$  Teilmengen (genannt Blocks) des Umfangs  $2 \leq k < v$  aus  $\mathcal{A}$  heißt BUB  $B[v,k,\lambda]$ , falls jedes Paar von Elementen aus  $\mathcal{A}$  in genau  $\lambda > 0$  Blocks auftritt. Zwei BUB  $B(v,k,\lambda)$  mit gleichen Parametern heißen gleich, wenn sie durch Umordnung der Blocks bzw. durch Permutation der Elemente  $1, \dots, v$  ineinander überführt werden können.

Aus der Definition folgt, daß

$$r = \frac{bk}{v} \quad (1)$$

eine positive ganze Zahl sein muß,  $r$  ist die Anzahl des Auftretens der einzelnen Elemente von  $\mathcal{E}$ . Neben (1) muß

$$\lambda = \frac{r(k-1)}{v-1} \quad (2)$$

gelten.

Unter Randomisation verstehen wir einerseits die (völlig unproblematische) zufällige Zuordnung von  $v$  Behandlungen zu den Zahlen  $1, \dots, v$  und andererseits die zufällige Auswahl einer BUB mit festem  $v, k, \lambda$  aus der Menge aller  $B[v, k, \lambda]$ . Hierzu ist es erforderlich, diese Menge angeben zu können. In vorliegender Arbeit sollen erste Ergebnisse in dieser Richtung mitgeteilt werden.

## 2. Vollständige und eingeschränkte $(v, k)$ -Familien von BUB

Wir gehen davon aus, daß in praktischen Fragestellungen die Größen  $v$  und  $k$  vorgegeben sind. Durch die Planung des Stichprobenumfanges wird dann  $b$  festgelegt und damit nach (1) und (2) auch  $\lambda$ . Anschließend geht es darum, aus allen  $B[v, k, \lambda]$  zufällig eine auszuwählen.

Wir geben die

**Definition 2:** Eine BUB  $B[v, k, \lambda]$  heißt zusammengesetzt, falls sie mindestens eine Teilmenge von Blocks enthält, die mit  $\lambda' < \lambda$  eine BUB  $B(v, k, \lambda')$  ist, andernfalls heißt sie elementar.

**Definition 3:** Gegeben seien positive ganze Zahlen  $v$  und  $k$  mit  $v > k \geq 2$ . Unter einer (vollständigen)  $(v, k)$ -Familie  $F[v, k]$  von BUB verstehen wir die Menge aller BUB  $B[v, k, \lambda]$  mit den angegebenen Parametern  $v$  und  $k$  und beliebigem  $\lambda$ . Unter einer eingeschränkten  $(v, k)$ -Familie  $F^*[v, k]$  verstehen wir die Teilmenge einer  $(v, k)$ -Familie, deren Elemente solche elementaren BUB sind bzw. aus solchen elementaren BUB zusammengesetzt sind, die keinen Block mehrfach enthalten. Mit  $f^*(v, k)$  bzw.  $f(v, k)$  werden die Anzahlen der in  $F^*(v, k)$  bzw.  $F(v, k)$  enthaltenen verschiedenen elementaren BUB bezeichnet. Aus Definition 2

folgt, daß, falls  $B_1, \dots, B_{f(v,k)}$  alle  $f(v,k)$  elementaren BUB sind, jedes Element  $B \in F(v,k)$  als "Linearkombination"

$$B = \sum_{i=1}^{f(v,k)} c_i B_i \quad (c_i \geq 0, \text{ ganz}) \quad (3)$$

der  $B_i$  dargestellt werden kann. Als "Linearkombination" von BUB  $B_i$  bezeichnen wir die BUB  $B$  die man erhält, indem man die Blocks von  $B_i$  in  $c_i$  (nicht notwendig verschiedenen) Permutationen hinschreibt, entsprechend ist die Summe in (3) zu verstehen.

Es ist sofort zu sehen, daß folgender Satz gilt.

Satz 1: In  $F(v,k)$  bzw.  $F^*(v,k)$  gelten die Beziehungen

$$f(v,2) = f^*(v,2) = 1$$

und  $f(v,k) = f(v,v-k)$

bzw.  $f^*(v,k) = f^*(v,v-k)$ .

Wir wollen im folgenden für einige  $v$  und  $k$  die Werte von  $f^*(v,k)$  bestimmen und beschränken uns wegen Satz 1 auf  $3 \leq k \leq \frac{v}{2}$ . Es gilt der

Satz 2: Es gibt  $(v,k)$ -Familien, für die in

$$f(v,k) \geq f^*(v,k) \quad (4)$$

das Größerzeichen gilt.

Für den Beweis genügt ein Beispiel. In der  $(10,3)$ -Familie ist

147	}	2 mal	123
158			12X
169			13X
248			23X
259			456
267			45X
349			46X
357			56X
368			789
			78X
	79X		
	89X		

eine elementare BUB, die nicht in  $F^*(10,3)$  liegt.

### 3. Bestimmung von $f^*(6,3)$

Satz 3: Es gilt  $f^*(6,3) = 1$ .

Beweis: Die  $\binom{6}{3} = 20$  Kombinationen von 6 Elementen zur dritten Klasse sind:

123	135	234	256	
124	136	235	345	
125	145	236	346	(5)
126	146	245	356	
134	156	246	456	

Das kleinste  $\lambda$ , das (2) erfüllt ist  $\lambda = 2$  und ergibt ein minimales  $r$  von 5. Die kleinste denkbare BUB in  $F^*(6,3)$  hätte folglich (falls sie existiert) die Parameter  $v=6$ ,  $k=3$ ,  $\lambda=2$ ,  $b=10$  und  $r=5$ . Wir zeigen, daß es genau eine solche BUB gibt. Wegen  $\lambda=2$  müssen zunächst aus den vier ersten Blocks in (5) zwei (mit 12) ausgewählt werden. Dazu gibt es sechs Möglichkeiten (Fall 1 bis 6). Aus den sechs restlichen Blocks mit einer 1 sind wegen  $r=5$  noch drei passend auszuwählen, und das ergibt jeweils zwei weitere Möglichkeiten, die die BUB aber auch eindeutig festlegen (Fall  $x,1$  bzw.  $x,2$  mit  $x = 1, \dots, 6$ ). Für diesen einfachen Fall wollen wir alle Möglichkeiten einmal aufschreiben.

Fall 1,1    Fall 1,2    Fall 2,1    Fall 2,2    Fall 3,1    Fall 3,2

123	123	123	123	123	123
124	124	125	125	126	126
136	135	134	136	134	135
145	146	146	145	145	145
156	156	156	146	156	146

Fall 4,1    Fall 4,2    Fall 5,1    Fall 5,2    Fall 6,1    Fall 6,2

124	124	124	124	125	125
125	125	126	126	126	126
134	135	135	134	134	134
136	136	136	135	135	136
156	146	145	156	146	145

Wir haben nur noch zu zeigen, daß diese 12 BUB  $B[6,3,2]$  gleich sind. Hierzu geben wir Permutationen an, die Fall 1,1 in die 11 anderen Fälle überführen.

$$\begin{pmatrix} 5 & 6 \\ 6 & 5 \end{pmatrix}, \begin{pmatrix} 4 & 5 & 6 \\ 5 & 6 & 4 \end{pmatrix}, \begin{pmatrix} 4 & 5 \\ 5 & 4 \end{pmatrix}, \begin{pmatrix} 4 & 6 \\ 6 & 4 \end{pmatrix}, \begin{pmatrix} 4 & 6 & 5 \\ 6 & 5 & 4 \end{pmatrix}, \begin{pmatrix} 3 & 5 \\ 5 & 3 \end{pmatrix},$$

$$\begin{pmatrix} 3 & 5 & 6 \\ 5 & 6 & 3 \end{pmatrix}, \begin{pmatrix} 3 & 6 \\ 6 & 3 \end{pmatrix}, \begin{pmatrix} 3 & 5 & 6 \\ 6 & 3 & 5 \end{pmatrix}, \begin{pmatrix} 3 & 4 & 5 & 6 \\ 5 & 6 & 4 & 3 \end{pmatrix}, \begin{pmatrix} 3 & 4 & 5 & 6 \\ 5 & 6 & 3 & 4 \end{pmatrix}.$$

Man kann nun leicht zeigen, daß es in  $F(6,3)$  keine weitere elementare BUB  $B[6,3,2]$  gibt, es gilt damit

**Satz 4:** In  $F(6,3)$  gibt es nur eine elementare BUB mit  $\lambda = 2$ .

**Beweis:** Gäbe es in  $F(6,3)$  eine elementare BUB  $B[6,3,2]$  außer der in Satz 3 angegebenen, so müßte sie aus  $F(6,3) - F^*(6,3)$  stammen und folglich wegen  $\lambda=2$  einige Blocks doppelt enthalten. O.B.d.A. sei  $12\bar{3}$  ein doppelt auftretender Block. Dann darf wegen  $r=5$  höchstens noch ein Block mit einer 1 doppelt auftreten, der aber weder eine 2 noch 3 enthalten darf. Es sei dieser Block  $145$  ( $146, 156$ ), aber dann müßte der fünfte Block mit der 1 zwei Sechsen (Fünfen, Vieren) enthalten und das ist nicht gestattet. Analog ergeben sich Widersprüche, wenn die doppelt auftretenden Blocks ein anderes Element als die Eins gemeinsam haben. Haben doppelt auftretende Blocks kein gemeinsames Element, so kommen nur  $12\bar{3}$  und  $456$  in Frage. Dann kann es aber wegen  $\lambda=2$  keinen weiteren Block geben. Blicke der Fall, daß genau ein Block doppelt auftritt, z. B. die  $12\bar{3}$ , dann müßten aber in den acht restlichen Blocks noch je drei Einsen, Zweien und Dreien getrennt untergebracht werden und das geht nicht. Es ist nach Satz 4 zu vermuten, daß  $f(6,3) = 1$  gilt.

#### 4. Bestimmung von $f^*(7,3)$

**Satz 5:** Es gilt  $f^*(7,3) = 2$ .

**Beweis:** Die  $\binom{7}{3} = 35$  Kombinationen von sieben Elementen zur dritten Klasse lassen sich in drei Teilmengen mit 7, 7 und 21 Blocks zerlegen, die elementare BUB sind

$$(\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 3).$$

Teilmenge 1	Teilmenge 2	Teilmenge 3		
$\lambda = 1$	$\lambda = 1$	$\lambda = 3$		
124	126	123	147	257
137	134	125	167	345
156	157	127	234	347
235	237	135	236	357
267	245	136	246	367
346	356	145	247	456
457	467	146	256	567

Die beiden ersten Teilmengen stellen die gleiche BUB dar, da die Permutation  $\begin{pmatrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 4 & 2 & 7 & 5 & 3 \end{pmatrix}$  die erste in die zweite überführt. Teilmenge 3 läßt sich nicht weiter zerlegen. Wäre das nämlich der Fall, so müßte eine weitere Teilmenge mit  $b=7$  ( $\lambda=1$ ) entstehen. Wäre 123 darin enthalten, so müßten zwei weitere Blocks mit einer 1 auftreten, in denen die Paare 14, 15, 16 und 17 enthalten sind. Da gibt es aber nur die Möglichkeiten 145 und 167, da 157 nicht in Teilmenge 3 ist. Folglich müßte die Teilmenge die Blocks 123, 145 und 167 enthalten. Es fehlen noch zwei Blocks mit der 2 und mit den Paaren 24, 25, 26 und 27, aber nicht mit 23. Dafür gibt es die Möglichkeiten 246 und 257 oder 247 und 256. Mögliche Teilgruppen müßten entweder

123	oder	123
145		145
167		167
246		247
257		256

als erste fünf Blocks enthalten. In beiden Fällen müssen wegen  $r = 3$  noch zwei Blocks mit 3 auftreten. 345 und 367 sind aber wegen  $\lambda=1$  nicht möglich. Im ersten Fall wäre 347 möglich, aber es fehlt dann der erforderliche Ergänzungsblock 356. Im zweiten Falle wäre 357 möglich, es fehlt dann aber 346. Folglich gibt es von Teilmenge 3 keine Teilmenge mit  $b=7$  Blocks und dem Block 123. Analog zeigt man, daß es solche Teilmengen auch

nicht mit 125 bzw. 127 geben kann. Da aber das Paar 12 in einer Teilmenge (mit  $\lambda=1$  und  $b=7$ ) von Teilmenge 3 vorkommen muß, gibt es überhaupt keine solche Teilmenge mit  $\lambda=1$  und damit auch keine mit  $\lambda=2$ .

Damit ist  $f^*(7,3) \geq 2$  gezeigt. Wir wollen zeigen, daß das Gleichheitszeichen gilt. Es gibt 30 Möglichkeiten aus den  $\binom{7}{3}$  verschiedenen Blocks sieben Blocks einer BUB  $B(7,3,1)$  auszuwählen. Diese sind durch vier Blocks eindeutig festgelegt. Es sind dies für die erste Gruppe (mit dem Block 123) die BUB

Fall	Fall	Fall	Fall	Fall	Fall
1,1,1	1,1,2	1,2,1	1,2,2	1,3,1	1,3,2
123	123	123	123	123	123
145	145	146	146	147	147
167	167	157	157	156	156
246	247	245	247	245	246

Analog findet man die anderen  $4 \times 6$  Fälle ausgehend von den Blocks 124, 125, 126 bzw. 127. Man kann sofort sehen, daß alle Fälle durch Permutation aus dem Fall 1.1.1 erzeugt werden können. Zum Beispiel führt  $\binom{67}{76}$  auf Fall 1.1.2 oder  $\binom{56}{65}$  auf Fall 1.2.1 usw. Damit gibt es nur eine elementare BUB mit  $\lambda=1$ . Man kann sich nun leicht überlegen, daß es  $120 = 6 \cdot 8 + 6 \cdot 6 + 6 \cdot 4 + 6 \cdot 2$  Möglichkeiten gibt, zwei elementare BUB  $B(7,3,1)$  zu einer  $B(7,3,2) \in F^*(7,3)$  zusammzusetzen. In jeder der 5 Gruppen (charakterisiert durch den ersten Block  $12x$  ( $x = 3,4,5,6,7$ )) gibt es ja 6 Fälle ( $y,1,1$  bis  $y,3,2$  mit  $y = 1,2,3,4,5$ ). Zu jedem dieser Fälle gibt es in jeder anderen Gruppe zwei passende (elementenfremde) Partner. Zum Beispiel zu der BUB 123, 145, 167, 246, 257, 347, 356 gehört aus der zweiten Gruppe entweder

124, 136, 157, 237, 256, 345, 467

oder

124, 137, 156, 235, 267, 346, 457.

Damit gibt es auch genau 120 Möglichkeiten eine elementare BUB mit  $\lambda=3$  aufzuschreiben. Diese sind aber alle aus Permutationen auseinander erzeugbar. Das folgt daraus, daß durch eine be-

liebige Permutation die elementaren BUB-Bestandteile mit  $\lambda = 1$  der zusammengesetzten BUB mit  $\lambda = 2$  wieder in zwei elementare BUB mit  $\lambda = 1$  überführt werden und man außerdem durch geeignete Wahl der Permutationen auch alle elementaren BUB mit  $\lambda = 1$  erhält. Das vervollständigt den Beweis, da es für  $\lambda = 2$  keine elementare BUB gibt.

#### 5. Bestimmung von $f^*(8,3)$ und einer unteren Schranke für $f(8,3)$

Satz 6: Es gilt  $f^*(8,3) = 1$ .

Beweis: Aus (2) folgt  $\lambda = \frac{2r}{7}$  bzw.  $r = 7u$  ( $u = 1, 2, \dots$ ). Aus (1)

folgt aber  $r = \frac{7b}{8}$  bzw.  $b = \frac{56u}{3}$  und daraus  $u \geq 3$ . Mit  $u = 3$  exi-

stiert tatsächlich eine BUB mit den Parametern  $b = 56$ ,  $r = 21$  und  $\lambda = 6$  und diese ist identisch mit den  $\binom{8}{3}$  Kombinationen von 8 Elementen zur dritten Klasse. Es gilt außerdem  $f(8,3) \geq 2$ .

#### 6. Bestimmung von $f^*(9,3)$

Satz 7: In  $F^*(9,3)$  gibt es genau eine elementare BUB mit  $\lambda = 1$ , und es gilt  $f^*(9,3) \geq 2$ .

Beweis: Man kann die  $\binom{9}{3}$  Kombinationen von 9 Elementen zur dritten Klasse in vier elementare BUB  $B[9,3,1]$  und eine elementare BUB  $B[9,3,3]$  zerlegen.

Teilmenge						
1. TM	2. TM	3. TM	4. TM	5. TM		
124	129	125	123	126	234	358
138	134	136	147	127	236	389
159	156	148	158	128	237	459
167	178	179	169	135	245	467
235	238	239	248	137	249	468
269	246	247	259	139	256	469
278	257	268	267	145	258	478
346	359	345	349	146	279	569
379	367	369	357	149	289	578

1. TM	2. TM	3. TM	4. TM	5. TM	
457	458	378	368	157	347 579
489	479	567	456	168	348 678
568	689	589	789	189	356 679

Zunächst soll gezeigt werden, daß die fünfte Teilmenge nicht weiter zerlegbar ist. Das geschieht durch systematische Untersuchung aller Möglichkeiten, wir demonstrieren hier nur den Anfang. Wäre die fünfte Teilmenge zerlegbar (eine zusammengesetzte BUB), so müßte eine der Kombinationen mit 12 in einer  $B[9,3,1]$  auftreten. Das gibt drei mögliche Anfangsblocks 126, 127 und 128 und führt jeweils zu einem von drei zu untersuchenden Fällen. Wir beschränken uns auf die Gruppe mit 126. In der  $B[9,3,1]$  muß auch 13 auftreten, so daß es drei Unterfälle 126 126 126  
135' 137' 139 gibt. Wir beschränken uns auf die Untergruppe 126. Es muß die Kombination 14 in  $B[9,3,1]$  auftreten, d. h., es

gibt die Möglichkeiten

126	126	126
135	135	135
145	146	149

von denen die beiden ersten ausscheiden, da 15 bzw. 16 zweimal vorkommen würde ( $\lambda = 1$ ). Die dritte mögliche Gruppe müßte durch 178 ergänzt werden weil anders die vierte 1 nicht denkbar ist, ein solcher Block ist aber in der vierten Teilmenge nicht vorhanden.

Damit ist der Unterfall  $\begin{matrix} 126 \\ 135 \end{matrix}$  nicht möglich. Analog zeigt man die Unmöglichkeit aller anderen Fälle. Nun ist zu zeigen, daß die vier ersten Teilmengen durch Permutationen ineinander überführt werden können.

Durch  $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 4 & 9 & 5 & 7 & 3 & 2 & 6 & 1 \end{pmatrix}$  entsteht z. B. die zweite Teilmenge. aus der ersten und die dritte aus der zweiten.

Wir danken Herrn Harnau für den Hinweis, daß es auch für  $\lambda = 2$  elementare BUB gibt und damit  $f^*(9,3) > 2$  gilt (Harnau 1978).

Satz 8: Gegeben sei eine BUB  $B(v,k,\lambda^*)$  mit allen  $\binom{v}{k}$  Kombina-

tionen von  $v$  Elementen zur  $k$ -ten Klasse. Sie sei nur in  $a$  elementare BUB  $B(v, k, \lambda_0)$  und eine elementare BUB  $B(v, k, \lambda_1)$  mit  $\lambda_1 > \lambda_0$  zerlegbar. Ferner seien die  $a$  elementaren BUB  $B(v, k, \lambda_0)$  durch Permutationen ineinander überföhrbar. Dann ist  $f^*(v, k) = 2$ .

Beweis: Mit  $N$  bezeichnen wir die Anzahl der möglichen  $B(v, k, \lambda_0) \in F^*(v, k)$ , die wir uns systematisch (beginnend mit 123...  $k$  als ersten Block) aufgeschrieben denken. Durch diese systematische Auflistung ist abzulesen, wie alle möglichen dieser BUB aus einer festen durch Permutation erhalten werden können. Je vier elementefremde solche BUB definieren eine  $B(v, k, \lambda_1)$ , so daß auch alle möglichen BUB  $B(v, k, \lambda_1) \in F^*(v, k)$  durch Permutationen ineinander überföhrbar sind.

#### Zusammenfassung

Es werden vollständige und eingeschränkte  $(v, k)$ -Familien von balancierten unvollständigen Blockanlagen (BUB) und die Begriffe elementare und zusammengesetzte BUB eingeföhrt. Mit  $f(v, k)$  bzw.  $f^*(v, k)$  werden die Anzahlen der in vollständigen bzw. eingeschränkten  $(v, k)$ -Familien enthaltenen elementaren BUB bezeichnet. Für eine möglichst umfassende Randomisation ist es wichtig, alle elementaren BUB einer Familie zu kennen, da durch sie alle Elemente der Familie erzeugbar sind. Es gilt  $f(v, 2) = f^*(v, 2) = 1$  und  $f(v, k) = f(v, v-k)$  bzw.  $f^*(v, k) = f^*(v, v-k)$ . Es reicht also aus, die  $f(v, k)$  und  $f^*(v, k)$  für  $3 \leq k \leq \frac{v}{2}$  zu kennen. Es wird gezeigt, daß in  $f(v, k) \geq f^*(v, k)$  nicht immer das Gleichheitszeichen gilt. Ferner wird bewiesen, daß  $f^*(6, 3) = 1$ ,  $f^*(7, 3) = 2$ ,  $f^*(8, 3) = 1$  und  $f^*(9, 3) \geq 2$  gilt. Die Vermutung  $f(6, 3) \doteq 1$  wird durch einen Satz erhärtet.

#### Literatur

- Bose, R. G., On the construction of balanced incomplete block designs  
Ann. Eugenics (London) (1939) 9, S. 358-398

- Federer, W. T.; Balaam, L. N.,**  
 Bibliography on experiment and treatment  
 designs pre 1968  
 Hafner Publ. Comp. Inc. New York 1973
- Fisher, R. A.,** An examination of the different possible  
 solutions of a problem in incomplete blocks  
 Ann. Eugenics 10 (1940) S. 52 - 75
- Herrendörfer, G.; Rasch, D.,**  
 Ein modellunabhängiger Zugang zur Theorie  
 der Versuchsanlagen - balancierte unvoll-  
 ständige Blockanlagen (BUB)  
 Wissenschaftl. Sitzungen zur Stochastik  
 Akademie d. Wissensch. der DDR, ZIMM,  
 Berlin 1978
- Kirkman, T. P.,** On a problem in combinations  
 Cambridge Dublin Math. J. 2 (1847) S.  
 191 - 204
- Kirkman, T. P.,** Note on an unanswered prize question  
 Cambridge Dublin Math. J. 5 (1850) S.  
 255 - 262
- Rasch, D.,** Einführung in die mathematische Statistik  
 Band 2 Anwendungen, Kap. 8  
 VEB Deutscher Verlag der Wissenschaften,  
 Berlin 1976
- Steiner, J.,** Combinatorische Aufgabe  
 J. für die reine und angewandte Mathematik  
45 (1853) S. 181 - 182
- Harnau, W.,** Die Anzahl nichtisomorpher wiederholungs-  
 freier elementarer BUB mit  $\lambda = 2$  in der  
 (9,3)-Familie,  
 Preprint W.-Pieck-Universität, Rostock 1978

eingegangen: 28. 10. 1977

Anschriften der Verfasser:

Dozent Dr. Dieter Rasch; Dr. rer. nat. Günter Herrendörfer  
Akademie der Landwirtschaftswissenschaften der DDR  
Forschungszentrum für Tierproduktion Dummerstorf-Rostock  
DDR-2551 Dummerstorf

Rolf Fröhlich

### Eine Verallgemeinerung der BACKUS-Systeme

Die BACKUS-Systeme haben seit ihrer Veröffentlichung mit dem Bericht für Algol 60 eine breite Anwendung zur Beschreibung der syntaktischen Grundstruktur von Programmiersprachen sowie auch anderer Kommunikationssprachen der Informationsverarbeitung gefunden. Die Ursache hierfür mag wohl in folgenden Vorzügen der BACKUS-Systeme liegen:

1. Die metalinguistischen Variablen, die in den üblichen Regelgrammatiken den Nichtterminalsymbolen entsprechen, sind durch Wörter der Umgangssprache darstellbar. Diese Wörter können begrifflich so gewählt werden, daß eine bestimmte syntaktische Einheit bereits semantisch näher charakterisiert wird.
2. Eine syntaktische Einheit ist durch eine einzige Regel formulierbar, wobei alle möglichen Formen dieser syntaktischen Einheit alternativ erfaßt sind.

Diese beiden Aspekte machen maßgeblich die BACKUS-Notation als Ausdrucksmittel für die Syntax sehr übersichtlich und anwenderfreundlich. Darüberhinaus stehen sie hinsichtlich der Exaktheit der Beschreibung den entsprechenden Regelgrammatiken nicht nach.

Eine bekannte Einschränkung der BACKUS-Systeme ist, daß sich mit ihnen nur kontextfreie Sprachen (Typ-2-Sprachen nach der CHOMSKY-Klassifikation) definieren lassen, obwohl gerade fast alle Programmiersprachen Kontextabhängigkeiten enthalten. Dieser Umstand war u. a. Ausgangspunkt für die Entwicklung weiterer Ausdrucksmittel, mit denen solche Kontextabhängigkeiten formal beschrieben werden können. Stellvertretend für die Vielzahl der Ergebnisse seien genannt die programmierten, die dynamischen, die mehrschichtigen Grammatiken. Ein weiterer Ansatz in dieser Richtung ist durch eine Verallgemeinerung der BACKUS-Systeme gegeben.

Welche Forderungen sind an derart verallgemeinerte BACKUS-Systeme zu stellen? Nimmt man die Verallgemeinerung auf eine natürliche Weise vor, so muß offensichtlich erfüllt sein:

1. Das verallgemeinerte BACKUS-System muß das übliche BACKUS-System als Spezialfall enthalten.
2. Mit einem verallgemeinerten BACKUS-System muß sich eine beliebige Regelsprache (Typ-0-Sprache nach der CHOMSKY-Klassifikation) definieren lassen.
3. Ein verallgemeinertes BACKUS-System muß die typischen Merkmale eines üblichen BACKUS-Systems besitzen. Als typische Merkmale eines BACKUS-Systems werden hierbei angesehen:
  - a) Die Produktionsregeln bestehen auf ihren linken Seiten aus genau einem Nichtterminalsymbol.
  - b) Die Produktionsregeln mit gleichen linken Seiten sind zu einer einzigen Regel zusammengefaßt.
  - c) Die Nichtterminalsymbole sind durch Wörter dargestellt, so daß die syntaktischen Einheiten semantisch charakterisiert werden können.

Im folgenden wird eine Verallgemeinerung der BACKUS-Systeme skizziert, welche die genannten Forderungen erfüllt. Zur Vereinfachung der Darstellungsweise soll zunächst auf die letzte Forderung c) verzichtet werden, d. h., die Nichtterminalsymbole werden vorläufig o. B. d. A. durch kleine griechische Buchstaben dargestellt.

Der Ausgangspunkt der Betrachtung ist der Zusammenhang der kontextfreien Regelgrammatiken mit den (üblichen) BACKUS-Systemen. Es sei  $G = [\Sigma, \Phi, R, \sigma]$  eine kontextfreie Grammatik mit den bekannten Eigenschaften:

$$\begin{array}{ll}
 \Sigma & \text{Terminalalphabet,} \\
 \Phi & \text{Nichtterminalalphabet,} \\
 \Gamma & = \Sigma \cup \Phi \text{ Vokabular mit } \Sigma \cap \Phi = \emptyset, \\
 R & \text{Regelsystem mit } R \subseteq \Phi \times \Gamma^*, \\
 \sigma & \text{Startsymbol mit } \sigma \in \Phi
 \end{array}$$

Mit dem durch  $G$  implizierten Semi-THUE-System  $[\Gamma, R]$  ist der übliche Ableitungsbegriff gegeben.

Die Regeln aus  $R$  besitzen die Form

$$\xi \rightarrow x \text{ mit } \xi \in \Phi, x \in \Gamma^* .$$

In  $R$  ist nun eine Klasseneinteilung vermöge der Äquivalenzrelation gls (gleiche linke Seite) definiert:

$$" \xi \rightarrow x \text{ gls } \xi' \rightarrow x' \text{ genau dann, wenn } \xi = \xi' " .$$

Eine solche Klasse sei mit  $[\xi]$  bezeichnet, so daß durch  $\xi' \rightarrow x' \in [\xi]$  die Zugehörigkeit einer Regel zu ihrer Klasse notiert werden kann.

Unter diesen Voraussetzungen ist  $B = [\Sigma, \Phi, BR, \sigma]$  ein (übliches) BACKUS-System, wobei gilt

$$BR = \left\{ \xi ::= x_1 | x_2 | \dots | x_n \mid \xi \rightarrow x_i \in [\xi], i = 1, 2, \dots, n, \right. \\ \left. \text{für jede Klasse } [\xi] \text{ von } R \right\} .$$

Der Ableitungsbegriff mit  $B$  wird vom Semi-THUE-System  $[\Gamma, R]$  übernommen, in dem jeweils aus einer BACKUS-Regel  $\xi ::= x_1 | x_2 | \dots | x_n$  eine Produktionsregel  $\xi \rightarrow x_i$  ( $1 \leq i \leq n$ ) ausgewählt wird.

Offensichtlich sind dann  $G$  und  $B$  äquivalent, denn man zeigt leicht, daß  $L(G) = L(B)$  gilt. Die üblichen BACKUS-Systeme definieren folglich nur Typ-2-Sprachen.

Die eben skizzierte Methodik zur Herleitung der üblichen BACKUS-Systeme wird jetzt auf beliebige Regelgrammatiken (Typ-0-Grammatiken) angewendet. Es sei dazu  $G = [\Sigma, \Phi, R, \sigma]$  eine Typ-0-Grammatik, das heißt, für das Regelsystem  $R$  gilt

$$R \subseteq \Gamma^* \Phi \Gamma^* \times \Gamma^* .$$

Die Regeln besitzen folglich mindestens ein Nichtterminalsymbol auf ihren linken Seiten und sind von der Form

$$u \xi v \rightarrow x \text{ mit } \xi \in \Phi; u, v, x \in \Gamma^* .$$

Eine Umformung dieser Regeln geschieht wie folgt:

1. Auf der linken Seite jeder Regel wird an genau einer Stelle ein Nichtterminalsymbol  $\xi$  ausgewählt.

2. Jede Regel  $u \xi v \rightarrow x \in R$  wird in Form

$$\xi \rightarrow u' w v'$$

dargestellt, wobei gilt:

- $u', v'$  sind die "inversen" Wörter von  $u, v$  (die Inversenbildung wird nachfolgend interpretiert),
- $u$  heißt der Links- und  $v$  der Rechtskontext der Regel,
- $w = \perp$ , falls  $|u| + |v| \neq 0$  und  $x = \acute{e}$  bzw.  $w = x$  sonst ( $|u|, |v|$  bedeuten hier die Länge von  $u, v$  und  $e$  sei das leere Wort).
- $\perp$  ist ein zusätzliches, ausgezeichnetes Symbol, das Kontexttrennzeichen genannt wird, und für das gilt:

$$\Phi \cap \{\perp\} = \emptyset$$

Zur Interpretation der inversen Wörter wird das Vokabular  $\Gamma$  wie folgt erweitert. Ist  $A$  ein Alphabet, so heie  $A' = \{a' \mid a \in A\}$  das zu  $A$  inverse Alphabet.  $\mathcal{A} = A \cup A'$  soll dann das gebrochene Alphabet von  $A$  genannt werden. Freilich wird hierbei  $A \cap A' = \emptyset$  gefordert. Mit diesen Erklrungen verstehen sich die Bezeichnungen

$$\mathcal{A}' = \Sigma \cup \Sigma', \quad \mathcal{A} = \Phi \cup \Phi', \quad \mathcal{A} = \Gamma \cup \Gamma'.$$

Sei nun  $a = a_1 a_2 \dots a_n \in A^*$  ein beliebiges Wort in Symboldarstellung ( $a_i \in A$ ); dann ist das  $a$  zugeordnete inverse Wort  $a'$  von der Gestalt  $a' = a'_n a'_{n-1} \dots a'_1$  mit  $a'_i \in A'$ . Das entspricht der blichen Inversenbildung wie sie in freien Gruppen erklrt ist.

Wir nennen nun  $G' = [\Sigma, \Phi, R', \mathcal{A}', \perp]$  eine kontextisolierte Grammatik (kurz: ci-Grammatik) von  $G = [\Sigma, \Phi, R, \mathcal{A}]$ ,

wobei gilt:

$$\begin{aligned} R' \subseteq & \Phi \times \Gamma'^* \Gamma^+ \Gamma'^* \cup \\ & \Phi \times \Gamma'^* \{\perp\} \Gamma'^* \cup \\ & \Phi \times \{e\} \end{aligned}$$

$\Gamma^+$  bedeute hierbei  $\Gamma^* - \{e\}$ .

Damit ist eine erste Forderung, die linken Seiten von  $R'$  bestehen aus genau einem Nichtterminalsymbol, erfüllt, um zu einem verallgemeinerten BACKUS-System übergehen zu können. Man sieht zugleich, sind die Links- und Rechtskontexte sämtlicher Regeln von  $R'$  leer, so ist  $G'$  eine übliche kontextfreie Grammatik.

Zunächst muß jedoch eine Anpassung des Ableitungsmechanismus vorgenommen werden. Zu diesem Zwecke sei  $\mathcal{A}^{\varphi}$  die freie Gruppe über dem Alphabet  $A$ . Es gibt dann eine Reduktion  $\varphi$ , die jedes Wort  $a \in \mathcal{A}^*$  auf ein reduziertes Wort  $\varphi(a) \in \mathcal{A}^{\varphi}$  derart abbildet, daß in  $\varphi(a)$  vermöge  $\varphi(a_i a_j) = \varphi(a_j a_i) = e$  nicht mehr zwei Symbole  $a_i a_j$  bzw.  $a_j a_i$  nebeneinander stehen. Die Elemente der freien Gruppe  $\mathcal{A}^{\varphi}$  seien also durch die reduzierten Wörter aus  $\mathcal{A}^{\varphi}$  dargestellt. Weiterhin sei  $\psi$  eine Abbildung mit  $\psi(\perp) = e$  bzw.  $\psi(x) = x$  für  $x \in A^*$ . Unter diesen Voraussetzungen ist eine  $\varphi$ -Ableitung mit  $G'$  wie folgt definiert:

Ein Wort  $q \in \Gamma^*$  heißt aus einem Wort  $p \in \Gamma^+ \varphi$ -abgeleitet, wenn es eine Regel  $\xi \rightarrow u' w v' \in R'$  und Wörter  $r, s \in \Gamma^*$  gibt derart, daß

$p = r u \xi v s$  und  $q = r \varphi(u u') \psi(w) \varphi(v' v) s$  gilt. Das heißt, entweder ist  $q = r s$ , falls  $w = \perp$ , oder es ist  $q = r x s$ , falls  $w = x$  mit  $x \in \Gamma^*$ .

Man sieht wieder: Wenn  $G'$  eine kontextfreie Grammatik ist, stimmt die  $\varphi$ -Ableitung mit dem üblichen Ableitungsbegriff als Spezialfall überein.

Man kann nun ohne viel Mühe zeigen: Ist  $G$  eine Typ-0-Grammatik und  $G'$  eine ihr zugeordnete ci-Grammatik, so gilt  $L(G) = L(G')$ .

Damit haben wir das angestrebte Ziel erreicht und erklären das ci-BACKUS-System:

Ist  $G' = [\Sigma, \Phi, R', \sigma, \perp]$  eine ci-Grammatik, so heißt

$B' = [\Sigma, \Phi, BR', \sigma, \perp]$  ein ci-BACKUS-System, wobei gilt

$$BR' = \left\{ \xi ::= u_1 w_1 v_1 \mid u_2 w_2 v_2 \mid \dots \mid u_n w_n v_n \right\}$$

$$\xi \rightarrow u_i w_i v_i \in [\xi], \quad i = 1, 2, \dots, n,$$

für jede Klasse  $[\xi]$  von  $R'$  .

Die Klasseneinteilung erfolgt hierbei wieder mit der Relation gls.

Als Ergebnis erhält man: Für jede Regelsprache  $L$  (vom CHOMSKY-Typ 0) gibt es ein ci-BACKUS-System  $BR'$ , so daß  $L = L(BR')$  gilt.

Eine Vereinfachung findet noch statt, wenn man sich auf die Typ-1-Sprachen beschränkt.

Sei  $G'$  eine beschränkte ci-Grammatik, das heißt, die sämtlichen Regeln von  $G'$  erfüllen die Bedingungen

$$(1) \xi \rightarrow u' x v' \text{ mit } \xi \in \Phi; u', v' \in \Gamma^* ; x \in \Gamma^*$$

oder

$$(2) \xi \rightarrow e \text{ und } \xi \text{ kommt in keinem Wort } x \text{ von (1) vor.}$$

Man kann jetzt auch das Kontexttrennzeichen weglassen und zeigen:

Für jede Typ-1-Sprache  $L$  gibt es eine beschränkte ci-Grammatik  $G'$  mit  $L = L(G')$ .

Dieses letzte Resultat läßt sich wieder auf "beschränkte" ci-BACKUS-Systeme übertragen, die dann bereits durch

$$B' = [ \Sigma, \Phi, BR', \zeta ] \text{ festgelegt sind.}$$

Um die ci-BACKUS-Systeme in konkreten Fällen als Ausdrucksmittel anwenden zu können, müssen noch einige reservierte Metasymbole, vor allem für die Notation der inversen Wörter festgelegt werden. Neben den üblichen Symbolen wie

	Alternativzeichen,
:: =	Definitionszeichen,
< >	Metaklammern für die Bezeichnungen der Nichtterminalsymbole,
a, b, c, ...	Symbole eines Beschreibungsalphabetes $A$ , für das nicht notwendig $A \cap \Sigma = \emptyset$ gelten muß, und die zur Bildung von Bezeichnungen

für die syntaktischen Einheiten benutzt werden,

führen wir die Symbole

[ ] als Metaklammern für die Inversen ein.

Anstelle von [ und ] können auch andere geeignete Metasybole Verwendung finden. Insbesondere kann man Linksinversenklammern und unterschiedliche Rechtsinversenklammern vereinbaren.

### Beispiel

Das ci-BACKUS-System  $B' = [ \Sigma, \Phi, BR', \langle \text{zuerst} \rangle ]$  mit

$\Sigma = \{ a, b \},$

$\Phi = \{ \langle \text{zuerst} \rangle, \langle \text{dann} \rangle, \langle \text{zuletzt} \rangle \},$

$BR' = \{ \langle \text{zuerst} \rangle ::= a \langle \text{zuerst} \rangle \langle \text{dann} \rangle \{ a \langle \text{zuletzt} \rangle, \langle \text{dann} \rangle ::= [ \langle \text{zuletzt} \rangle ] b \langle \text{zuletzt} \rangle a [ a ] \langle \text{dann} \rangle a, \langle \text{zuletzt} \rangle ::= b a \},$

definiert die Typ-1-Sprache  $L(BR') = \{ a^n b^n a^n \mid n \in \mathbb{N} \}.$

Eine Ableitung von  $a^3 b^3 a^3$  mit  $B'$  ist zum Beispiel:

$\langle \text{zuerst} \rangle \rightarrow a \langle \text{zuerst} \rangle \langle \text{dann} \rangle$   
 $\rightarrow aa \langle \text{zuerst} \rangle \langle \text{dann} \rangle \langle \text{dann} \rangle$   
 $\rightarrow aaa \langle \text{zuletzt} \rangle \langle \text{dann} \rangle \langle \text{dann} \rangle$   
 $\rightarrow aaa \varphi (\langle \text{zuletzt} \rangle [ \langle \text{zuletzt} \rangle ] ) b \langle \text{zuletzt} \rangle a \langle \text{dann} \rangle$   
 $= aaab \langle \text{zuletzt} \rangle a \langle \text{dann} \rangle$   
 $\rightarrow aaab \langle \text{zuletzt} \rangle \varphi ( a [ a ] ) \langle \text{dann} \rangle a$   
 $= aaab \langle \text{zuletzt} \rangle \langle \text{dann} \rangle a$   
 $\rightarrow aaab \varphi (\langle \text{zuletzt} \rangle [ \langle \text{zuletzt} \rangle ] ) b \langle \text{zuletzt} \rangle aa$   
 $= aaabb \langle \text{zuletzt} \rangle aa$   
 $\rightarrow aaabbbaaa.$

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. Rolf Fröhlich  
Technische Hochschule Karl-Marx-Stadt  
Sektion Rechentechnik und Datenverarbeitung  
DDR-90 Karl-Marx-Stadt, Straße der Nationen



## Eine Betrachtung zur formalen Beschreibung von Programmiersprachen und Sprachverarbeitungssystemen

---

### 1. Zur Benutzung formaler Sprachbeschreibungen

---

Die Schaffung, die Definition, das Erlernen und die Implementierung von Programmiersprachen geschieht heute - gemessen an den hohen Anforderungen - mit noch relativ unangemessenen Mitteln und zu hohem Aufwand. Es sind deshalb seit langem Bestrebungen im Gange, formalsprachliche Mittel zur Beschreibung von Programmiersprachen und Sprachverarbeitungssystemen bereitzustellen und die Entwicklung von Sprachverarbeitungssystemen zu automatisieren oder zumindest durch den Automaten zu unterstützen.

Formale Beschreibungen von Programmiersprachen bzw. Sprachverarbeitungssystemen können für folgende Zwecke eingesetzt werden:

- für eindeutige, systematische und konzise Sprachreports ,
- als Zwischenstadium zur Erarbeitung von umgangs- (fach-) sprachlichen Anwenderdokumentationen ,
- als Entwurfssprache für die Konzipierung von Sprachen und Sprachverarbeitungssystemen und -algorithmen ,
- für exakte Festlegung von Leistungsanforderungen, Aufgabenstellungen und Übergabeleistungen ,
- für die Aufteilung eines Programmierproblems in wohldefinierte Teilprobleme unter Festlegung von Anschlußbedingungen ,
- zum Vermitteln bestimmter Prinzipien der Konstruktion von Programmiersprachen und spezieller Techniken für Sprachverarbeitungssysteme ,
- als Eingangssprache eines Simulationssystems, das Funktionserprobungen und Ermittlungen von Eigenschaften ermöglicht ,

- als Referenz zur teilautomatisierten Testung von ausprogrammierten Sprachverarbeitungssystemen ,
- als Eingangsinformation zur (teil-)automatisierten Herstellung von Sprachverarbeitungssystemen ,
- als Basisinformation für die automatengestützte Unterweisung über Programmiersprachen.

Die bisher aus der Literatur bekannten einschlägigen Arbeiten haben zum großen Teil entweder eine theoretische Orientierung oder sind unmittelbar im Zusammenhang mit Automatisierungsvorhaben (Compiler - Compiler) entwickelt worden. Formale Beschreibungen, die dem größeren Teil des vorn aufgeführten Anwendungsspektrums genügen, fehlen bisher weitgehend. Wir haben mit unseren Arbeiten versucht, einen Beitrag zum Schließen dieser Lücken zu leisten.

## 2. Programmiersprachen, Sprachverarbeitungssysteme und deren formale Beschreibung

Es soll hier dargelegt werden, was wir unter den Begriffen Programmiersprache und Sprachverarbeitungssystem verstehen wollen. Weiterhin werden wir unsere Vorstellungen zur formalen Beschreibung von Programmiersprachen und Sprachverarbeitungssystemen formulieren.

### 2.1. Programmiersprachen

Wir wollen bei Programmiersprachen im folgenden zwei Aspekte, die Syntax und die Semantik, unterscheiden. Will man den Begriff Programmiersprache formal erfassen, dann kann man eine Programmiersprache  $FS$  als ein Paar  $(P, \sigma)$  betrachten, wobei  $P$  die Syntax und  $\sigma$  die Semantik repräsentiert /Mau69/.  $P$  ist die Menge der in  $FS$  zulässigen Sätze, die bei Programmiersprachen Programme genannt werden.  $\sigma$  definiert die Bedeutung jedes Elementes aus  $P$  und legt damit die Wirkung eines jeden Programms fest.

Die beiden Komponenten  $P$  und  $\sigma$  sollen nun mit mathematischen

Mitteln charakterisiert werden. Dazu sei  $X$  das Alphabet, aus dessen Zeichen die Sätze von  $PS$  gebildet werden.  $P$  ist dann also eine Teilmenge der Menge der Worte über dem Alphabet  $X$ ,  $P \leq X^* \cdot X^* = S$  ist die Menge der Worte über dem Alphabet  $X$ . Das Alphabet  $X$  wollen wir als Grundalphabet bezeichnen. Die Menge  $P$  kann man sich durch eine Regelgrammatik definiert denken /Gin66, Mau69/. Es sei  $G = (X, M, R, w)$  eine solche Grammatik, wobei  $X$  die Menge der Terminalsymbole (Grundalphabet),  $M$  die Menge der metasprachlichen Variablen (Hilfssymbole),  $R = \{(1, r) \mid 1, r \in (X \cup M)^*\}$  die Regelmenge und  $w \in M$  die Wurzel der Grammatik sind.  $P$  kann nun als die Menge derjenigen Worte über dem Grundalphabet  $X$  angesehen werden, die mit Hilfe einer Grammatik aus der Wurzel ableitbar sind /Mau69/. Soll  $P$  durch die Grammatik  $G$  definiert werden, so kann man schreiben:

$$P(G) = \{p \mid p \in S \wedge w \Rightarrow p\}.$$

" $w \Rightarrow p$ " bedeutet, daß  $p$  aus  $w$  ableitbar ist.

Zur Erläuterung dessen, was wir unter Semantik einer Programmiersprache verstehen wollen, werden noch folgende Mengen benötigt:

- $E \leq S$  ist die Menge der Eingabesätze für Programme der Programmiersprache  $PS$  (Eingabesprache, Datensprache).
- $A \leq S$  ist die Menge der Ausgabesätze für Programme der Programmiersprache  $PS$  (Ausgabesprache, Ergebnissprache).

Zur Definition der Semantik von  $PS$  wird vorausgesetzt, daß die Mengen  $E$  und  $A$  bekannt sind.

Die Semantik der Programmiersprache  $PS = (P, \epsilon)$ , die ja die Wirkung jedes  $PS$ -Programms festlegen muß, wird durch eine partielle Abbildung  $\epsilon : P \times E \rightarrow A$  definiert. Partiiell ist die Abbildung, weil

- a) für ein gegebenes  $p \in P$  nicht jedes  $e \in E$  ein zulässiger Eingabesatz sein muß und
- b) die Abarbeitung des Programms  $p \in P$  unter Umständen nicht abbricht, wenn zum Beispiel unendliche Zyklen durch Sprunganweisungen oder rekursive Proseduraufrufe enthalten sind.

Die Abbildung  $\sigma$  ordnet also gewissen Paaren  $(p, e)$  mit  $p \in P$  und  $e \in E$  ein  $a \in A$  zu. In der Praxis sind natürlich nur diejenigen Fälle interessant, in denen  $\sigma$  einen Wert liefert.

Es wird nun ein spezielles Programm  $p' \in P$  betrachtet. Als Wirkung von  $p'$  kann die partielle Abbildung  $\tilde{\pi}: E \rightarrow A$  angesehen werden, die sich aus  $\sigma$  durch die Beziehung  $\tilde{\pi}(e) = \sigma(p', e)$  ergibt. Die Partialität von  $\tilde{\pi}$  folgt damit unmittelbar aus der Partialität von  $\sigma$ .

## 2.2. Sprachverarbeitungssysteme

Als Sprachverarbeitungssystem einer Programmiersprache PS wollen wir eine Einrichtung bezeichnen die bei einer gegebenen Zeichenkette  $P \in S = X^*$  und einem gegebenen Eingabesatz  $e \in E$  folgendes ausführt:

1. Prüfung, ob  $p$  ein Satz der Programmiersprache PS ist
2. Berechnung von  $\sigma(p, e)$ , falls sich  $p$  als Satz von PS erwiesen hat.

Die Wirkung eines Verarbeitungssystems für PS kann wieder durch eine partielle Funktion  $\varphi: S \times E \rightarrow A \cup D$  definiert werden.  $D \neq S$  ist dabei eine endliche Menge von Diagnosesätzen, die einen bei der Verarbeitung aufgetretenen Programm- oder Datenfehler anzeigen. Genau wie  $E$  und  $A$  muß auch die Menge  $D$  gegeben sein. Es werden zwei Fehlerarten und damit zwei disjunkte Teilmengen  $D_1$  und  $D_2$  von  $D$  unterschieden ( $D = D_1 \cup D_2$ ,  $D_1 \cap D_2 = \emptyset$ ).  $D_1$  ist die Menge von Fehlermitteilungen für die Fälle, daß die gegebene Zeichenkette  $p$  kein Satz von PS ist ( $p \notin P$ ), während  $D_2$  die Menge von Mitteilungen für die Fälle ist, daß  $e \in E$  kein zulässiger Eingabesatz für  $p \in P$  ist. Um unterscheiden zu können, ob ein Wert  $\varphi(p, e)$  ein normales Ergebnis  $\varphi(p, e) \in A$  oder eine Diagnosemitteilung  $\varphi(p, e) \in D$  ist, müssen  $A$  und  $D$  disjunkte Mengen sein ( $A \cap D = \emptyset$ ).

Bei der Anwendung von  $\varphi$  auf ein  $p \in S$  und ein  $e \in E$  können folgende Fälle auftreten:

1.  $\varphi$  liefert bei Anwendung auf  $p$  und  $e$  keinen Wert (möglich wegen Partialität von  $\varphi$ ). In diesem Fall bricht der Sprachverarbeitungsprozeß nicht ab.
2.  $\varphi(p, e) \in D_1$ ;  $p$  ist kein Satz der Programmiersprache PS ( $p \in P$ ).
3.  $\varphi(p, e) \in D_2$ ;  $e$  ist kein gültiger Eingabesatz für  $p$ .
4.  $\varphi(p, e) \in A$ ,  $\varphi$  liefert auf  $p$  und  $e$  angewendet mit dem Ausgabesatz  $\varphi(p, e)$  ein gültiges Ergebnis.

Die Syntax und die Semantik einer Programmiersprache  $PS = (P, \sigma)$  müssen in dem Verarbeitungssystem für PS in der im folgenden beschriebenen Weise ihren Niederschlag finden.

a) Syntax

Auf Grund der Entscheidungsmöglichkeit, ob  $\varphi(p, e) \in D_1$  ist, wird durch  $\varphi$  eine (u. U. unvollständige) syntaktische Analyse durchgeführt. Vollständig kann die Analyse dann sein, wenn die Menge der Sätze der Programmiersprache entscheidbar ist /Her61/. Im allgemeinen Fall, wenn Sprachen vom Chomsky-Typ-0 vorliegen, sind diese Mengen aber im allgemeinen nicht entscheidbar.

Ist  $P$  entscheidbar, wie das bei den kontextfreien Sprachen der Fall ist, so existiert ein abbrechendes Entscheidungsverfahren  $\xi : S \rightarrow \{0,1\}$ , das  $P$  auf folgende Weise implizit definiert:  $P = \{p \mid \xi(p) = 1\}$ . Dieses Entscheidungsverfahren kann dann Bestandteil des Sprachverarbeitungssystems sein, das dann eine vollständige Syntaxanalyse durchführt.

b) Semantik

Für ein Programm  $p \in P$  und einen Eingabesatz  $e \in E$  muß das Sprachverarbeitungssystem für PS natürlich die gleichen Ergebnisse liefern, wie das durch  $\sigma$  für PS festgelegt wird. Ein Unterschied kann nur dann auftreten, wenn  $\sigma(p, e)$  wegen eines unzulässigen Eingabesatzes undefiniert ist. Aus praktischen Gründen wird das Sprachverarbeitungssystem in solchen Fällen einen Diagnosesatz  $d \in D_2$  ausgeben, falls das möglich ist. Für  $p \in P$  gilt  $\varphi(p, e) = \sigma(p, e)$ , falls  $\sigma(p, e)$

existiert.

In der Praxis wird  $\varphi$  meist aus mehreren Abbildungen zusammengesetzt ( $\varphi = [\varphi_1, \dots, \varphi_n]$ ). Bei dieser Trennung spielen praktische verfahrens- und programmierungstechnische Gründe die wesentliche Rolle. Bei der Zerlegung  $\varphi = [\varphi_1, \dots, \varphi_n]$  gilt:

$$\varphi_i : P_{i-1} \rightarrow P_i \cup D_i, \quad i = 1, \dots, n-1,$$

$$\varphi_n : P_{n-1} \times E \rightarrow A \cup D.$$

$P_0 = S$ ,  $P_i$  sind Zwischensprachen,  $D_i \cap P_i = \emptyset$   $i = 1, \dots, n-1$ .

Auf Grund der notwendigen Unterscheidbarkeit müssen alle Zwischensprachen  $P_i$  zu  $D_i$  disjunkt sein.

$\varphi$  ergibt sich aus den Komponenten in folgender Weise:

$$\varphi(p, e) = \begin{cases} \varphi_j \circ \dots \circ \varphi_1(p), & \text{falls } \varphi_j \circ \dots \circ \varphi_1(p) \in D \wedge j < n \\ & \wedge \exists k < j : \varphi_k \circ \dots \circ \varphi_1(p) \in D \\ \varphi_n(\varphi_{n-1} \circ \dots \circ \varphi_1(p), e), & \text{sonst.} \end{cases}$$

Wir wollen die ersten  $n-1$  Schritte der Verarbeitung, bei denen in eine Zwischensprache abgebildet wird, als Übersetzungsschritte und den letzten, bei dem unter Verwendung der Daten  $e \in E$  das Ergebnis  $a \in A$  ermittelt wird, als Interpretationsschritt bezeichnen.

Beispiele für Verarbeitungssysteme einer Programmiersprache PS  
Beispiel (1).

$$\varphi_1 : S \rightarrow MS \cup D_1;$$

$P_1 = MS$ ,  $MS$  ist die Maschinensprache eines speziellen Rechners;

$$\varphi_2 : MS \times E \rightarrow A \cup D.$$

$\varphi_1$  ist eine Übersetzung von in PS geschriebenen Programmen in Maschinensprache.  $\varphi_2$  ist eine Interpretation der Maschinenprogramme unter Verwendung von Eingabedaten. Die Funktion  $\varphi_2$  ist nur für die  $mp \in MS$  definiert, für die  $mp = \varphi_1(p)$  mit  $p \in P$

gilt. Sie ist dort, wo sie definiert ist, äquivalent zur Arbeitsfunktion  $q : MS \times S \rightarrow S$  des Rechners mit der Maschinensprache  $MS$  (Interpreter der Maschinensprache). Man kann deshalb  $q(mp, e)$  statt  $\varphi_2(mp, e)$  schreiben. Das gilt im folgenden entsprechend für alle Abbildungen, die Einschränkungen von  $q$  auf kleinere Definitionsgebiete sind.

Beispiel (2).

$\varphi_1$  : Syntaxanalyse;

Aufgabe der Syntaxanalyse ist es, neben der Prüfung, ob eine gegebene Zeichenkette Satz der Sprache ist oder nicht, ein strukturiertes Programm, häufig auch Ableitungsbaum oder Syntaxbaum genannt, in dem semantisch zusammengehörende Teile zusammengefaßt sind, aufzubauen. In der Praxis wird die Analyse gewöhnlich nach einer kontextfreien Grammatik durchgeführt, durch die allerdings häufig nicht genau die Sätze der Sprache erfaßt werden.

$P_1$  : Zwischensprachen zur Darstellung des strukturierten Programms (z. B. Listensprache);

$\varphi_2$  : Interpretation des strukturierten Programms.

Beispiel (3).

$\varphi_1$  :  
 $P_1$  : } wie Beispiel (2);

$\varphi_2$  : Optimierung des Programms in strukturierter Form;

$P_2 = P_1$ ,

$\varphi_3$  : Übersetzung des optimierten Programms in ein Maschinenprogramm,

$P_3$  : Maschinensprache,

$\varphi_4 = Q$  : Arbeitsfunktion des Rechners.

Das Beispiel (2) kann als Modell dafür dienen, wie sich der Mensch die Verarbeitung einer Sprache vorstellt, wenn es ihm

vor allem auf die Wirkung der einzelnen Sprachelemente ankommt und nicht so sehr auf reale Verarbeitung bei einem in der Praxis eingesetzten Verarbeitungssystem.

Im Beispiel (3) wird ein Verarbeitungsprozeß dargestellt, wie er für praktische Compiler typisch ist. Nach der erfolgten Syntaxanalyse wird eine Optimierung durchgeführt, bevor das Maschinenprogramm generiert wird. Das angegebene Modell beschreibt natürlich auch nur grob den Gesamtprozeß. Zum Beispiel setzt sich  $\Phi_1$  gewöhnlich aus lexikalischer Analyse und eigentlicher Syntaxanalyse zusammen und es werden noch vorbereitende Arbeiten für die sich anschließenden Phasen, zum Beispiel Aufstellen der Symboltabelle, durchgeführt.  $P_1$  muß dann natürlich das Darstellen aller Ergebnisse, zum Beispiel auch der Symboltabelle, ermöglichen.

Die Beispiele (2) und (3) sind für uns besonders interessant, weil sie Modelle für die Beschreibung von Sprachverarbeitungssystemen darstellen. Werden die in den Beispielen auftretenden Prozesse beschrieben, dann kann im Falle des Beispiels (2) diese Beschreibung als eine für den Benutzer der Sprache geeignete Definition der Sprachverarbeitung angesehen werden, während beim Beispiel (3) die reale Sprachverarbeitung beschrieben wird. Setzt man bei einer durch  $\Phi = [\Phi_1, \dots, \Phi_n]$  dargestellten Sprachverarbeitung für FS eine Zwischensprache  $P_i$ ,  $i < n$ , als bekannt und definiert voraus, so kann man sich bei der Definition der Sprachverarbeitung für FS mit der Angabe von  $\Phi' = (\Phi_1, \dots, \Phi_i)$  begnügen.  $\Phi'$  ist dann eine Übersetzung von FS in die Zwischensprache  $P_i$ .

Beispiel (4).

FS sei eine Spezialsprache, die eine Erweiterung von ALGOL sei.

$\Phi_1$  :  
 $P_1$  : } wie Beispiel (2);

$\Phi_2$  : Übersetzung des strukturierten Programms in ALGOL (Vorübersetzung);

$P_2$  : ALGOL.

Von ALGOL als definiert ausgehend, kann ein Verarbeitungssystem für die Spezialsprache FS durch Angabe von  $(\varphi_1, \varphi_2)$  als hinreichend dargestellt betrachtet werden.

### 2.3. Formale Beschreibung

#### 2.3.1. Formale Beschreibung von Programmiersprachen

Zur formalen Beschreibung einer Programmiersprache  $FS = (P, \sigma)$  müssen die Menge  $P$  und die Abbildung  $\sigma$  beschrieben werden. Im Abschnitt 2.1. wurde bereits gesagt, daß  $P$  durch eine Regelgrammatik definiert werden kann. Wird die Regelgrammatik in einer geeigneten Sprache aufgeschrieben, so ist damit die formale Beschreibung von  $P$  erfolgt. Zur Darstellung kontextfreier Regelgrammatiken /Mau69, Gin66/ wird seit dem Erscheinen von ALGOL 60 größtenteils die Backus-Naur-Form (abgekürzt BNF) verwendet /ALGOL60, Mau69/. Eine für die Beschreibung einer größeren Klasse von Regelgrammatiken (Chomsky-Typ-0) geeignete Notationsform wurde mit dem ALGOL-68-Bericht eingeführt /ALGOL68/. Durch eine kontextfreie Grammatik  $G_{kf}$  kann man im allgemeinen nicht genau die Menge  $P$  einer in der Praxis verwendeten Programmiersprache, sondern nur eine Obermenge  $P_{kf}$  von  $P$  ( $P_{kf} \supseteq P$ ) darstellen. Die Einengung von  $P_{kf}$  auf  $P$  muß dann zusammen mit der Semantikfunktion erfolgen. Man beschreibt also an Stelle von  $(P, \sigma)$  ein  $(P_{kf}, \sigma_{kf})$ , wobei folgendes gilt:

$$P_{kf} \supseteq P,$$

$$\sigma_{kf} : P_{kf} \times E \rightarrow A \cup \{kp\}, \quad kp \notin A,$$

$$\sigma_{kf}(p, e) = \begin{cases} \sigma(p, e) & \text{für } p \in P \\ kp & \text{für } p \in P_{kf} \setminus P. \end{cases}$$

Die Semantikfunktion  $\sigma_{kf}$  ist für  $p \in P$  identisch mit  $\sigma$  und liefert für  $p \in P_{kf} \setminus P$  einen Wert  $kp$  (kein Programm), der nicht in  $A$  enthalten sein darf.

Zur Beschreibung der Semantik einer Programmiersprache muß die

Semantikfunktion  $\mathcal{G}$  bzw.  $\mathcal{G}_{\text{KF}}$  in einer geeigneten Weise dargestellt werden. In der Praxis wurden diese Funktionen bisher meist mit Hilfe der natürlichen Sprache und durch Beispiele erläutert. Als bekannteste Beschreibung dieser Art ist der ALGOL-60-Bericht zu nennen. Bei einer formalen Sprachdefinition muß natürlich auch die Semantik formal beschrieben werden.

### 2.3.2. Formale Beschreibung von Sprachverarbeitungssystemen

Zur formalen Beschreibung eines Verarbeitungssystems der Programmiersprache FS muß die Abbildung  $\varphi: S \times E \rightarrow A \cup D$  (siehe 2.2.) in einer geeigneten Sprache BS, die als Beschreibungs- oder Metasprache bezeichnet werden soll, dargestellt werden. Wie in 2.1. dargelegt, unterscheidet man bei Sprachen, also auch bei Metasprachen, zwischen Syntax und Semantik:  $BS = (F, \mathcal{G})$ . Ein Satz  $f \in F$  der Metasprache stellt eine Abbildung  $\varphi: S \times E \rightarrow A \cup D$  dar und beschreibt damit ein Sprachverarbeitungssystem mit der durch  $\varphi$  definierten Wirkung. Zur Ausführung aller in BS darstellbaren Verarbeitungssysteme wird ein universeller Algorithmus  $\alpha: F \times S \times S \rightarrow S$  für BS (universelles Sprachverarbeitungssystem, Verarbeitungssystem der Metasprache BS) benötigt. Wenn  $f \in F$  die formale Beschreibung eines Verarbeitungssystems der Wirkung  $\varphi: S \times E \rightarrow A \cup D$  sein soll, dann muß für  $p \in S$  und  $e \in E \subseteq S$   $\alpha(f, p, e) = \varphi(p, e)$  gelten. Die Verwendung von S bei  $\alpha$  an Stelle von E und  $A \cup D$  bei  $\varphi$  ist wegen der angestrebten Universalität von  $\alpha$  nötig. Wenn verschiedene Verarbeitungssysteme mit Hilfe von  $\alpha$  realisiert werden sollen, dann müssen die verschiedenen Eingabe-, Ausgabe- und Diagnosemengen berücksichtigt werden. Setzt sich ein Verarbeitungsprozess  $\varphi$  aus Einzelprozessen  $\varphi_i (i = 1, \dots, n)$  zusammen ( $\varphi = [\varphi_1, \dots, \varphi_n]$ ), so sollten diese Einzelprozesse in möglicherweise verschiedenen Sprachen  $BS_i = (F_i, \mathcal{G}_i)$  dargestellt werden. Für alle  $BS_i$  müssen universelle Verarbeitungssysteme  $\alpha_i$  vorhanden sein, für die gilt:

$$\alpha_i: F_i \times S \rightarrow S \text{ so, daß } \alpha_i(f_i, p) = \varphi_i(p), i = 1, \dots, n-1,$$

$$\alpha_n: F_n \times S \times S \rightarrow S, \text{ so daß } \alpha_n(f_n, p, e) = \varphi_n(p, e).$$

Die Darstellung von  $\alpha$  durch die  $\alpha_j$  erfolgt entsprechend der Darstellung von  $\varphi$  durch die  $\varphi_j$ .

$$\alpha(f, p, e) = \begin{cases} \alpha_j(f_j, \alpha_{j-1}(\dots, \alpha_1(f_1, p)\dots)), & \text{falls} \\ \alpha_j(f_j, \alpha_{j-1}(\dots, \alpha_1(f_1, p)\dots)) \in D \wedge j < n \\ \wedge \rightarrow \exists k < j : \alpha_k(f_k, \alpha_{k-1}(\dots, \alpha_1(f_1, p)\dots)) \in D \\ \alpha_n(f_n, \alpha_{n-1}(f_{n-1}, \dots, \alpha_1(f_1, p)\dots), e), & \text{sonst.} \end{cases}$$

**Bemerkung:**

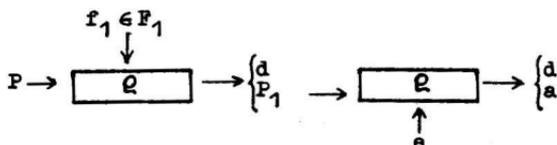
Es ist möglich, daß durch einen Satz  $f$  einer Metasprache  $BS_1$  in einem Fall ein Übersetzer beschrieben ist. Die entsprechende universelle Funktion ist dann zweistellig. In einem anderen Fall kann dagegen  $f'$  in derselben Sprache  $BS_1$  einen Interpreter beschreiben. Hier muß die entsprechende universelle Funktion dreistellig sein. Das Definitionsgebiet von  $\delta_1$  muß demnach die Vereinigung von  $(F_1 \times S)$  und  $(F_1 \times S \times S)$  sein. In den Beispielen wird immer nur der gerade benötigte Fall betrachtet. Im folgenden soll skizziert werden, wie man sich die Sprachverarbeitung bei den Beispielen (1) bis (3) aus 2.2. über die Beschreibung in Metasprachen vorstellen kann.

Zu Beispiel (1).

$BS_1$  : Maschinensprache MS;

$\alpha_1 = \varrho : MS \times S \rightarrow S$ ;

$BS_2$  : Die Sprache  $BS_2$  müßte die Beschreibung der Maschinensprache, also die Beschreibung der Hardwarefunktion, ermöglichen. Diese Aufgabe liegt am Rande des hier betrachteten Problembereiches.



Hier wird die Maschinensprache als Metasprache verwendet. Mit ihr sind aber keine Beschreibungen mit den in 1. geforderten Eigenschaften möglich.

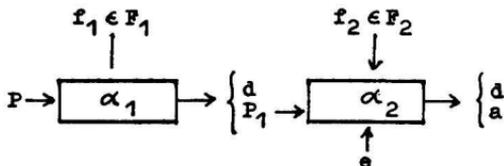
Zu Beispiel (2).

$BS_1$ : Sprache zur Beschreibung der Syntaxanalyse;

$\alpha_1$ : universeller Syntaxanalysealgorithmus;

$BS_2$ : Sprache, in der die Interpretation von strukturierten Programmen beschrieben werden kann;

$\alpha_2$ : Verarbeitungssystem für  $BS_2$ .



Zu Beispiel (3).

$BS_1$ :  
 $\alpha_1$ : } wie zu Beispiel (2);

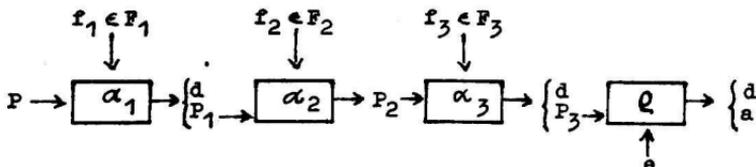
$BS_2$ : Sprache, in der Manipulationen mit dem strukturierten Programm gut ausdrückbar sind;

$\alpha_2$ : Verarbeitungssystem von  $BS_2$ ;

$BS_3$ : Sprache in der Übersetzungen von strukturierten Programmen in Maschinenprogramme oder in Programme beliebiger Zielsprachen dargestellt werden können;

$\alpha_3$ : Verarbeitungssystem für  $BS_3$ ;

$BS_4$ : siehe  $BS_2$  zu Beispiel (1).



Es wurde dabei angenommen, daß  $\varphi_2 : P_1 \rightarrow P_1$  nur eine Optimierung von  $p_1 \in P_1$  vornimmt und dabei keine Diagnoseaufgaben ausgeführt werden.

Betrachtet man die Beispiele (2) und (3), so stellt man fest, daß zwei Typen von Metasprachen benötigt werden. Zu den Sprachen des ersten Typs gehören  $BS_1$  von Beispiel (2) und Beispiel (3). In ihnen muß die Syntaxanalyse ausdrückbar sein. In den Sprachen des zweiten Typs, zu denen  $BS_2$  von Beispiel (2) und  $BS_2$  und  $BS_3$  von Beispiel (3) gehören, werden semantische Aspekte wie Interpretation, Optimierung und Übersetzung in den Beispielen dargestellt. Wenn zwei Sprachen gefunden werden können, von denen die eine die Beschreibung der Syntaxanalyse und die andere die Darstellung der verschiedenen semantischen Funktionen erlaubt, dann wäre damit die Voraussetzung für die Beschreibung weitgehend aller Sprachverarbeitungssysteme geschaffen. Auf dieser Annahme beruht unser Ansatz, der im Abschnitt 3. kurz vorgestellt wird.

### 3. Darstellung der vorgeschlagenen Beschreibungsmethode

Wir sind bei der von uns gewählten Beschreibungsmethode von der in 2.3.2. bereits formulierten Annahme ausgegangen, daß zwei Metasprachen für eine benutzerfreundliche Beschreibung der wesentlichen Sprachverarbeitungsprozesse ausreichen, eine für die syntaktischen und eine für die semantischen Aspekte. Entsprechend den Beispielen (2) und (3) im Abschnitt 2. sind damit Beschreibungen zweier grundverschiedener Typen von Sprachverarbeitungssystemen möglich. Die eine, entsprechend Beispiel (2), beschreibt im wesentlichen das äußere Verhalten des Verarbeitungssystems und kann als eine für den Benutzer geeignete Beschreibung der durch das System realisierten Programmiersprache angesehen werden. Die Beschreibung entsprechend Beispiel (3) stellt dagegen den realen Sprachverarbeitungsprozeß dar. Daß die Beschreibung eines Sprachverarbeitungsprozesses  $\Phi = [\Phi_1, \Phi_2]$  (entsprechend Beispiel (2)) als gleichwertig zu einer Sprachbeschreibung durch eine kontextfreie Regelgrammatik und die Beschreibung einer entsprechenden semantischen Funktion betrachtet werden kann, soll im folgenden erläutert werden:

$PS = (P, \sigma)$  sei die betrachtete Programmiersprache, die in der

Form  $(P_{kf}, \delta_{kf})$  beschrieben sei.  $\varphi = [\varphi_1, \varphi_2]$  sei das betrachtete Verarbeitungssystem für KS, wobei  $\varphi_1$  durch  $f_1 \in F_1$  und  $\varphi_2$  durch  $f_2 \in F_2$  beschrieben sein soll. Wenn wir davon ausgehen, daß  $\varphi_1 : S \rightarrow P_1 \cup D_1$  eine kontextfreie syntaktische Analyse darstellt, dann kann wegen der Entscheidbarkeit der kontextfreien Sprachen  $P_{kf}$  als durch  $\varphi_1$  und damit durch  $f_1$  definiert gelten:

$P_{kf} = \{p \mid p \in S \wedge \varphi_1(p) \in P_1\}$ . Die Semantikfunktion

$\delta_{kf} : P_{kf} \times E \rightarrow A \cup \{kp\}$  (siehe 2.3.1.) wird durch  $\varphi_1$  und  $\varphi_2$

durch  $\delta_{kf}(p, e) = \varphi'_2(\varphi_1(p), e)$  realisiert, wobei für

$\varphi'_2 : P_1 \times E \rightarrow A \cup \{kp\}$  gilt:

$$\varphi'_2(p_1, e) = \begin{cases} \varphi_2(p_1, e), & \text{wenn } \varphi_2(p_1, e) \in A \\ kp & , \text{ wenn } \varphi_2(p_1, e) \in D. \end{cases}$$

Der Unterschied besteht also lediglich darin, daß bei dem Verarbeitungssystem eine Diagnosemenge D zur näheren Fehlerspezifizierung verwendet wird, während es bei der Sprache einfach nur um die Entscheidung geht, ob ein Programm der Sprache vorliegt oder nicht.

Wir sind der Ansicht, daß die formale Beschreibung des äußeren Verhaltens eines Sprachverarbeitungssystems die Bedürfnisse der Sprachnutzer nach einer geeigneten Definition der Sprache weitgehend befriedigt, während die genauere strukturelle Beschreibung eines Verarbeitungssystem den Anforderungen, die bei Implementationen der Sprache auftreten, genügt. Mit zwei geeigneten Metasprachen zur Beschreibung der syntaktischen und semantischen Aspekte (gemäß den Ausführungen im Abschnitt 2.) und den entsprechenden Verarbeitungssystemen kann demnach der größte Teil der in 1. aufgeführten Einsatzgebiete für formale Beschreibungen von Programmiersprachen und Verarbeitungssystemen überdeckt werden.

Im folgenden sollen die von uns vorgeschlagenen zwei Metasprachen kurz charakterisiert werden.

In /Gru Me 75/ sind die Sprachen definiert und ausführlich erläutert und es ist die Anwendungsmethodik an Beispielen dargestellt.

### 3.1. Syntaxbeschreibungssprache

Die mit SYBS bezeichnete Syntaxbeschreibungssprache muß die Spezifizierung von Syntaxanalysealgorithmen erlauben. Die Entwicklung syntaxorientierter Compiler hat die Möglichkeit aufgezeigt, kontextfreie Regelgrammatiken zur Steuerung des Analyseprozesses zu verwenden.

Eine kontextfreie Regelgrammatik  $G_{kf} = (X, M, R, w)$  ist eine Regelgrammatik (siehe Abschnitt 2.), bei der für jede Regel  $(l, r) \in R$  gilt:  $l \in M, r \in (X \cup M)^*$ .

Zur Darstellung solcher Grammatiken wird bei Sprachdefinitionen häufig die Backus-Naur-Form (abgekürzt BNF) bzw. Erweiterungen davon verwendet. Dabei werden von der Grammatik nur die Regelmenge  $R$  und die Wurzel  $w$  explizit angegeben.  $X$  und  $M$  sind implizit in den Regeln gegeben /Mau 69/. In einer BNF-Regel werden alle Regeln der darzustellenden Grammatik  $G_{kf} = (X, M, R, w)$  mit gleichen ersten Komponenten zusammengefaßt.

In der Theorie kontextfreier Sprachen werden Grammatiken zur Definition der kontextfreien Sprachen folgendermaßen verwendet: Sei  $G_{kf} = (X, M, R, w)$  eine kontextfreie Grammatik. Dann ist  $P_{kf}$

die Menge der aus  $w$  ableitbaren Worte über  $X$ ,

$P_{kf} = \{p \mid p \in X^* \wedge w \Rightarrow p\}$  (siehe 2.1.). Wenn man die Backus-

Naur-Form als Metasprache in diesem Sinne verwendet, dann ist  $BNF' = (SY, \mathcal{J}')$  eine Programmiersprache, wobei  $SY$  die Menge der BNF-Darstellungen kontextfreier Grammatiken und  $\mathcal{J}': SY \rightarrow X^*$  ein Generierungsalgorithmus ist.

$\mathcal{J}'$  liefert für ein gegebenes  $g \in SY$  einen Satz  $p = \mathcal{J}'(g)$  der durch  $g$  definierten kontextfreien Sprache.

Wird eine Grammatik zur Steuerung des Syntaxanalyseprozesses verwendet, so muß die Backus-Naur-Form  $BNF'' = (SY, \mathcal{J}'' )$  als eine andere Sprache mit der gleichen Syntax aber einer anderen

Semantik  $\delta''$  angesehen werden.  $\delta'': SY \times S \rightarrow \{ja, nein\}$  ist ein Entscheidungsalgorithmus, der für gegebene  $g \in SY$  und  $p \in S$  entscheidet, ob  $p$  Element der durch  $g$  definierten kontextfreien Sprache ist oder nicht.

Soll durch die Analyse nicht nur entschieden werden, ob eine gegebene Zeichenkette  $p$  Satz einer Programmiersprache ist oder nicht, sondern ein strukturiertes Programm aufgebaut werden (siehe Beispiel (2) und (3) in 2.2.), dann muß eine Sprache BNF =  $(SY, \delta')$  mit  $\delta': SY \times S \rightarrow P_1 \cup D_1$  verwendet werden.

Die so aufgefaßte Backus-Naur-Form ist Ausgangspunkt und Kern der von uns geschaffenen Syntaxbeschreibungssprache SYBS. Die BNF wurde um verschiedene Ausdrucksmittel erweitert, die eine kürzere und besser überschaubare Beschreibung ermöglichen bzw. die Strukturierung des zu analysierenden Programms betreffen.

Es wurden Erweiterungen für folgende Aufgaben definiert:

- Faktorisierung ("Ausklammern", gemeinsamer Glieder in Alternativen), wahlweise mit Einschließung der leeren Alternative,
- Iteration zur Angabe von Wiederholstrukturen (wahlweise mit Angabe der minimalen, maximalen oder festen Schrittzahl),
- Ausschließung: Hierbei handelt es sich um ein Sprachelement zur Beschreibung solcher Zeichenreihenmengen, die sich am einfachsten als Differenz zweier Mengen darstellen lassen,
- Strukturierung des Programms durch explizite Angabe von Vorrangregeln und expliziten Strukturklammern.

### 3.2. Semantikbeschreibungssprache

Die Verarbeitungsprozesse, die mit Hilfe der Semantikbeschreibungssprache (abgekürzt: SEBS) formuliert werden sollen, gehen von dem durch die Syntaxanalyse erzeugten strukturierten Programm aus. Um Interpretationen und Übersetzungen ausdrücken zu

können, ist es zweckmäßig, in der Semantikbeschreibungssprache insbesondere Ausdrucksmittel für folgende Aufgaben bereitzustellen:

- Sukzessives Durchgehen des strukturierten Programms und Zuordnung entsprechender semantischer Aktionen,
- Erkennen von semantisch zusammengehörigen Teilen des strukturierten Programms,
- Aufbewahren von Attributen und Werten zur Behandlung der Kontextabhängigkeiten,
- Ausführung von Grundoperationen, die von der zu beschreibenden Sprache abhängig sind (z. B. arithmetische Grundoperationen bei Interpretationen von Sprachen),
- Aufbau eines Programms in einer beliebigen Zielsprache (für Übersetzung).

Zur Erledigung derartiger Aufgaben sind Operationen mit Zeichenketten und Listen notwendig. Zur Formulierung solcher nichtnumerischer Prozesse sind Listenverarbeitungssprachen geeignet, von denen die Sprache LISP /Leh 73/ am weitesten verbreitet ist.

Deshalb schien uns bei der Konzipierung der SEBS eine Orientierung an LISP als zweckmäßig. Wir haben eine der M-Sprache von LISP ähnliche Notationsform gewählt und als Erweiterungen solche Ausdrucksmittel geschaffen, die uns zur Beschreibung von Sprachverarbeitungsprozessen als notwendig oder sinnvoll erschienen. Die von anderen Programmiersprachen bekannten Standardausdrucksmittel wie Blockstruktur, Laufanweisungen, Sprunganweisungen und rekursiv aufrufbare Prozeduren sind in der SEBS enthalten.

Für die Syntaxbeschreibungssprache und die Semantikbeschreibungssprache wurden experimentelle Sprachverarbeitungssysteme auf der Grundlage eines LISP-Systems für ESER-Rechner entwickelt und angewendet.

## Literatur

- /ALGOL 60/ Revidierter Bericht über die algorithmische Sprache ALGOL 60; Akademie Verlag, Berlin, 1966
- /ALGOL 68/ Bericht über die algorithmische Sprache ALGOL 68; Akademie Verlag, Berlin, 1972
- /Gin 66/ Ginsburg, S.: The Mathematical Theory of Context free Languages; McGraw-Hill Book Company, 1966
- /Gru Me 75/ Grušl, J.; Merbeth, G.: Formale Beschreibung von Programmiersprachen und Sprachverarbeitungssystemen; Dissertation, TU Dresden, 1975
- /Her 61/ Hermes, H.: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit; Springer Verlag, 1961
- /Leh 73/ Lehmann, E.: LISP - eine Programmiersprache zur nichtnumerischen Informationsverarbeitung; Rechentechnik Datenverarbeitung, 1973
- /Mau 69/ Maurer, H.: Theoretische Grundlagen der Programmiersprachen - Theorie der Syntax; Hochschultaschenbücher, Bibl. Institut, Mannheim, Wien, Zürich, 404/404a, 1969

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

Manuskripteingang: 31. 5. 1976

### Anschriften der Verfasser:

Dr. Ing. Jürgen Grušl  
VEB ZFT Robotron  
801 Dresden,  
Leningrader Str. 15

Dr. Ing. Günter Merbeth  
IH Dresden  
8019 Dresden,  
Hans-Grundig-Str. 25

A method for construction of compilers and compiler-generating systems based on grammars of syntactical functions

0. Introduction

The structure of a compiler-generating system (CGS) and of compilers produced by it based on a kind of attribute grammars, the grammars of syntactical functions (GSF), is described. In Section I. we give a definition of GSF and some terms we need. The description of the structure of compilers resp. of CGS proposed in Section II. resp. Section III. is based on this terms.

For illustration we give some examples.

GSF were introduced in 1969 - 1972 in connection with the syntactical analysis of ALGOL 68-programs /Rie 72/. They are similar to the affix grammars /Ko 71/. They differ from these grammars in the application for the construction of compilers and, what is essential, in the use of attributes.

Differences in the form of grammars and in the use of attributes occur too between GSF and the attribute grammars in /Bo 74, 76/, /LRS 74/, /Lo 75/ and /Bor 75/.

In the following years the definition of GSF was revised. Principles of the construction of compilers by application of GSF are given in /Rie 761/. The description of a compiler from ASPLE /Uz 73/ to FORTRAN 63 based on this principles the reader can find in /Rie 762/.

I. Terminology

A GSF is a 10-tuple  $G = (F, B, A, S, V, C, R, ST, M, W)$  where

- F is a finit set of names of syntactical functions,
- B is a set of names of basic functions, B is a subset of F,
- A is a finit set of names of auxiliary syntactical functions,
- S is a finit set of names of semantic functions,
- V is a finit set of variables,
- C is a finit set of constants, V and C form together the set

of parameters P.

R is a finite set of syntactical rules of the form

$$f(P_1^f, \dots, P_{n_f}^f) : f_1(P_1^{f_1}, \dots, P_{n_{f_1}}^{f_1}), \dots, f_r(P_1^r, \dots, P_{n_r}^r), \\ a_1(P_1^{a_1}, \dots, P_{n_{a_1}}^{a_1}), \dots, a_s(P_1^s, \dots, P_{n_{a_s}}^s).$$

where  $f, f_1, \dots, f_r \in F, f \in F-B, a_1, \dots, a_s \in A \cup S,$

$P_k^f, P_1^{f_1}, P_m^{a_j} \in P (i=1, \dots, r; j=1, \dots, s; k=1, \dots, n_f;$

$l=1, \dots, n_{f_1}; m=1, \dots, n_{a_j}; r \geq 1, s \geq 0, n_{f_1}, n_{a_j} \geq 0).$

ST is the start symbol.

M is a mapping from F into the set of positive integers.

W is a warning symbol.

The sets V and C are disjoint. The sets F, A and S are likewise disjoint.

$(P_1^f, \dots, P_{n_f}^f)$  from  $f(P_1^f, \dots, P_{n_f}^f)$  where  $f \in F \cup A \cup S$  and  $n_f \geq 1$  is

a list of parameters.

$f(P_1^f, \dots, P_{n_f}^f)$  with  $f \in F$  resp.  $f \in A$  resp.  $f \in S$  is called

syntactical function resp. auxiliary syntactical function resp. semantic function.

For  $n_f = 0$  the list of parameters is deleted and a name of a syntactical function etc. is identical with a syntactical function etc.

Let  $f(P_1^f, \dots, P_{n_f}^f)$  be a syntactical function then M must

suffice the condition  $0 \leq M(f) \leq n_f$ .

Each function has a domain depending on the occurrence of the function in a syntactical rule. Variables in the list of parameters of a function may be replaced by values from the domain of the function. Constants, of course, are special values.

Replacing all variables of a syntactical function by values from the domains of the function we obtain a value of the syntactical function.

The value of an auxiliary syntactical function resp. semantic function is either the warning symbol  $W$  or the empty string  $\epsilon$  depending on the values of its parameters.

Let be  $f \in F-B$ . Then any value of the function  $f$  is called protonotion.

A protonotion  $N$  is a notion if a syntactical rule  $r \in R$  and a simultaneous replacing of all variables by values from domains of the functions in  $r$  exist so that the left side of  $r$  is identical  $N$  and no function with a name from  $A \cup S$  on the right side of  $r$  supplies  $W$ . The sequence of values of the syntactical functions on the right side of  $r$  is a direct production of  $N$ . The rule obtained from  $r$  by replacing of all variables by suitable values is called production rule.

A production of a notion  $N$  is either a direct production of  $N$  or the result of replacing the notions in a production of  $N$  by its direct productions.

A production which components are only values of basic functions is a terminal production.

Now it is possible to define a word as a terminal production of a notion supplied by some replacing of all variables of  $ST$ . The GSF  $G$  defines the language  $L(G)$  which is the set of all words.

Let  $w$  be a word from  $L(G)$ . Let  $ST(v_1, \dots, v_l)$  be a notion from which  $w$  is derived by a finite set of production rules. Let be  $M(ST)=m$ . Then  $v_m$  is called the semantic value of  $w$ .

The semantic value of a subword is defined analogously.

Remark: The separation of the sets  $A$  and  $S$  is not necessary, but it is useful from the practical point of view. Namely, auxiliary syntactical functions can be used for context analysis (for instance for examination of the following condition: For any applied occurrence of an identifier a defining occurrence of that identifier exists.) and semantic functions for generation of objectcode. It is also possible to apply semantic functions for interpretation.

**Example:**  $G = (F, B, A, S, V, C, R, ST, M, W)$

$F = \{ST, z, y, 0, 1\}$ ,  $B = \{0, 1\}$ ,  $A = \{NT\}$ ,  $S = \{NWO, NW1\}$ ,  
 $V = \{W1, W2, N1, N2\}$ ,  $C = \{0, 1\}$ ,

$M : M(ST) = 1, M(z) = 1, M(y) = 1$

The rules of R are

1  $ST(W1) : y(W1, N1)$ .

2  $z(W1, N1) : y(W1, N1)$ .

3  $z(W1, N1) : z(W2, N2), 0, NT(N1, N2), NWO(W1, W2)$ .

4  $y(W1, 0) : z(W2, N1), 1, NW1(W1, W2)$ .

5  $y(1, 0) : 1$ .

**NT:** The value of NT is  $\xi$  if the value of the second parameter is not equal two and the value of the first parameter is the value of the second parameter plus one.

Otherwise the value of NT is W.

**NWO:** If the value of the first parameter is the twofold of the value of the second parameter then the value of NWO is  $\xi$ . Otherwise the value of NWO is W.

**NW1:** If the value of the first parameter is the twofold of the value of the second parameter plus one then the value of NW1 is  $\xi$ . Otherwise the value of NW1 is W.

$z(W1, N1)$  is a syntactical function.  $NT(N1, N2)$  is an auxiliary syntactical function and  $NWO(W1, W2)$  is a semantic function.

$z(3, 0)$  is an instance of a value of  $z(W1, N1)$ .

$L(G)$  is a subset of the set of binary numbers, namely all words composed from 0 and 1 beginning and ending with 1 and not containing three consecutive zeros 0). The semantic value of such a number is a positive integer (here as a decimal number). For instance 1101 is from  $L(G)$ . Its semantic value is 13 for 1101 results from the application of the following production rules in this order (the production rule 1' is deduced from the syntactical rule 1 by suitable replacing of the variables by values):

1'  $ST(13) : y(13, 0)$ .

4'  $y(13, 0) : z(6, 1)$ . The value of  $NW1(13, 6)$  is  $\xi$ .

3'  $z(6, 1) : z(3, 0), 0$ . The value of  $NT(1, 0)$  resp.  
 $NWO(6, 3)$  is  $\xi$ .

- 2' z(3, 0): y(3, 0).  
 4' y(3, 0): z(1, 0), 1. The value of NW1(3, 1) is 6.  
 2' z(1, 0): y(1, 0).  
 5' y(1, 0): 1.

Remark: The parameters of ST, y and z have specific roles. So the value of the first parameter of ST resp. y and z is the semantic value of a word resp. subword. The second parameter of z and y is a counter for consecutive zeros (0).

## II. Compilers based on GSF

### Basic grammar and corresponding set

Before using a GSF for compiler construction it is separated in a basic grammar and a corresponding set (but the construction of a GSF is done, obviously, from a basic grammar and a corresponding set):

A basic grammar BG is a 4-tuple  $BG=(F, B, R', ST)$  where  $R'$  is deduced from  $R$  by omitting of all lists of parameters and of all functions with names from  $A \cup S$ .

For practical use a GSF should be defined so that BG is a reduced and unambiguous context-free grammar.

The corresponding set  $R''$  results from  $R$  by omitting of all elements from  $F$ .

By construction: For every rule from  $R'$  exactly one rule from  $R''$  exists.

Example: GSF G from Section I.;  $BG=(F, B, R', ST)$

- |          |  |    |          |
|----------|--|----|----------|
| R': 11   | ST: y.   | 41 | y: z, 1. |
| 21       | z: y.  | 51 | y: 1.    |
| 31       | z: z, 0.                                       |    |          |
| R'': 111 | (W1): (W1, N1).                                |    |          |
| 211      | (W1, N1): (W1, N1).                            |    |          |
| 311      | (W1, N1): (W2, N2), , NT(N1, N2), NWO(W1, W2). |    |          |
| 411      | (W1, 0): (W2, N1), , NW1(W1, W2).              |    |          |
| 511      | (1, 0): .                                      |    |          |

A rule from  $R''$  with number i1 corresponds to a rule from  $R'$  with number i.

### Structure of the compiler

The logical parts of the compiler are as usually

1. the lexical analysis,
2. the syntactical analysis (parsing),
3. the semantic analysis and code generation.

The lexical analysis is done in the usual manner.

The syntactical analysis is realized regarding to the basic grammar. It is possible to use any suitable method for context-free languages. The result of the syntactical analysis is a syntactical tree regarding to the basic grammar.

The semantic analysis and the code generation means the determination of the semantic value of the analysed word. That means: From nodes of the syntactical tree we must deduce notions, that is, from rules of  $R'$  applied for syntactical analysis we must deduce production rules.

Example: GSF  $G$  from Section I.,  $w=1101 \in L(G)$

1. The lexical analysis is omitted.
2. The result of the syntactical analysis regarding to  $BG$  delivers the tree in Fig. 1. (The numbers of edges are the numbers of the applied syntactical rules of  $BG$ .)
3. The result of the semantic analysis and of the code generation is shown in Fig. 2. (The numbers of edges give the numbers of the applied production rules. The production rule  $i'$  is deduced from the syntactical rule  $i$  from  $R$  of  $G$ .)  
 resulting code = semantic value = 13

The main problem at the code generation is the determination of the values of the parameters. Here we use a so called interpretation system for interpretation of the rules of the corresponding set and therefore for the determination of the values of the parameters. (The interpretation system does not belong to the definition of a GSF because it is not necessary for the definition of the semantic value. Whereas the definition of a semantic value enables to decide whether a given value may be a semantic value or not, an interpretation system enables the determination of the semantic value of a given word.) What rules of the corresponding set are interpreted for a given word is delivered from the syntactical analysis. Let



$i_0, i_1, \dots, i_n$  be the rules of the basic grammar BG used for the production of a word  $w$  from  $L(BG)$ . Then the rules  $i_0^1, i_1^1, \dots, i_n^1$  from the corresponding set are interpreted. The order of the interpretation depends on the method of the syntactical analysis and on the interpretation system.

**Example:** Interpretation system for the GSF  $G$  from Section I.

**Assumptions:** The first parameter of each function with a name from  $A \cup S$  is called output parameter. Each of the following parameters, if exist, is called input parameter. These functions are realized by suitable algorithms where the value of the output parameter is determined from the values of the input parameters. The algorithm for the realization of the function  $f(X_0, X_1, \dots, X_N)$  we choose so that the following is valid:

$A$  is a value of the output parameter for the values  $E_1, E_2, \dots, E_N$  of the input parameters iff the value of  $f(A, E_1, E_2, \dots, E_N)$  is  $\epsilon$ .

**Interpretation rules:**

1. If the same variable occurs in some lists of parameters of syntactical functions on both sides of a syntactical rule then the value of the variable on the left side is chosen equal the value of the variable on the right side.
2. If the same variable occurs on the left side of a syntactical rule and as output parameter of a function with a name from  $A \cup S$  then the value of the variable on the left side is chosen equal the value of the output parameter.
3. If the same variable occurs in a list of parameters of a syntactical function and as an input parameter of a function the name of which is from  $A \cup S$  then the value of the input parameter is chosen equal to the value of the variable in the list.

This interpretation system interprets the rules of the corresponding set  $R''$  of  $G$  from Section I. in the following manner:

1111 The value of the parameter of ST results directly from the value of the first parameter of  $y$ .

- 2111 The values of the first resp. second parameter of  $z$  result directly from the values of the first resp. the second parameter of  $y$ .
- 3111 a. The value of the input parameter of NWO results directly from the value of the first parameter of  $z$  on the right side.  
b. NWO (that is the corresponding algorithm) determines the value of the output parameter  $W1$ .  
c. The value of the first parameter of  $z$  on the left side results directly from the value of  $W1$ .  
d. The value of the input parameter of NT results directly from the value of the second parameter of  $z$  on the right side.  
e. The value of the output parameter  $N1$  of NT is determined.  
f. The value of the second parameter of  $z$  on the left side results directly from the value of  $N1$ .
- 4111 a. The value of the input parameter of  $NW1$  results directly from the value of the first parameter of  $z$ .  
b. The value of the output parameter  $W1$  of  $NW1$  is determined.  
c. The value of the first parameter of  $y$  results directly from the value of  $W1$ .  
d. The value of the second parameter of  $y$  is 0.
- 5111 The value of the first parameters of  $y$  is 1 and the value of the second parameter is 0.

Algorithms for the realization of the functions with names from  $A \cup S$ :

- NT( $N1, N2$ ): If the value of  $N2$  is two then an error is announced (the third consecutive zero was recognized) and the interpretation terminates. Otherwise the value of  $N1$  results from the value of  $N2$  plus one (increasing of the value of the zero counter by one).
- NWO( $W1, W2$ ): The value of  $W1$  results from two times the value of  $W2$ .

NW1(W1,W2): The value of W1 results from two times the value of W2 plus one.

The semantic value of 1101 at the beginning of this section was determined by this interpretation system. The numbers of rules (1', 4', 3', 2', 4', 2', 5') of the basic grammar applied for the production of 1101 were delivered by the syntactic analysis. Thus the rules of the corresponding set are known which must be interpreted.

Remarks: 1) The interpretation is syntax-directed.

2) The interpretation system of the example is very simple. For practical use more complicated systems are needed. Such systems use global parameters, references, the delay of the call of functions and other. An example of such a system can be found in /Rie 762/.

3) It is possible to realize the lexical analysis by a syntactical analysis with following code generation. The syntactical analysis examines the structure of symbols whereas the code generation is applied for instance for the construction of symbol tables etc.

The structure of a compiler based on a GSF may be seen from Fig. 3. Here the syntactical analysis is realized by a LR(k)-algorithm.

### III. Compiler-generating system

As was shown in Section II. all compilers based on GSF have the same structure. This enables their automatic construction by a CGS.

For practical reasons we suppose that all rules which describe the structure of a symbol form a separate set of rules. In the GSF these symbols are represent by basic functions. For instance a real number may be represent in a GSF by real number symbol (N). The value of N is either the value of a given real number or a pointer to a table of real numbers.

#### Structure

The CGS has three parts corresponding to the parts of a con-

structured compiler (see Fig. 4):

1. The constructor of the algorithm for the lexical analysis (COLEX): It constructs from the rules describing the structure of symbols the algorithm for the lexical analysis or parts of it.
2. The constructor of the parser (COFA) for basic grammar: First it constructs the basic grammar and the corresponding set from the rules of the GSF. Then from a library a parser (for instance a LR(k)-parser) is chosen and necessary resources (for instance LR(k)-tables) are prepared. Which algorithm is chosen may depend on a parameter or on the properties of the basic grammar.
3. The preparation of the code generation (PRECO): From a library an interpretation system is chosen depending on the value of a parameter and modified for interpretation of the given corresponding set.

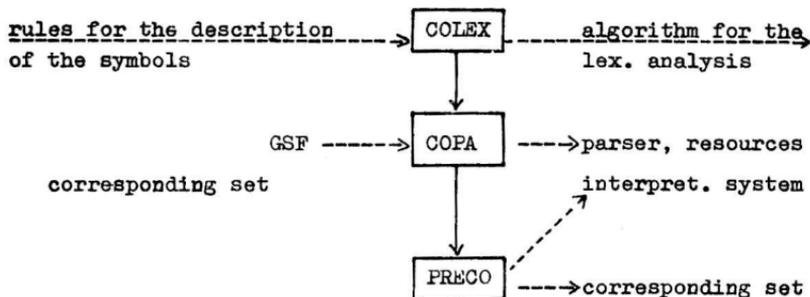


Fig. 4

#### IV. Final remarks

Here we give some advantages of compilers constructed by a CGS based on GSF:

1. The dependence of the compiler on the object language appears only in the algorithms for the realization of the semantic functions. Therefore a change of the object language implies an exchange of these algorithms.
2. The compiler has a simple structure. The correspondence between syntactical constructions and semantic values is transparent.  
The principle of orthogonality of semantic is realizable.
3. The expanding or changing of GSF does not imply changes of the compiler as a whole.
4. The application of GSF for the description of languages enables a clear representation of the language. The designer of the language needs no special knowledge of the compiler construction.
5. For easier construction of algorithms for the realization of auxiliary syntactical functions and semantic functions a library with, for instance, algorithms for list processing etc. could be constructed. Also it is possible to formalize the description of the algorithms for the realization of those functions.
6. The application of the compiler does not require computers with a memory of great size because of the separation of auxiliary syntactical functions and semantic functions.
7. GSF are suitable for the description of the analysis and the necessary program.

#### Summary

The structure of a compiler-generating system and of the compilers produced by it based on a kind of attribute grammars, the grammars of syntactical functions, is described. Grammars of syntactical functions differ from other attribute

grammars in form and use.

They may be constructed from context-free grammars by adding of parameters to the syntactical elements (the result are syntactical functions) and by embedding of so called auxiliary syntactical functions and semantic functions.

The parameters are evaluated by interpretation systems.

### References

- /Bo 74/ Bochmann, G. V., Attribute grammars and compilation,  
Document de travail 54
- /Bo 76/ Bochmann, G. V., Semantic Evaluation from Left to  
Right, Communications of the ACM  
(1976), vol. 19, No. 2, Febr.  
pp. 55 - 62
- /Bor 75/ Borowiec, J., Pragmatics in a compiler producing  
system,  
Proceed. of the Conf. Impl. of Alg.  
Lang., Warsaw 1975
- /Ko 71/ Koster, C. H. A., Affix grammars,  
Mathematisch Centrum Amsterdam, 1971
- /LRS 74/ Lewis, P. M., Rosenkrantz, D. J., Stearns, R. E.,  
Attributed translations,  
J. of Comp. and Syst. Sc.,  
vol. 9, No. 3, Dec. 1974
- /Lo 75/ Lorho, B., Semantic attributes processing in  
the system DELTA,  
Le Chesnay, 1975
- /Rie 72/ Riedewald, G., Syntaktische Analyse von ALGOL 68-  
Programmen,  
Thesis, Rostock 1972

- /Rie 761/ Riedewald, G., Gramatiky syntaktických funkcí a jejich používání v kompilátorech, acta polytechnica IV /1976/, 2, práce ČVUT v Praze, 6, pp. 89 - 103
- /Rie 762/ Riedewald, G., Compilermodell auf der Grundlage einer Grammatik syntaktischer Funktionen, FG Programmiersprachen der Sektion Mathematik der Wilhelm-Pieck-Universität Rostock 1976
- /U 73/ Uzgalis, R., What every programmer should know about grammars, Research supported by NSF, GJ 809

eingegangen: 03. 10. 1977

Anschrift des Verfassers:

Dr. rer. nat. Günter Riedewald  
Wilhelm-Pieck-Universität Rostock  
Sektion Mathematik  
DDR-25 Rostock  
Universitätsplatz 1

## Parallele Bearbeitung von Prozessen

### 1. Einleitung

Die Idee der parallelen Informationsverarbeitung auf einem Rechnersystem mit mehreren Prozessoren kam in den 50er Jahren auf.

Wichtig war dabei, daß die Resultate der Arbeit eines Prozessors von allen anderen Prozessoren zur weiteren Bearbeitung benutzt werden können.

Die Programmierung von Aufgaben für solche mehrprozessorigen Systeme erhielt die Bezeichnung "parallele Programmierung". Die Entwicklung der parallelen Programmierung begann in der ersten Hälfte der 60er Jahre, wobei sich bald vier Hauptrichtungen herauskristalisierten:

1. Schaffung von Methoden und Sprachen für die parallele Programmierung.
2. Schaffung von Algorithmen zur Umgestaltung sequentieller Programme in parallele und umgekehrt.
3. Prognosen über die zu erwartende Zeit der Aufgabenlösung und über die erforderliche Anzahl von Prozessoren.
4. Aufstellen von formalen Modellen paralleler Programme zum Bau von Systemen.

Durch die parallele Programmierung kommen völlig neue Probleme auf die Compilerbauer und die Anwender zu. Eines der wichtigsten ist dabei das Problem der Identität von Programmen.

Definition: (vgl. /Kot74/)

Ein paralleles Programm heißt identisch, wenn es auf einem mehrprozessorigen System bei jeder Ausführung mit gleichen Eingangsdaten das gleiche Ergebnis liefert.

Der Berechnungsprozeß selber kann sich auf mehrprozessorigen Systemen ändern, da er von der Anzahl der zugänglichen Pro-

zessoren und von der "Reihenfolge" der Abarbeitung der parallelen Zweige abhängig ist, nur das Ergebnis soll immer das gleiche sein.

Ein weiteres Problem der parallelen Programmierung ist das Erkennen von "kritischen Abschnitten" innerhalb paralleler Programme.

Definition:

Ein Abschnitt heißt kritisch, wenn er von mehreren Prozessen durchlaufen wird, aber zu einem festen Zeitpunkt nur von einem Prozeß durchlaufen werden darf.

Es ist also eine Aufeinanderabstimmung (Synchronisation) der Prozesse notwendig.

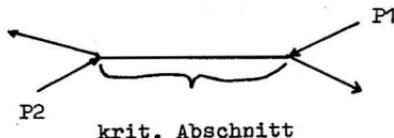


Abb. 1

2. Beschreibung zeitlicher Parallelität durch Petrinetze

Petrinetze bilden eine Erweiterung der Flußdiagramme. Ein Flußdiagramm (spezielles Netz) besitzt eine aktive Spinne, die in dem Netz einen vorgegebenen Weg entlangläuft und sich an Knotenpunkten für einen Zweig "entscheiden" muß. Petrinetze können mehrere aktive Spinnen besitzen. Dabei kann es auch passieren, daß sie gleiche Wege gehen wollen (kritische Abschnitte). Es ist also eine Synchronisation notwendig.

Definition:

Ein Petrinetz ist ein gerichteter Graph mit zwei verschiedenen Typen von Knoten, genannt Stellen S (durch  $\bigcirc$  dargestellt) und Transitionen T (durch  $|$  dargestellt). Jede Kante verbindet entweder eine Stelle mit einer Transition oder eine Transition mit einer Stelle.

Die kleinsten Petrinetze sind demnach:  $\bigcirc \longrightarrow |$  und  $| \longrightarrow \bigcirc$  .

Ein Petrinetz beschreibt einen Automaten, wobei die Menge der Zustände den Stellen und die Menge der Zustandsänderungen den Transitionen entspricht. Die Stellen in Petrinetzen können markiert werden  $\odot$  , wodurch Automaten simuliert werden können.

Definition:

Eine Transition heißt aktiviert, wenn sämtliche Stellen, von denen Kanten zu den Transitionen laufen (Eingabestellen), markiert sind.

Eine aktivierte Transition kann zum "Schalten" ausgewählt werden, d. h. die Markierungen der Eingabestellen werden gelöscht und die Ausgabestellen werden markiert.

Beispiel:



Im folgenden werden einige Beispiele zur Darstellung einfacher Programmabläufe durch Petrinetze gegeben. Dabei entsprechen die Transitionen den Anweisungen in einem Programm, und die Stellen werden als Anweisungsmarkierer (Programmzeiger) gebraucht. Es ist immer diejenige Stelle markiert, die vor der nächsten ausführbaren Anweisung steht.

- lineare Programme

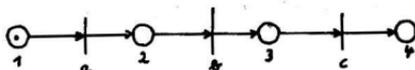


Abb. 2

Die Stelle 1 ist markiert, d. h. die Transition a kann geschaltet werden (Anweisung a kann ausgeführt werden), usw.

Das Petrinetz entspricht also dem Programm

begin a; b; c end

- verzweigte Programme

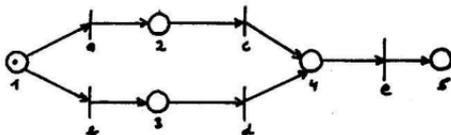


Abb. 3

Die Transitionen a und b sind aktiviert. Die Auswahl einer Transition zum Schalten wird durch eine hier nicht dargestellte programmierte Bedingung (Zustand von Stelle 1) getroffen.

begin if 1 then a; c else b; d fi; e end

- Unterprogrammtechnik

Das Petrinetz

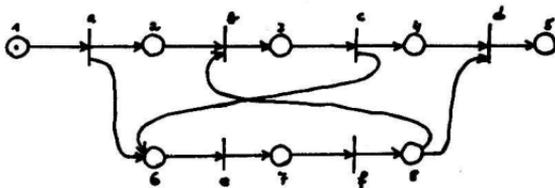


Abb. 4

entspricht der Anweisungsfolge

begin proc upro = void:(e; f);

a; upro; b; c; upro; d

end

wobei die Stellen 2 und 4 die Rückkehradressen im aufrufenden Programm kennzeichnen.

- parallele Prozesse

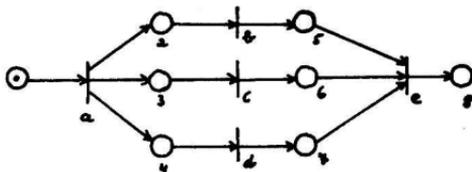


Abb. 5

Die Transition a entspricht dem Öffnen einer parallelen Klausel (par begin) und die Transition e entspricht dem Schließen

(end).

par begin b, c, d end

- Semaphore (siehe auch Abschnitt 3)

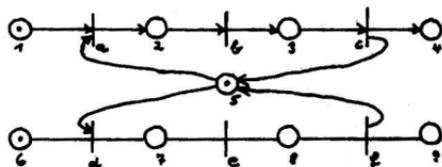


Abb. 6

Die Stelle 5 hat die Funktion eines Semaphors.

Die Transitionen a und d entsprechen der Anweisung down sema.

Die Transitionen c und f entsprechen der Anweisung up sema.

begin sema sema = level 1;

par begin (down sema; b; up sema)

(down sema; e; up sema) end

end

Durch die Möglichkeit der Darstellung paralleler Prozesse mit Hilfe von Petrinetzen lassen sich leicht kritische Abschnitte erkennen. Es werden also diejenigen Abschnitte der Prozesse erkannt, die synchronisiert werden müssen um Systemverklebungen (deadlock) zu vermeiden.

### 3. Sprachen und Methoden der parallelen Programmierung

Für die parallele Programmierung gibt es folgende Grundforderungen an eine Sprache (vgl. /Kot74/):

1. Sie soll die Möglichkeit geben, den natürlichen Parallelismus auszudrücken.
  2. Sie soll systemunabhängig sein, d. h. man braucht keine Systemparameter für die Anzahl der benötigten Prozessoren, für die Abarbeitungszeit einzelner Programnteile u. a. angeben.
  3. Die Steuerung und Kontrolle der Abarbeitung des parallelen Programms in der Sprache soll automatisch und einfach sein.
- Im folgenden werden drei Programmiersprachen vorgestellt, die

eine parallele Bearbeitung von Prozessen simulieren, bzw. die Konstruktionen für echte parallele Programmierung enthalten.

### SIMULA

SIMULA wurde als Erweiterung von ALGOL 60 in den Jahren 1965 - 1967 von O. J. Dahl, B. Myhrhaug und K. Nygaard am Norwegian Computing Center in Oslo entwickelt. Ursprünglich war sie nur für Simulationszwecke gedacht. Es stellte sich aber bald heraus, daß der Anwendungsbereich der Sprache mit denen von ALGOL 68 und PL/1 vergleichbar ist. Durch das Koroutinenkonzept wird die Simulation diskreter Modelle, d. h. Modelle, die durch den Ablauf von Prozessen bestimmt sind, ermöglicht. Solche Modelle bestehen aus einem System von  $m+1$  Koroutinen  $k(1), \dots, k(m), v$  mit folgenden Eigenschaften:

1.  $m$  ist die Anzahl der ablaufenden Prozesse.
2. Nur die Koroutine  $v$  aktiviert eine Koroutine  $k(i)$  und jede aktivierte Koroutine  $k(i)$  gibt die Regie an  $v$  zurück ( $1 \leq i \leq m$ ).
3. Eine Koroutine  $k(i)$  kann bestimmen, welche Koroutine danach von  $v$  aktiviert werden soll.

Die Simulation erfolgt durch die Standardklasse "simulation", die eine Klasse "process", die Prozeduren "passivate", "wait(s)", "cancel(p)" und "hold(t)" sowie die Simulationsanweisungen ACTIVATE und REACTIVATE enthält. (Beispiele siehe in /Roh73/).

### SIMONE

SIMONE ist eine Programmiersprache, die 1975 an der Queen's University Belfast auf der Grundlage von PASCAL geschaffen wurde und die die Simulationsmöglichkeiten von SIMULA ohne Klassen realisiert. (Siehe /Kau76/).

Dafür wurden drei neue Sprachelemente eingeführt:

- Prozesse (Teile des Programms, die parallel ausgeführt werden können)

```
PROCESS prozeßname (Parameterliste);  
    Deklaration lokaler Variablen und Prozeduren;  
    BEGIN Aktionen END;
```

Die Aktivierung des Prozesses erfolgt durch einen Aufruf.

- Monitore (Zusammenfassung aller Variablen und Prozeduren, die von verschiedenen Prozessen benutzt werden)

MONITOR monitorname;

Deklaration lokaler Daten und Prozeduren;

BEGIN Initialisierung der lokalen Daten END;

Sie dienen zur Kontrolle der Wechselbeziehungen zwischen den Prozessen.

- Durch Bedingungsvariablen kann eine Warteschlange für die Prozesse gebildet werden.

Bed.-var.: CONDITION;

Eine Prozedur im Monitor kann dann folgende Operationen ausführen:

Bed.-var.. WAIT(p);

Der Prozeß, in dem diese Operation vorkommt, wird in eine Warteschlange mit der Priorität p eingeordnet.

(ohne p-Angabe, an das Ende der Warteschlange)

Bed.-var.. SIGNAL;

Der Prozeß mit der höchsten Priorität wird aktiviert.

#### Beispiel: Philosophenproblem

Fünf Philosophen sitzen die ganze Zeit ihres Lebens an einem runden Tisch, an dem sie miteinander philosophieren. Zwischen- durch essen sie unabhängig voneinander. Jeder Philosoph hat an dem Tisch einen festen Platz und zwischen je zwei benachbarten Plätzen liegt eine Gabel.

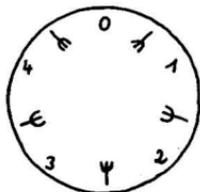


Abb. 7

Zum Essen benötigt jeder Philosoph die Gabeln links und rechts von seinem Platz. Er muß sich also vergewissern, ob beide Gabeln frei sind. Danach ergreift er sie, nimmt seine Mahlzeit

ein und gibt sie wieder frei, wenn er damit fertig ist. Da nicht alle gleichzeitig essen können, ist eine Synchronisation notwendig.

```
MONITOR FORKS;
VAR FORK: ARRAY [0..4] OF 0..2;
    READY: ARRAY [0..4] OF CONDITION;
    I: INTEGER;
PROCEDURE PICKUP(I:INTEGER); /*Aufnahme der Gabeln*/
BEGIN IF FORK[I] #2 THEN /*Beide Gabeln sind nicht frei, warten*/
        READY(I).WAIT;
        FORK[(I-1)MOD 5]:=FORK[(I-1)MOD 5]-1;
        FORK[(I+1)MOD 5]:=FORK[(I+1)MOD 5]-1
        /*Gabeln werden für beide Nachbarn gesperrt*/
END;
PROCEDURE PUTDOWN(I:INTEGER); /*Ablegen der Gabeln*/
BEGIN FORK[(I-1)MOD 5]:=FORK[(I-1)MOD 5]+1;
        FORK[(I+1)MOD 5]:=FORK[(I+1)MOD 5]+1;
        /*Beide Gabeln werden für die Nachbarn frei gegeben*/
        IF FORK[(I-1)MOD 5]=2 THEN READY[(I-1)MOD 5].SIGNAL;
        IF FORK[(I+1)MOD 5]=2 THEN READY[(I+1)MOD 5].SIGNAL
        /*Nachbarn von Philosoph I könnten essen, wenn sie auch die zweite Gabel zur Verfügung haben*/
END;
BEGIN FOR I:=0 TO 4 DO FORK[I]:=2
        /*Monitorinitialisierung,
        FORK[I]=2, Phil. I hat beide Gabeln zur Verfügung
        FORK[I]=1, Phil. I hat nur eine Gabel zur Verfügung
        FORK[I]=0, Phil. I hat keine Gabel zur Verfügung*/
END;
PROCESS TIMER;
BEGIN HOLD(LIFETIME); /*Lebenszeit des Phil. = Simulationszeit*/
        TERMINATE
END;
PROCESS PHIL(I:INTEGER);
BEGIN WHILE TRUE DO
```

```

BEGIN FORKS.PICKUP(1); /*Phil. I ist hungrig, will Gabeln
                        aufnehmen*/
    HOLD(RANDOM NUMBER); /*Essenperiode*/
    FORKS.PUTDOWN(1); /*Phil. I ist satt*/
    HOLD(RANDOM NUMBER); /*Denkperiode*/
END
END;
BEGIN TIMER; PHIL(0); PHIL(1); PHIL(2); PHIL(3); PHIL(4) END;

```

### ALGOL 68

In ALGOL 68 gibt es zwei Konstruktionen für die parallele Programmierung: - kollaterale Klausel,  
- parallele Klausel.

Während die "Reihenfolge" der Abarbeitung bei der kollateralen Klausel vom Programmierer nicht zu beeinflussen ist, kann bei der parallelen Klausel durch die von E. W. Dijkstra geschaffene Semaphortechnik (siehe /Dij68/) eine Steuerung der Abarbeitungsfolge der Einzelklauseln vorgenommen werden. (Beachtung kritischer Abschnitte, Synchronisation möglich).

ALGOL 68 ordnet den Semaphoren einen standardmäßig definierten Modus sema zu:

```
mode sema = struct(ref int ?);
```

Der Programmierer hat zu dem Steuerwert ? keinen direkten Zugriff. Er kann ihn nur durch die Operatoren level, up und down beeinflussen.

Durch level kann man einem Semaphor einen Steuerwert zuweisen:

```
sema signal = level 1;
```

oder den Steuerwert als Integer-Zahl aufbewahren:

```
i := level signal;
```

Der Operator down erniedrigt den Steuerwert um 1, wenn er größer oder gleich 1 ist. Sonst wird der Prozeß, in dem dieser Operator angewandt wird, gestoppt und in eine dem Semaphor zugeordnete Warteschlange eingeordnet.

Der Operator up erhöht den Steuerwert um 1, wenn in der zugehörigen Warteschlange keine Prozesse warten. Sonst wird einer

der wartenden Prozesse fortgesetzt. (Spezielle Festlegungen dazu sind vom Betriebssystem abhängig.)

Beispiel:

Es sollen mehrere Werte in einer Zeile gedruckt werden. Diese Werte werden in einer Struktur zusammengefaßt und in einen Puffer gefüllt. Sobald in dem Puffer Strukturen stehen, können diese parallel zur Pufferfüllung ausgegeben werden.

(siehe /Lin77/)

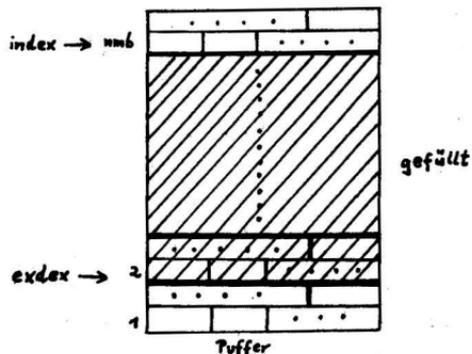


Abb.8

Das Programm arbeitet mit zwei Zeigern. Während der Zeiger exdex auf die nächste auszugebende Struktur zeigt, gibt der Zeiger index den ersten freien Speicherplatz im Puffer an. Man erkennt sofort, daß es zwei kritische Ereignisse gibt:

- der Puffer ist leer (keine Ausgabe möglich),
- der Puffer ist voll (keine Pufferfüllung möglich).

In beiden Fällen ist eine Synchronisation zwischen Ausgabe und Pufferfüllung notwendig. Dazu wird die Semaphortechnik angewandt;

```

begin mode item = struct (c Sammlung von Werten, die in einer
                          Zeile gedruckt werden sollen c);
proc item generate = (c Berechnung der nächsten Ausgabe-
                     zeile in bel. Zeit) c);
int nmb = c Anzahl der Strukturen, die der Puffer aufnehmen kann c;
[1:nmb]item buffer;
  
```

```

int index:=0, exdex:=0;
bool work to be done:=true, printing to be done:=true;
sema free slots = level nmb, full slots = level 0;
  c Puffer ist leer c
par (c Pufferfüllung c
  while work to be done
  do down free slots; index modab nmb +=1;
    buffer [index] :=generate;
    if c keine Zeilen zu berechnen c
    then work to be done := false fi;
    up full slots
  od,
  c Ausgabe c
  while printing to be done
  do down full slots; exdex modab nmb +=1;
    print(buffer[exdex]);
    printing to be done := work to be done ✓
    index ≠ exdex;
    up free slots
  od)
end

```

### Literatur

- /Dij68/ Dijkstra, E. W., Cooperating Sequential Processes  
in: F. Genuys (ed.): Programming  
Languages Academic Press,  
London/New York 1968
- /Kau76/ Kaubisch, W. H., Perrot, R. H., Hoare, C. A. R.,  
Quasiparallel Programming  
Software - Practice and Experiments;  
Vol. 6(1976)

/Kot74/ Kotov, W. E., Teorija paralelnowo programmirowanija-prikladnye aspekty Kibernetika, No. 1, Kiew 1974

/Lin77/ Lindsey, C. H., van der Meulen, S. G., Informal Introduction to ALGOL 68 North-Holland Publishing Company London - New York - Oxford 1977

/Roh73/ Rohlfing, H., SIMULA-Eine Einführung Bibliographisches Institut, Mannheim 1973

/Son74/ Sonnenberg, W., Zur Beschreibung der Ablaufstruktur von Programmen mit Petrinetzen Internbericht 10/74, Institut für Informatik der Universität Karlsruhe 1974

Vortrag zum Problemseminar "Programmiersprachen und Implementierung" der TU Dresden in Weißig, November 1977

eingegangen 14. 12. 1977

Anschrift des Verfassers:

Dipl.-Math. Reinhard Deth  
Wilhelm-Pieck-Universität Rostock  
Sektion Mathematik  
DDR-25 Rostock  
Universitätsplatz 1

A n m e r k u n g e n

# Anmerkungen



