ALGEBRA UND
UMGEBUNGSUNABHÄNGIGE SPRACHEN

Oskar ITZINGER

Forschungsbericht Nr.65

Februar 1972

INHALT

Seite

V o r w o r t

Dieses Paper ist gedacht einerseits als Zusammenstellung der grundsätzlichen Begriffe der algebraischen Linguistik, andererseits als Einführung in einige der hauptsächlichsten Forschungsbereiche.

Zu diesem Zweck werden die wichtigsten Hilfsmittel aus der Algebra an den jeweils relevanten Stellen entwickelt, da nicht vorausgesetzt werden kann, daß der überwiegend statistisch-empirisch arbeitende Verhaltenswissenschaftler mit jenen vertraut ist, in der einschlägigen Fachliteratur diese Kenntnisse aber vorausgesetzt sind.

Es ist hier noch eine Bemerkung bezüglich der Notation von Mengen, Funktionen etc. anzuschließen. Häufig verwenden Fachpublikationen - obwohl weit von einer einheitlichen Bezeichnungsweise entfernt - griechische Buchstaben. In diesem Paper werden aus schreibtechnischen Gründen nur deutsche Buchstaben verwendet. Eine Zuordnung zur Literatur erfolgt dann am besten über die gegebenen Definitionen und Sätze.

Wien, Februar 1972 Oskar Itzinger

ALGEBRA UND UMGEBUNGSUNABHÄNGIGE SPRACHEN

I. EINLEITUNG

Der vorliegende Aufsatz behandelt gewisse Gemeinsamkeiten natürlicher und künstlicher Sprachen. Die Art der Betrachtungsweise fällt dabei in den Bereich der mathematischen Linguistik, den man zum Unterschied von statistischer Linguistik (welche sich mit statistischen Eigenschaften wie Auftrittshäufigkeit gewisser Buchstaben, informationstheoretischen Gesichtspunkten und verwandten Gebieten auseinandersetzt) als algebraische Linguistik bezeichnet.

Algebraische Linguistik untersucht die formalen Gegebenheiten von natürlichen Sprachen (im weiteren Sinne auch von künstlichen Sprachen wie Programmiersprachen), um dadurch Aufschlüsse über die Struktur der jeweiligen Sprache zu erhalten. Zu bemerken ist allerdings, daß es sich jedenfalls um einen Abstraktionsprozeß handelt.

Damit die weitere Vorgangsweise verständlich wird, sind zunächst einige grundsätzliche Begriffe zu klären, die bei der algebraischen Untersuchungsweise eine fundamentale Rolle spielen.

Unter einer Sprache wollen wir eine Menge von sogenannten "Ketten" über einer endlichen Menge von Symbolen, dem Vokabular, verstehen.

Als Grammatik bezeichnen wir eine Menge von Regeln, die es gestatten, die zur Sprache gehörenden Ketten rekursiv aufzuzählen. Diese Grammatik erzeugt (generates) dann die Ketten der betrachteten Sprache.

In natürlichen Sprachen findet sich ein Analogon zu diesen Bezeichnungen in folgendem Beispiel:

Kette	Satz der Sprache
Kettenglied	Wort der Sprache
Vokabular	Alphabet der Sprache

Für künstliche Sprachen, also beispielsweise für Programmiersprachen, läßt sich folgende Zuordnung treffen:

Kette	Programm der Programmiersprache
Kettenglied	Einzelnes Programmstatement
Vokabular	Zeichenvorrat der Programmiersprache

In beiden Beispielen besteht die Grammatik aus denjenigen Regeln, die angeben, wie Sätze (oder Programme) gebildet werden. Die Bildung der Kettenglieder, über die die Grammatik ja keine Aussagen macht, wird später zu beschreiben sein.

Damit eine Klasse von Grammatiken linguistisches Interesse verdient, muß es eine Prozedur geben, die jedem Paar (s, G) - wobei s eine Kette und G eine Grammatik aus dieser Klasse sind - eine hinreichende Strukturbeschreibung der Kette s in Hinblick auf die Grammatik G zuordnet; die Strukturbeschreibung muß daher speziell andeuten, ob s eine wohlgeformte Kette der von der Grammatik G erzeugten Sprache L (G) ist (wo dies immer der Fall ist). Ist s keine wohlgeformte Kette, dann soll die Strukturbeschreibung angeben, in welcher Hinsicht s von der Wohlgeformtheit abweicht.

Uns interessiert nur eine Eigenschaft der Strukturbeschreibung, nämlich ihre Unterteilung einer gegebenen Kette in verschiedene Kategorien.

An einem Beispiel aus der deutschen Sprache läßt sich dies leicht demonstrieren. Bekanntlich sind in dieser solche Kategorien gegeben durch "Nomen","Verb", "Nominalphrase", "Verbalphrase","Präposition", "Artikel","Adjektiv", "Satz", "Adverb".

Mit diesen Begriffen zerlegen wir jetzt den Satz

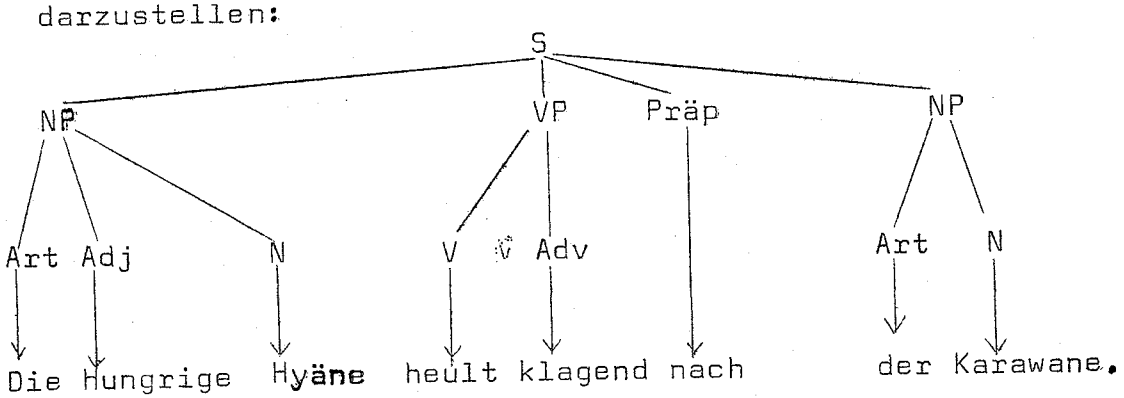
"Die hungrige Hyäne heult klagend nach der Karawane".

Wir bedienen uns dazu der in der Linguistik üblichen indizierten Klammerung ("labelled bracketing"):

(1) [S [NP [Art Die] [Adj hungrige] [N Hyäne]] [VP [V heult] [Adv klagend]] [Präp nach] [NP [Art der] [N Karawane]]]

Es ist naheliegend, diese Struktur durch einen Baum darzustellen:

(2)



In (2) haben wir - abgesehen von den Strichführungen - zwei Symbolklassen konstruiert:

a) S, NP, Art, Adj, N, VP, V, Adv, Präp,

b) der erzeugte Satz

Man bezeichnet die Symbole von (a) als Hilfssymbole ("non-terminals"), während die Symbole von (b) als Endsymbole ("terminals") bekannt sind.

Die Regelmenge, die uns schließlich den Satz lieferte, kann leicht so geschrieben werden:

(3)
S ⟶ NP VP PRÄP NP
NP ⟶ Art Adj N
NP ⟶ Art N
Art ⟶ die
Art ⟶ der
Adj ⟶ hungrige
N ⟶ Hyäne
N ⟶ Karawane
VP ⟶ V Adv
V ⟶ heult
Adv ⟶ klagend
PRÄP ⟶ nach

Die Regelmenge (3) bezeichnet man als Produktionsregeln oder Produktionen; man sagt, daß der Satz mit diesen Produktionen abgeleitet wurde.
Eine Grammatik G ist dann gegeben durch ein Quadrupel
G = (A,B,P,S).

Es bedeuten:

1) A ist eine nichtleere Menge von metalinguistischen Variablen ("Hilfssymbolen")

2) B ist eine nichtleere Menge von Basiszeichen ("Endsymbolen")

3) P ist eine **nichtleere** Regelmenge ("Produktionen")

4) S ist ein ausgezeichnetes Element aus A.

Ferner soll gelten $A \cap B = \emptyset$

Die Vereinigung von A und B bezeichnet man auch als Gesamtalphabet G_1, sodaß eine Grammatik kürzer schreibbar ist als

$$G = (G_1, P, S).$$

Bei der Betrachtung des Systems (3) fällt auf, daß auf der linken Seite des Pfeils jeweils genau ein Hilfssymbol steht. Grammatiken, deren Produktionen diese Eigenschaften haben, bezeichnet man seit CHOMSKY als umgebungsunabhängige ("context-free", CF) Grammatiken. Eine Menge von Endsymbolen, die durch eine solche Grammatik erzeugt wird, bezeichnet man als umgebungsunabhängige Sprache. Nebenbei sei bemerkt, daß die Verletzung der Forderung nach "links vom Pfeil genau ein Hilfssymbol" natürlich zu anderen Grammatiken führt (zum Beispiel zu umgebungsabhängigen - "context-sensitive"-Grammatiken).

Die Untersuchung des Systems (3)zeigt aber auch, daß der abgeleitete Satz nicht der einzig mögliche ist. Genauso kann ja der Satz " Die hungrige Karawane heult klagend nach der Hyäne" abgeleitet werden. Uns interessiert hier nicht so sehr die semantische Unsinnigkeit (zumindest im allgemeinen) dieses Satzes, sondern die Eigenschaft des Systems (3), welche diesen Satz liefert. Diese Eigenschaft wird als Mehrdeutigkeit (ambiguity) bezeichnet.

Diese Mehrdeutigkeit ist mit ein Grund, daß CF-Grammatiken nicht hinreichend sind für natürliche Sprachen. Dagegen haben sich CF-Grammatiken als adäquat für die Beschreibung von künstlichen Sprachen und hier wieder speziell für Programmiersprachen, erwiesen. Es muß jedoch darauf hingewiesen werden, daß z.B. Algol 60 (wo man der Meinung war, daß es sich um eine umgebungsunabhängige Sprache handelte) nicht umgebungsunabhängig ist, da es dort intuitive Regeln, wie "jeder Variablenname muß in einer Vereinbarung vorkommen" gibt. Allerdings handelt es sich dabei um ein umgehbares Formulierungsproblem.

Bevor wir uns speziellen Fragen der algebraischen Eigenschaften zuwenden, muß noch auf die fundamentale Bedeutung von CF-Grammatiken für den Compilerbau von Rechenanlagen hingewiesen werden. Als Compiler bezeichnet man bekanntlich ein Programm, welches als Eingabedaten ein Programm wie etwa ein Algol- oder Fortranprogramm annimmt, und als Ausgabe einen maschinenorientierten Code erzeugt, der von der Rechenanlage ausgeführt werden kann. Für die meisten Compilertechniken bedient man sich im weitesten Sinne der umgebungsunabhängigen Sprachen.

II. FORMALE POTENZREIHEN ÜBER GRAMMATIKEN

Nachdem wir im ersten Kapitel ganz elementar linguistische Grundbegriffe bereitgestellt haben, mit denen wir hauptsächlich im folgenden operieren werden, interessieren wir uns also weiters für algebraische Eigenschaften von Grammatiken (und damit auch von Sprachen). Da von jetzt ab der Text formalisierter geschrieben werden soll als bisher, wollen wir zunächst einige Hilfsmittel aus der Algebra heranziehen.

Es sei gegeben eine nichtleere Menge V. Mit dieser Menge bilden wir induktiv die kartesischen Produkte

$$V_1 := V$$

$$V_{n+1} := V \S V_n$$

Da "x" als Symbol für Zeichen eines Alphabets (oder auch für Wörter) noch verwendet wird, deutet "§" das kartesische Produkt von Mengen an (in Ermangelung eines anderen Symbols).

Die Vereinigung aller dieser Mengen bezeichnen wir mit

$$W(V) := \bigcup_{n \text{ aus } N} V_n$$

Für eine spezielle Kette w_i aus W(V) schreiben wir auch

$$w_i = v_i^1 \, v_i^2 \, \ldots\ldots \, v_i^n := (v_i^1, v_i^2, \ldots, v_i^n)$$

mit $i = 1, 2 \ldots\ldots\ldots p$

Der Absolutbetrag

$$\left| w_i \right| = n$$

heißt Länge von w_i.

Da eine Kette eine geordnete endliche Folge ist, ist es erlaubt, zwei Ketten durch eine Verknüpfungsoperation zu einer dritten Kette zu verbinden.

$$w_1 \frown w_2 := v_1^1 \; v_1^2 \; \ldots\ldots \; v_1^n \; v_2^1 \; v_2^2 \; \ldots\ldots \; v_2^k$$

Diese Operation " $\frown$ " heißt Konkatenation; die vorstehende Verknüpfung zweier Ketten kann - wenn keine Verwechslungen zu befürchten sind - abgekürzt werden zu

$$w_1 \, w_1 w_2 := w_1 \frown w_2$$

Wird die Menge V als "Alphabet oder Vokabular" interpretiert, so heißt W(V) Menge aller Worte über V; w_i ist dann das i-te Wort aus W(V).

Da in der Algebra eine Menge zusammen mit einer zweistelligen Verknüpfung als Halbgruppe definiert ist, ist daher in Analogie (und bei Interpretation von V als Vokabular) die Menge W(V) zusammen mit der Konkatenation als Halbgruppenverknüpfung eine Worthalbgruppe über V.

In kartesischen Produkten ist die Gleichheit von Elementen durch die komponentenweise Gleichheit definiert; daher erhält man leicht den

Satz: Jedes w_i aus W(V) ist eindeutig als Produkt von Elementen v aus V darstellbar.

Sei jetzt F eine beliebige Halbgruppe und E eine Teilmenge davon. E^n sei induktiv definiert durch

$$E^1 = E$$

$$E^{n+1} = E E^n$$

Die Vereinigungsmenge

$$E^* = \bigcup_{n \text{ aus } N} E^n$$

heißt die von E erzeugte Unterhalbgruppe.

Gilt

$$E^* = F$$

so heißt E ein Erzeugendensystem von F.

Seien F und H Halbgruppen. Eine Abbildung

$$p: F \longrightarrow H$$

heißt ein Homomorphismus von F in H, wenn gilt

$$p(fg) = p(f)\, p(g) \qquad \text{(für f, g aus F)}$$

Da es auch spezielle Homomorphismen gibt (wie Epimorphismus, Isomorphismus), wird hier zunächst die Menge aller Homomorphismen mit Hom (F,H) bezeichnet.

Sei wieder eine Halbgruppe F gegeben und sei E eine Teilmenge davon.

Gilt 1. $E^* = F$

2. Zu jeder Halbgruppe H und jeder Abbildung

$p_o: E \longrightarrow H$

gibt es einen Homomorphismus Hom (F,H), derart, daß

$p \mid E = p_o$,

dann heißt F frei über E.

($p \mid E$ ist diejenige Abbildung, die durch Einschränkung der Abbildung p auf die Menge E gegeben ist).

Der Sinn dieser Definitionen ist darin zu sehen, daß damit die Begriffe eines (ohne Beweis gegebenen) Satzes bereitgestellt wurden, der eine Beziehung zwischen Halbgruppen und Worthalbgruppen herstellt, was die algebraische Betrachtungsweise von Grammatiken und Sprachen erst rechtfertigt.

Satz: Die Halbgruppe F ist genau dann frei, wenn es eine Menge V gibt, so daß F isomorph zu W(V) ist.

(Ein Homomorphismus heißt Isomorphismus, wenn p surjektiv und injektiv, also bijektiv ist.)

Es stimmt daher die Klasse der (uns interessierenden) Worthalbgruppen mit der Klasse der freien Halbgruppen überein.

Leider gibt es in der algebraischen Linguistik das Problem, daß man annimmt, die Worthalbgruppe enthält ein Einselement.

Ein Element e_l (e_r) aus einer Halbgruppe F heißt Linkseins (Rechtseins), wenn für alle f aus F gilt:

$$e_l f = f \quad (\text{bzw. } f\, e_r = f).$$

(Eine Halbgruppe mit Einselement heißt auch Monoid.)

Für diese Annahme besteht kein Zwang, obwohl man natürlich ein Einselement zu W(V) hinzunehmen kann. Man faßt dieses Einselement formal als das "leere Wort" auf (das "keine Buchstaben" enthält und die "Länge" Null hat); doch ist diese erweiterte W(V) nicht mehr frei im Sinne unserer Definition.

Diese Einschränkung ist im Auge zu behalten, wenn trotzdem von "freien Monoiden" gesprochen wird, um mit der Literatur konform zu gehen.

Bevor wir zu umgebungsunabhängigen Sprachen übergehen, seien skizzenhaft Beispiele für die Verwendung von Halbgruppen bei Programmiersprachen gegeben (und zwar speziell bei Listensprachen).

Das Gesamtalphabet von SNOBOL besteht aus folgenden Zeichen:

A | B | C | ..... | X | Y | Z |

0 | 1 | 2 | ,,,,,,,,, | 9 | + | - | / | * | . | , | ' | b | = | (|) | $ |

Das Symbol b zeigt ein blank an, der Strich | ist kein Zeichen des Gesamtalphabets und zeigt hier nur die Trennung an.

Eine Kette ist eine endliche geordnete Folge von Symbolen aus V (also ein w_i aus W(V)).

Ein SNOBOL-Programm ist eine geordnete Menge von Statements.
Die Verknüpfung der Konkatenation ist beispielsweise so gegeben

X 'AB' X

ist identisch mit

'ABABAB'

wenn X = 'AB'.

Die Definition von rekursiven Funktionen z.B. in LISP kann folgendermaßen als Homomorphismus definiert werden:

Beispiel:
map soll eine Funktion f jedem Element einer Liste zuordnen, z.B.:
map (sqrt,(1,2,3,4)) = (1.00, 1.41, 1.73, 2.00)

Dies wird als rekursive Funktion so geschrieben:

```
let recursive map (f,x) = if nullx then nil
                          else cons(f(carx),map(f(cdrx))).
```

Der entsprechende Homomorphismus lautet:

Ist $f : X \longrightarrow Y$, so ist die einstellige Funktion map_f definiert durch

$$map_f : X^1 \longrightarrow Y^1$$

$$map_f x = f_x \qquad \text{für } x \text{ in } X$$

Dabei bezeichnet X^1 die freie Halbgruppe mit Einselement (analog Y^1).

Die gegebenen Definitionen sollen nun beachtet werden, wenn wir ein endliches Vokabular G_1, welches in zwei Mengen aufgespalten werden kann, untersuchen. Diese Teilmengen sind

1) $A = \{\text{Menge der Hilfssymbole}\}$

2) $B = \{\text{Menge der Basiszeichen}\}$

(A und B disjunkt).

Eine <u>Sprache</u> besteht dann aus einer Teilmenge des freien Monoids über B, welches mit B^* bezeichnet wird.

Jede Kette $b_1, b_2 \ldots\ldots b_n$ aus B^* können wir durch ein Verfahren r in die Menge Z der ganzen Zahlen abbilden, sodaß

$$r : B^* \longrightarrow Z$$

gilt.

Diese Abbildung liefert eine <u>formale Potenzreihe</u>, wenn

$$R = \sum_{i}^{n} \langle r, b_i \rangle \; b_i = \langle r, b_1 \rangle \, b_1 + \langle r, b_2 \rangle \, b_2 + \ldots\ldots + \langle r, b_n \rangle \, b_n$$

gilt ("formal" deswegen, weil es egal ist, ob die Reihe divergiert oder konvergiert).

Die Ausdrücke $\langle r, b_i \rangle$

bezeichnet man als die <u>Koeffizienten</u> der Potenzreihe.

Vereinbarungsgemäß können Summanden der Gestalt $0b$ weggelassen sowie Summanden der Gestalt $1b$ als b geschrieben werden. Umgekehrt wird b als $1b$ verstanden und ein Wort, das nicht in der Potenzreihe auftritt, als $0b$.

Zu einer vorgelegten Potenzreihe R bildet die Menge derjenigen Ketten, die einen Koeffizienten ungleich Null aufweisen, die sogenannte Stütze ("support") der Potenzreihe R,

$$\mathrm{Sup}\ (R) = (b_i \text{ aus } B^* \mid \langle r, b_i \rangle \neq 0).$$

Weiters heißt eine Potenzreihe charakteristisch, wenn jeder Koeffizient entweder Null oder Eins ist.

Da uns in erster Linie algebraische Eigenschaften interessieren, wollen wir verschiedene algebraische Operationen über Potenzreihen betrachten.

Ganz allgemein wird in der Algebra eine algebraische Struktur mit zwei zweistelligen inneren Verknüpfungen (im allgemeinen Addition und Multiplikation genannt) als Ring bezeichnet, wenn für beliebige Elemente a,b,c,x aus dieser Struktur folgende Gesetze erfüllt sind:

a) kommutatives Gesetz der Addition $[a+b = b+a]$
b) assoziatives Gesetz der Addition $[a+(b+c)= (a+b)+c]$
c) Umkehrbarkeit der Addition $[a+x = b]$
d) assoziatives Gesetz der Multiplikation $[a.(b.c)=(a.b).c]$
e) distributive Gesetze $[a.(b+c) = a.b + a.c;$
$(b+c).a = b.a + c.a]$

Angewendet auf Potenzreihen (welche nach Definition den Charakter von Ringen haben), kann die Multiplikation mit einer ganzen Zahl erklärt werden (unter Beachtung der Nicht-Kommutativität):

$$n\ R := \langle n\ r,\ b_i \rangle .$$

Die Multiplikation zweier Potenzreihen R_1 und R_2 ist durch eine Potenzreihe R mit den Koeffizienten

$$\langle r_1 r_2, b\rangle = \sum_i \sum_j \langle r_1, b_i\rangle \langle r_2, b_j\rangle$$

gegeben, wobei $b = b_i b_j$.

Ähnlich läßt sich die Addition zweier Potenzreihen R_1 und R_2 definieren, mit den Koeffizienten

$$\langle r_1 + r_2, b\rangle = \langle r_1, b\rangle + \langle r_2, b\rangle .$$

Für die Stützen von Potenzreihen gilt:

(1) $\mathrm{Sup}\,(RR_1) = \mathrm{Sup}\,(R) . \mathrm{Sup}\,(R_1)$

(2) $\mathrm{Sup}(R+R_1) = \mathrm{Sup}\,(R) \cup \mathrm{Sup}\,(R_1)$

Zwei Potenzreihen heißen <u>äquivalent modulo Grad n</u>, wenn für jede Kette b der Länge $\leq n$ gilt

$$\langle r_1, b\rangle = \langle r_2, b\rangle .$$

$$[R_1 \equiv R' (\mathrm{mod\ deg\ } n)]$$

Mit diesem Begriff können wir die Ringstruktur erweitern.

Bekanntlich hat man eine <u>topologische Struktur</u>, wenn in einer Menge T ein System von Teilmengen T_k von T ausgezeichnet ist, derart, daß die beiden folgenden Axiome gelten:

(1) Die Vereinigung von Mengen aus T_k ist eine Menge von T_k. Die leere Menge gehört zu T_k.

(2) Der Durchschnitt endlich vieler Mengen aus T_k ist eine Menge von T_k. Die Menge T gehört zu T_k.

Die Mengen von T_k heißen <u>offene Mengen</u> bezüglich der topologischen Struktur. (Es können auch andere, gleichwertige Axiomensysteme angegeben werden, auf die wir nicht einzugehen brauchen.)

Ein topologischer Ring ist dann eine Menge E, wobei E nichtleer ist, welche mit einer Ringstruktur und einer topologischen Struktur derart versehen ist, daß folgende Axiome erfüllt sind:

(1) Die Abbildung $(x,y) \longrightarrow x + y$ des topologischen Produktes $E \S E$ in E ist stetig.

(2) Die Abbildung $x \longrightarrow -x$ von E in E ist stetig.

(3) Die Abbildung $(x,y) \longrightarrow x.y$ von $E \S E$ in E ist stetig.

Eine Menge T mit einer auf T erklärten topologischen Struktur heißt ein topologischer Raum. Dann erklärt man das topologische Produkt auf folgende Weise:

Seien T_g (g aus einer Indexmenge I) topologische Räume; sei $T = \prod_{g \text{ aus } I} T$ das kartesische Produkt der Mengen T_g.

Die Mengen der Form $\prod_{g \text{ aus } I} O_g$ (wobei für alle g O_g in T_g offen ist und $O_g = T_g$ für alle g bis auf endlich viele ist) bilden eine Basis einer Topologie auf T (ein System H von offenen Teilmengen eines topologischen Raumes T heißt eine Basis der Topologie, wenn jede offene Menge aus T Vereinigungsmenge von Elementen aus H ist).

Diese definierte Topologie heißt die Produkttopologie von T,

$$T = \prod_{g \text{ aus } I} T_g$$

Topologische Stetigkeit wird folgendermaßen definiert:
Eine Abbildung f eines topologischen Raumes T in einen topologischen Raum T' heißt stetig in einem Punkt x aus T, wenn es zu jeder Umgebung U' von f(x) in T' eine Umgebung U von x in T gibt, sodaß $f(U) \subset U'$. (Umgebung eines Punktes x in einem topologischen Raum T ist eine Teilmenge von T, die mit x eine x umfassende offene Menge enthält.)

Um diese Definitionen auf die Potenzreihen anzuwenden, brauchen wir nur noch den Begriff des Umgebungsfilters von x.

Ein nichtleeres Mengensystem F von Teilmengen einer Menge M heißt ein Filter auf M, wenn gilt:

(1) Die leere Menge gehört nicht zu F
(2) Jede Teilmenge von M, die eine Menge aus F umfaßt, gehört zu F
(3) Der Durchschnitt endlich vieler Mengen aus F gehört zu F.

Klarerweise bilden die Umgebungen eines Punktes x eines topologischen Raumes ein Filter (Umgebungsfilter).

Die oben erwähnte Abbildung f ist genau dann stetig in x, wenn für jeden Filter F, der gegen x konvergiert, f(x) Grenzwert von f bezüglich F ist. Schließlich ist ein Punkt x aus einem topologischen Raum T ein Grenzwert des Filters, wenn jede Umgebung von x eine Menge aus F umfaßt (F konvergiert gegen x). Nach diesen Erklärungen kehren wir zur Potenzreihe zurück.

Sei eine unendliche Folge von Potenzreihen R_1, R_2 .... gegeben, sodaß für alle n und alle n' größer n gilt:

$$R_n \equiv R_{n'} \pmod{\deg n}$$

Dann ist der Limes der Folge definiert; dieser entsteht nämlich dann, wenn man alle R_n des Grades größer als n wegläßt.
Wir erhalten daher nach dieser Prozedur einen topologischen Ring (mit den geforderten Eigenschaften) für die Potenzreihen.

Nachdem jetzt einige algebraische Eigenschaften der Potenzreihen erklärt worden sind, können wir uns dem einer umgebungsunabhängigen Grammatik zugeordneten Gleichungssystem zuwenden (und der Konstruktion von R).

Sei G eine umgebungsunabhängige Grammatik mit einer Menge A von Hilfssymbolen (a_1, ... a_n; a_1 entspricht im System (3) dem Symbol S). Auf diese Weise läßt sich G als Gleichungssystem darstellen. Es seien nämlich Ausdrücke der Art

$$a_i \longrightarrow p_{i,j} \quad [\text{mit} \quad (1 \leq j \leq m_i)]$$

Regeln, sodaß p_{i1}, ..., p_{i,m_i} Ketten sind.

Es entspricht a_i ein Polynomausdruck s_i derart, daß

$$s_i = p_{i,1} + p_{i,2} + ... + p_{i,m_i} \quad ;$$

dann entspricht außerdem der Grammatik G das Gleichungssystem

$$a_1 = s_1$$
$$a_2 = s_2$$
$$\vdots \quad \vdots$$
$$a_n = s_n .$$

Da wir weiter oben bemerkt haben, daß der Ring der Potenzreihen abgeschlossen ist bezüglich der Operationen: Multiplikation mit einer ganzen Zahl, Multiplikation zweier Potenzreihen und Addition, können wir jede Gleichung

$$a_i = s_i$$

auffassen als Abbildung g_i, die ein n-Tupel $(R_1 ..... R_n)$ von Potenzreihen überführt in Potenzreihen, die man durch Ersetzung der a_j in s_i durch R_j, definiert.

Daher definiert das Gleichungssystem

$$\begin{aligned} a_1 &= s_1 \\ &\vdots \quad \vdots \\ a_n &= s_n \end{aligned}$$

eine Abbildung g, sodaß

$$g(R_1, \ldots, R_n) = (R_1', \ldots R_n')$$

mit $$R_1' = g_i(R_1, \ldots, R_n).$$

Für die Lösung des Gleichungssystems betrachten wir die unendliche Folge der n-Tupel von Potenzreihen

$$k_n = (R_{0,1}, \ldots R_{0,n}) = (0, \ldots 0)$$

$$k_1 = (R_{1,1}, \ldots R_{1,n})$$

$$\vdots$$

$$k_j = (R_{j,1} \ldots R_{j,n})$$

$$\vdots$$

wo für jedes i,j $(j \geq 0)$ gilt:

$$R_{j,i} = g_i (R_{j-1,1}, \ldots R_{j-1,n}).$$

0 ist die Potenzreihe, wo alle Koeffizienten Null sind. Jede $R_{j,i}$ hat nur endlich viele von Null verschiedene Koeffizienten und ist daher ein Polynom.

Für jedes i,j,j', sodaß $j' > j \geq 0$, $1 \leq i \leq n$, gilt weiters

$$R_{j,i} \equiv R'_{j,n} \pmod{\deg j}.$$

Daher ist der Grenzwert der unendlichen Folge $R_{1,i}, R_{2,i}, \ldots$ für jedes i definiert (formal geschrieben $R_{\infty,i}$).

Für unser Gleichungssystem

$$\begin{aligned} a_1 &= s_1 \\ &\vdots \\ a_n &= s_n \end{aligned}$$

ist das n-Tupel $(R_{\infty,1}; \ldots\ldots R_{\infty,n})$ dann <u>Lösungstupel.</u>

Jede in der Lösung eines solchen algebraischen Systems enthaltene Reihe heißt <u>algebraische Reihe.</u> Weist die Reihe nur positive Koeffizienten auf, so heißt sie <u>umgebungsunabhängige Reihe.</u>

Um das geschilderte Vorgehen beispielhaft zu erläutern, untersuchen wir die Konstruktion der Lösung der Grammatik

$$S \longrightarrow S\ b\ S$$
$$S \longrightarrow a$$

Da diese Grammatik nur ein Hilfssymbol aufweist, nämlich $a_1 = S$, besteht das zugeordnete Gleichungssystem aus nur einer Gleichung:

$$S = a + SbS$$

Da diese Gleichung die Abbildung g definieren soll mit

$$g(R) = a + RbR$$

(wobei R eine Potenzreihe darstellt), betrachten wir die unendliche Folge

$$\begin{aligned} k_0 &= R_0 = 0 \\ k_1 &= R_1 = a + R_0bR_0 = a + 0b0 = a \\ k_2 &= R_2 = a + R_1bR_1 = a + aba \\ &\vdots \end{aligned}$$

Für jedes j,j' sodaß $j' > j > 0$, erhalten wir bekanntlich

$$R_j \equiv R_{j'} \pmod{\deg j} .$$

Der Grenzwert ist daher definiert, also erhalten wir ferner

$$R_\infty = \sum_n \left[\begin{matrix}2n\\ n\end{matrix}\right] \frac{1}{n+1} (ab)^n a =$$

$$= a+aba + 2(ab)^2a + 5(ab)^3a + 14(ab)^4a + \ldots.$$

Diese Reihe stellt die Lösung der Gleichung

$$S = a + SbS$$

dar.
Es sei darauf verwiesen, daß diese Reihe nicht charakteristisch ist.

Die Stütze dieser Reihe ist die von der Grammatik erzeugte Sprache:

$$L(G) = (\Lambda, a, aba, ababa, abababa, \ldots.)$$

(Λ ist das leere Wort.)

III. WEITERE OPERATIONEN ÜBER POTENZREIHEN

Eine Potenzreihe, in derder Koeffizient des leeren Wortes Null ist, heißt quasi-regulär:

$$\langle r, \Lambda \rangle = 0$$

Daher ist die Nullreihe quasi-regulär.

Sei nun s eine quasi-reguläre Reihe und betrachten wir die Folge

$$\begin{aligned} t_1 &= s \\ t_2 &= s + s^2 \\ t_3 &= s + s^2 + s^3 \\ &\vdots \end{aligned}$$

Diese Reihe strebt gegen einen Limes, den man mit s^* bezeichnet. Es gilt offenbar

$$s + s^2 + s^3 + \ldots + s^P + \ldots = s^*$$
$$s + s^* s = s + s s^* = s^*$$

s^* nennt man das Quasi-inverse von s.

Neben der Quasi-Regularität interessiert uns noch das Hadamard'sche Produkt zweier Reihen.

Seien R_1 und R_2 zwei Potenzreihen. Ihr Hadamard'sches Produkt ist dann die Potenzreihe R mit Koeffizienten

$$\langle r_1 \odot r_2, b \rangle = \langle r_1, b \rangle \langle r_2, b \rangle,$$

für alle b.

Sind R_1 und R_2 charakteristische Reihen, dann ist auch $R_1 \odot R_2$ eine charakteristische Reihe; die Stütze von R ist dann

$$\mathrm{Sup}(R_1 \odot R_2) = \mathrm{Sup}\ (R_1) \cap \mathrm{Sup}\ (R_2).$$

Es lassen sich mit den beiden Operationen dieses Abschnittes etliche Beziehungen zur klassischen Analysis finden, die uns aber hier nicht weiter beschäftigen müssen.

IV. ARTEN VON UMGEBUNGSUNABHÄNGIGEN GRAMMATIKEN

Nachdem im zweiten und dritten Abschnitt beschreibende Hilfsmittel erklärt wurden, interessieren wir uns jetzt dafür, welche Beziehungen zu den umgebungsunabhängigen Grammatiken konstruiert werden können.

Wie erwähnt, ist eine solche Grammatik dadurch ausgezeichnet, daß links vom Pfeil genau ein Hiffssymbol steht (bei allen Regeln). Bestimmte Arten von umgebungsunabhängigen Grammatiken können durch Spezifikationen der Ausdrücke rechts vom Pfeil gewonnen werden. Allerdings muß man festhalten, daß sich die Menge der umgebungsunabhängigen Grammatiken nicht injektiv in die Menge der umgebungsunabhängigen Sprachen abbilden läßt (eine Abbildung heißt injektiv, wenn verschiedene Elemente des Definitionsbereiches stets verschiedene Bilder haben).

Eine Regel heißt <u>linear</u>, wenn sie von der Form

$$C \longrightarrow x\ D\ y$$

ist, mit C, D aus A und x,y aus B.

Sie heißt rechtslinear (bzw. linkslinear), wenn sie von der Form

$$C \longrightarrow x\ D \quad (\text{bzw. } C \longrightarrow D\ x)$$

ist.
Die Regel heißt abschließend, wenn sie von der Form

$$C \longrightarrow x$$

ist.

Eine umgebungsunabhängige Grammatik heißt dann linear, wenn sie nur lineare oder abschließende Regeln enthält; sie heißt rechtslinear (bzw. linkslinear), wenn jede Regel rechtslineare (bzw. linkslinear) oder abschließend ist. Eine Grammatik heißt ferner einseitig linear, wenn sie rechts-linear oder linkslinear ist.

Schließlich bezeichnet man eine umgebungsunabhängige Grammatik als metalinear, wenn jede Regel linear oder abschließend oder von der Form

$$S \longrightarrow x$$

ist und wenn es keine Regel gibt, in der rechts S vorkommt.

Enthält eine umgebungsunabhängige Grammatik kein Symbol C (aus A), von welchem man sowohl eine Kette b' als auch eine Kette $b_1 C b_2$ (b', b_1, b_2 aus G_1^*) ableiten kann, dann ist die erzeugte Sprache L(G) endlich und die Grammatik heißt Polynomgrammatik.

Ist jetzt eine Grammatik gegeben, die polynomial oder einseitig linear oder linear oder metalinear oder umgebungsunabhängig ist, so bezeichnet man die jeweils dazugehörigen Potenzreihen und die Stützen entsprechend.

Die korrespondierenden Symbole haben diese Form:

	Potenzreihe R	Stütze Sup (R)
Polynomial	P+	Sup (P+)
einseitig linear	L_0+	Sup (L_0+)
linear	L+	Sup (L+)
metalinear	L_m+	Sup (L_m+)
umgebungsunabhängig	C+	Sup (C+)

Es ist jetzt leicht, eine Hierarchie zu konstruieren. Nach drei Sätzen aus der Theorie der Regelsprachen existiert folgende Hierarchie (ohne Beweis wiedergegeben):

(1) Die einseitig linearen Sprachen sind eine echte Teilmenge der linearen Sprachen.
(2) Die linearen Sprachen sind eine echte Teilmenge der metalinearen Sprachen.
(3) Die metalinearen Sprachen sind eine echte Teilmenge der umgebungsunabhängigen Sprachen.

Für die Stützen Sup (K) gilt ebenfalls eine Hierarchieordnung

$$\text{Sup}(P+) \subset \text{Sup}(L_0+) < \text{Sup}(L+) < \text{Sup}(L_m+) < \text{Sup}(C+)$$

Eine Grammatik heißt <u>einhüllend</u>, wenn sie ein Symbol C aus A und nichtleere Ketten b_1, b_2 aus G_1^* enthält, sodaß

$$C \Longrightarrow b_1\ C\ b_2$$

gilt.

"$\Longrightarrow$" deutet an, daß es sich um eine nicht-unmittelbare Ableitung handelt (zum Unterschied von "$\longrightarrow$":

$$C \longrightarrow x\ C\ y$$

bedeutet, daß x C y in <u>einem</u> Schritt abgeleitet wird.).

Die Stützen der vorhin erwähnten Potenzreihen haben ebenfalls interessante algebraische Eigenschaften:

Sup (P+) ist abgeschlossen bezüglich Vereinigung, Produkt, Durchschnitt

Sup (L_0+) ist abgeschlossen bezüglich Vereinigung, Produkt, Sternoperation nach Kleene, Durchschnitt, Komplement

Sup (L+) ist abgeschlossen bezüglich Vereinigung

Sup (L_m+) ist abgeschlossen bezüglich Produkt

Sup (C+) ist abgeschlossen bezüglich Vereinigung, Produkt und Sternoperation,

Die vorstehende Betrachtung von verschiedenen Unterfamilien der Klasse der umgebungsunabhängigen Grammatiken bezog sich auf Struktureigenschaften.

Aus anderen möglichen Klassifikationsgesichtspunkten soll hier nur einer herausgegriffen werden.

Eine umgebungsunabhängige Grammatik $G = (A,B,P,S)$,

$G_1 = A \cup B$, mit $L(G) \neq \emptyset$ heißt <u>reduziert</u>, wenn folgende zwei Bedingungen erfüllt sind:

(1) für jede Variable C aus A gibt es eine Kette b aus B^* mit $C \longrightarrow b$,

(2) für jede Variable C aus $A - \{S\}$ gibt es Ketten b_1, b_2 aus G_1^* (und daher sogar aus B^*) mit $S \rightarrow b_1 C b_2$.

<u>Satz:</u> Zu jeder umgebungsunabhängigen Grammatik $G = (A,B,P,S)$ kann man eine äquivalente, reduzierte, umgebungsunabhängige Grammatik $G' = (A',B',P',S)$ konstruieren.

(Zwei Grammatiken G und G' heißen äquivalent, wenn $L(G) = L(G')$ gilt.)

Eine umgebungsunabhängige Grammatik heißt <u>sequentiell,</u> wenn die Hilfssymbole a_i in der Form

a_1 (Anfangssymbol), a_2, ..., a_n angeordnet werden können, sodaß es keine Regel

$$a_i \longrightarrow b_1\ a_j\ b_2 \qquad (j < i)$$

gibt.
(Ein nicht-terminales Zeichen verschwindet ein für allemal nach Anwendung aller Regeln, in denen es vorkommt).

Sei S+ die Familie der sequentiellen Potenzreihen und Sup (S+) die Familie ihrer Stützen. Dann ist Sup (S+) abgeschlossen bezüglich Vereinigung, Produkt und Sternoperation.

V. EINE WEITERE CHARAKTERISIERUNG VON UMGEBUNGS-UNABHÄNGIGEN SPRACHEN

In diesem Abschnitt wird gezeigt, daß die uns interessierenden Sprachen auch anders beschrieben werden können als bisher.

Sei zunächst B ein Alphabet. Eine reguläre Menge über B wird dann so definiert:

(1) Jede Menge, die nur ein Element aus B enthält, ist regulär.

(2) Die Vereinigung und das Produkt zweier regulärer Mengen sind wieder reguläre Mengen.

(3) Die Kleenesche Sternoperation über einer regulären Menge liefert wieder eine reguläre Menge.

(4) Nur Mengen, die durch (1) - (3) in endlich vielen Schritten erhalten werden können, sind reguläre Mengen.

Nach (1) und (2) ist jede endliche Menge von Worten über B regulär; nach (2) und (3) auch Vereinigung, Produkt und Stern solcher Mengen.
Reguläre Mengen werden auch als reguläre Sprache bezeichnet.

Es gilt dann der

Satz: Eine Menge M ist regulär genau dann, wenn sie eine einseitige lineare Sprache ist.

Für die algebraischen Eigenschaften von regulären Mengen ist wichtig der

Satz: Die Menge der regulären Sprachen über einem Alphabet B ist abgeschlossen bezüglich Durchschnittsoperation und Komplementbildung.

Sei nun B wieder die Menge der Endsymbole. Ferner sei eine Teilmenge J_1 von B ausgezeichnet, die erlaubte Initialsymbole einer möglichen Kette enthält; analog eine Teilmenge J_2, die erlaubte Finalzeichen umfaßt.
Dann kann die Menge der Wörter von B^*, die mit einem Zeichen aus J_1 beginnen und mit einem Zeichen aus J_2 enden, durch die Formel

$$J_1 B^* \cap B^* J_2$$

dargestellt werden.

Sei weiters vorgeschrieben, daß die möglichen Ketten keine der Zwei-Zeichenketten aus einer definierten Teilmenge J_3 von B§B enthalten dürfen. Diese Menge von nicht-erlaubten Ketten ist durch die Formel

$$B^* J_3 B^*$$

darstellbar.

Unter Berücksichtigung, daß man bei der zweiten Formel für die Gewinnung der erlaubten Ketten das Komplement bilden muß, können beide Formeln zusammengefaßt werden zu einer Sprache K,

$$K = \left[J_1 B^* \cap B^* J_2\right] \cap \left[B^* - (B^* J_3 B^*)\right] ,$$

d.h. die Sprache K ist gleich dem Durchschnitt des Quasi-Ideals der Menge von Ketten, die mit einem Zeichen aus J_1 beginnen und mit einem Zeichen aus J_2 enden, mit dem Komplement des zweiseitigen Ideals der verbotenen Wörter.

In der Algebra wird eine nichtleere Teilmenge E einer Halbgruppe F als Linksideal (Rechtsideal) bezeichnet, wenn sie bezüglich Linksmultiplikation (Rechtsmultiplikation) mit Elementen aus F abgeschlossen ist. Ist E gleichzeitig Links- und Rechtsideal, so heißt E zweiseitiges Ideal. Ferner gilt folgender

Satz: Sei $\{T_i : i \text{ aus Indexmenge } I\}$ eine Familie von Links-(Rechts-, zweiseitigen) Idealen einer Halbgruppe F, so ist auch der Durchschnitt (sofern er nicht leer ist) und die Vereinigung dieser Familien wieder ein Links-(Rechts-, zweiseitiges) Ideal.

Die eben definierte Sprache K wird in der Literatur häufig als Standard-K-Sprache bezeichnet.

Es gilt jetzt folgender

Satz: Zu jeder beliebigen regulären Sprache M läßt sich eine Standard-K-Sprache K und ein Homomorphismus p finden, sodaß gilt

$$p(K) = M.$$

Neben den Standard-K-Sprachen benötigen wir noch zur anderen Charakterisierung der uns interessierenden umgebungsunabhängigen Sprachen die sogenannte Dyck-Sprache.

Sei B ein Alphabet mit zwei Teilmengen B_1 und B_2, sodaß gilt

$$B_1 \cup B_2 = B$$
$$B_1 \cap B_2 = \emptyset .$$

Sei ferner d eine Abbildung

$$d: B_1 \longrightarrow B_2$$

und P ein Produktionssystem mit (B,H), wobei

$$H = (bb' \longrightarrow e \mid b \text{ aus } B_1,\ b' = d(b))$$

gilt.

Ein Wort x aus B^* heißt <u>d-reduzierbar</u> zu y (aus B^*), wenn $x \longrightarrow y$ gilt. Die Menge aller zu e reduzierbaren Worte heißt <u>Dycksprache D</u> (e ist das leere Wort).

<u>Beispiel:</u>

Sei B = $\{(,[,),]\}$ ein Alphabet und sei

d: $\{(,[\} \longrightarrow \{),]\}$ durch d $(() =)$ und

d([ ) =] definiert. Dann sind etwa

(), ([ ]), $\big([() ()] \ ([ ])\big)$ reduzierbare Worte.

Die Dycksprache D ist die Menge aller wohlgeformten Klammerausdrücke der beiden Klammerarten '(' bzw. ')' und '[' bzw. ']'.

Es gilt folgender

<u>Satz:</u> Jede Dycksprache D ist umgebungsunabhängig.

Als Gesamtresultat ergibt sich in diesem Abschnitt der <u>Homomorphiesatz:</u> Ist L aus B^* eine umgebungsunabhängige Sprache, dann gibt es ein Alphabet B_1 (welches eine echte Teilmenge von B ist), eine Dycksprache D (Teilmenge von B_1^*) und eine Standard-K-Sprache K, sodaß für den Homomorphismus

p von B_1 auf B, welcher durch
p(a) = a, für **a** aus B
p(a) = e, für **a** aus B_1 - B

definiert ist, die Beziehung

$$L = p(K \cap D)$$

gilt.

Demnach ist jede umgebungsunabhängige Sprache das homomorphe Abbild des Durchschnittes einer regulären Sprache mit einer Dyck-Sprache und der Homomorphiesatz das Fundamentaltheorem für umgebungsunabhängige Sprachen.

Durch die Anwendung des Homomorphiesatzes ist daher eine andere Charakterisierung der umgebungsunabhängigen Sprachen als durch Potenzreihen gegeben.

VI. UNENTSCHEIDBARKEIT

In diesem Abschnitt wird die Unentscheidbarkeit einer Anzahl von Problemen nachgewiesen.

Ausgangspunkt ist die Unentscheidbarkeit des Halteproblems für Turing-Maschinen; d.h. daß es keinen Algorithmus gibt, der für jede Turingmaschine und jedes Wort, das als Eingabe dient, feststellt, ob die Turingmaschine nach endlich vielen Schritten stoppt oder nicht.

Speziell ist ein Wortproblem durch ein Tripel (P_s, x, y) gegeben, wobei $P_s = (G_1, P)$ ist und x, y Elemente aus G_1^* mit $x \neq y$.

Das Wortproblem heißt lösbar, wenn y aus x abgeleitet werden kann. Aus der Unentscheidbarkeit des Halteproblems folgt jedoch, daß es keinen abbrechenden Algorithmus gibt, der für jedes Wortproblem feststellt, ob es lösbar ist oder nicht.
Eine Erweiterung dieses Problems bildet das sogenannte Entsprechungsproblem:

Ein Entsprechungsproblem ist ein Tripel (B,D,F). Dabei ist B ein Alphabet von Basiszeichen und D und F sind m-Tupeln ($m \geq 1$) von Worten

$$D = (d_1, d_2, \ldots\ldots, d_m)$$
$$F = (f_1, f_2, \ldots\ldots, f_m)$$

d_i, f_i aus B^* für $1 \leq i \leq m$.

Das Entsprechungsproblem heißt lösbar, wenn es ein n-Tupel I von Zahlen $I = (i_1, i_2, \ldots\ldots, i_n)$ mit $n \geq 1$ und $1 \leq i_j \leq m$ für $1 \leq j \leq n$ gibt, sodaß $d_{i_1}, d_{i_2} \ldots\ldots, d_{i_n} = f_{i_1}, f_{i_2}, \ldots\ldots, f_{i_n}$ gilt.

Das n-Tupel I heißt dann Lösung(stupel) des Entsprechungsproblems.

POST hat nun nachgewiesen, daß es keinen abbrechenden Algorithmus gibt, der für jedes Entsprechungsproblem feststellt, ob es lösbar ist oder nicht (Postscher Entsprechungssatz). Dieser Satz ist übrigens eine Folgerung aus dem Unentscheidbarkeitsnachweis des Wortproblems.

Unter Hinzunahme von einigen Hilfssätzen, die uns aber nicht näher beschäftigen müssen, ergeben sich mehrere Konsequenzen für die uns interessierenden umgebungsunabhängigen Sprachen, die in folgendem Satz zusammengefaßt sind:

Satz:

Seien im folgenden beliebige umgebungsunabhängige Grammatiken G^1 und G^2 mit $L^1 = L(G^1)$ und $L^2 = L(G^2)$ gegeben. Dann sind folgende Fragen unentscheidbar:

(1) Ist $L^1 \cap L^2 \neq 0$?
(2) Ist $L^1 \cap L^2$ eine unendliche Sprache ?
(3) Ist $L^1 \cap L^2$ regulär ?
(4) Ist $L^1 \cap L^2$ umgebungsunabhängig ?
(5) Ist $B^* - L^1 \neq 0$?
(6) Ist $B^* - L^1$ eine unendliche Sprache ?
(7) Ist $B^* - L^1$ regulär ?
(8) Ist $B^* - L^1$ umgebungsunabhängig ?
(9) Ist $L^1 = B^*$?
(10) Ist $L^1 = L^2$?
(11) Ist $L^1 \subsetneq L^2$?

Am Rande sei erwähnt, daß für umgebungsunabhängige Grammatiken G es entscheidbar ist, ob L(G) leer, endlich oder unendlich ist, was für umgebungsabhängige Grammatiken nicht entscheidbar ist.

VII. MEHRDEUTIGKEIT

Im Abschnitt II wurde schon erwähnt, daß eine Potenzreihe R genau dann charakteristisch ist, wenn ihre Koeffizienten

$$\langle r, b \rangle$$

entweder Null oder Eins sind.

Die Lösung der im selben Abschnitt untersuchten umgebungsunabhängigen Grammatik

$$S \longrightarrow SbS$$
$$S \longrightarrow a$$

lautete:

$$R_\infty = \sum_n \left[{2n \atop n} \right] \frac{1}{n+1} \quad (a\ b)^n\ a$$

Eine umgebungsunabhängige Grammatik heißt <u>eindeutig</u>, wenn der "Hauptausdruck" ihrer Lösung charakteristisch ist (im obigen Fall also der Ausdruck

$$\left[{2n \atop n} \right] \frac{1}{n+1} \quad).$$

Da weiters erwähnt wurde (im Abschnitt IV), daß die Menge der umgebungsunabhängigen Grammatiken sich nicht injektiv in die Menge der umgebungsunabhängigen Sprachen abbilden läßt, so bezeichnet man eine umgebungsunabhängige Sprache nur dann als <u>eindeutig</u>, wenn <u>jede</u> sie erzeugende umgebungsunabhängige Grammatik eindeutig ist.

Daß mehrdeutige umgebungsunabhängige Sprachen existieren, hat PARIKH nachgewiesen. So ist zum Beispiel die Sprache

$$L = (a^i b^i c^j \mid i,j \geq 1) \cup (a^i b^j c^j \mid i,j \geq 1)$$

eine mehrdeutige, umgebungsunabhängige Sprache.

Aus dem Unentscheidbarkeitssatz des Abschnittes VI folgt übrigens auch der wichtige

Satz: Sei G^1 eine beliebige, umgebungsunabhängige Grammatik mit $L^1 = L(G^1)$. Dann sind folgende beiden Probleme unentscheidbar

1. Ist G^1 eindeutig ?
2. Ist $L(G^1)$ eindeutig ?

VIII. ENDLICHE TRANSDUKTIONEN

Für die Betrachtung von Transformationen zwischen Sprachen benötigen wir wieder einen Homomorphismus.

Sei L(G) eine beliebige Sprache mit einem Endvokabular B. Existiere ferner zu jedem b aus B eine Sprache L_b über einem anderen Vokabular E.
Mit M(L) bezeichnen wir die Menge aller Ketten in E, die erhalten werden können durch Ersetzung von jedem b_{i_j} in einem Wort

$$w = b_{i_1} b_{i_2} \cdots\cdots b_{i_m} \quad (w \text{ aus } L)$$

durch ein beliebiges Wort aus

$$L_{b_{i_j}} .$$

Es ist klar, daß auf diese Weise eine Abbildung (Homomorphismus) definiert wird.

Angewendet auf umgebungsunabhängige Grammatiken erhalten wir folgendes Analogon:

Seien L und L_b umgebungsunabhängige Sprachen. L soll durch die umgebungsunabhängige Grammatik G (mit Hilfssymbolmenge A) erzeugt werden und jede L_b durch die umgebungsunabhängige Grammatik G_b (mit Hilfssymbolmenge A_b und ausgezeichnetem Element $a_{b,0}$). Die Mengen A_b seien disjunkt. Die umgebungsunabhängige Grammatik $\overline{G}$ habe als Hilfssymbole $A \cup B \cup E \cup A_b$

und als Regeln die Regeln von G und allen G_b sowie die Regel

$$b \longrightarrow a_{b,0} \qquad (b \text{ aus } B) .$$

Dann erzeugt $\overline{G}$ genau die Menge M(L)

IX. FORMALE SPRACHEN UND AUTOMATENTHEORIE

Einleitend sei hier nocheinmal auf die allgemeine Definition verwiesen, wonach eine formale Sprache L über einem Alphabet B eine Teilmenge von B^* ist.

Im Mittelpunkt der Untersuchung von Erkenntnisverfahren, die sich zur Definition von formalen Sprachen im obigen Sinn eignen, steht die Idee des Automaten (welche eine mathematische Theorie ist).

Anschaulich läßt sich ein Automat aber als Gerät vorstellen, das aus mehreren Einheiten besteht.

1) Eingabeeinheit: Diese besteht aus einem, in nebeneinanderliegende Felder geteilten, Eingabeband. Jedes Feld enthält entweder ein Zeichen des Eingabealphabets oder ein Sonderzeichen (welches für den betreffenden Automaten spezifisch ist).
   Außerdem gehört zur Eingabeeinheit noch ein Lesekopf. Dieser befindet sich stets über genau einem Feld des Eingabebandes, dessen Inhalt zusammen mit der Stellung des Lesekopfes eine Konfiguration der Eingabeeinheit bestimmt. Da sich diese aber nur durch Veränderung der Lesekopfstellung ändert, so spricht man dann von einer Bewegung des Lesekopfes.

2. Speichereinheit: Diese besteht aus mindestens einem Speicherband. Jedes von ihnen hat einen Lese-Schreibkopf, welche sich stets über genau einem des ebenfalls in Felder eingeteilten Speicherbandes befinden. Die Eintragungen in die Felder sind entweder Zeichen des Speicheralphabets oder Sonderzeichen (wieder spezifisch). Die Zeichen auf den Speicherbändern und die Stellungen der Lese-Schreibköpfe bestimmen eine Konfiguration der Speichereinheit. Diese Konfiguration kann durch eine Veränderung der Stellung eines Lese-Schreibkopfes ("Bewegung") oder durch Veränderung ("Überschreiben") eines Zeichens unter einem Lese-Schreibkopf abgeändert werden.

3. Steuereinheit: Die Steuereinheit überwacht die Änderungen des Automaten und hat selbst endlich viele Zustände.

Eine Konfiguration der Eingabeeinheit, ein Zustand der Steuereinheit und eine Konfiguration der Speichereinheit bestimmen eine Situation des Automaten.

Ein Automat, der sich in einer bestimmten Situation befindet, geht, wenn er in Tätigkeit gesetzt wird, in eine neue Situation über, aus dieser wieder in eine neue, usw. Seine Tätigkeit wird also durch eine Folge von Situationen beschrieben; kann der Automat aus einer Situation nicht mehr in eine neue Situation übergehen, so hält der Automat.

Eine mögliche Klassifizierung liegt auf der Hand: ist nämlich jede Situation durch die vorhergehende eindeutig bestimmt, so heißt der Automat deterministisch, andernfalls non-deterministisch.
Je nachdem, ob sich der Lesekopf der Eingabeeinheit nach beiden Seiten oder nur nach einer Seite bewegen kann, spricht man von zwei-seitigen bzw. einseitigen Automaten.

Ein spezieller Automat wurde bereits im Abschnitt VI bei den Unentscheidbarkeitsproblemen vorgestellt, nämlich die Turingmaschine. Bei dieser wird nicht zwischen Eingabeband und Speicherband unterschieden, sondern es wird nur ein Band betrachtet, welches beide Funktionen erfüllt.

Ist ein Automat K gegeben (mit einem Eingabealphabet B) und ordnet man jedem Wort b aus B^* Anfangssituationen und Endsituationen zu, so <u>akzeptiert</u> K das Wort b genau dann, wenn es eine zu b gehörende Anfangssituation gibt, aus welcher K durch endlich viele Situationswechsel zu einer zu b gehörenden Endsituation gelangt. Das Wort b gehört dann zu einer mit T(K) bezeichneten Sprache, die eine Teilmenge von B^* ist.

Formal läßt sich ein deterministischer, endlicher Automat K durch ein Quintupel angeben:

$$K = (A, B, d, F, S)$$

Es bedeuten dabei

- A ..... Alphabet von Zuständen der Steuereinheit
- B ..... Alphabet von Eingabezeichen für die Eingabeeinheit
- S ..... ein ausgezeichnetes Element von A (der "Startzustand")
- F ..... eine Teilmenge von A (Menge der Endzustände)
- d ..... Überführungsfunktion, die das Verhalten des Automaten angibt:

  $d : A \times B \longrightarrow A$

Dieser Automat K heißt auch kürzer <u>endlicher</u> Automat K.

Die Überführungsfunktion wird gewöhnlich durch eine Zustandstafel angegeben, die für jeden Zustand A_i aus A und jedes Eingabezeichen b^i aus B den Zustand $d(A_i,b^i)$ angibt.

Beispiel:

d	b^1	b^2
S	S	A_2
A_1	A_2	A_2
A_2	A_1	A_2

mit $A = (S,A_1,A_2)$
$B = (b^1,b^2)$
$F = (A_2)$

Es braucht hier nicht ausdrücklich darauf hingewiesen werden, daß die Untersuchung von Automaten vorteilhaft mit Halbgruppen und Homomorphismen betrieben werden kann.

Diese kurzen Erklärungen waren notwendig, um einige Sätze über die Verknüpfung von umgebungsunabhängigen Sprachen mit der Automatentheorie zu geben.

Hauptsatz für endliche Automaten:

Eine Menge L von Worten über einem Alphabet B ist genau dann regulär, wenn ein endlicher Automat K mit $L = T(K)$ existiert.
(Zur Definition einer regulären Menge siehe Abschnitt V).

Satz:

Sind G^1 und G^2 zwei rechtslineare Grammatiken, so gibt es einen abbrechenden Algorithmus, der entscheidet, ob

$$L(G^1) = L(G^2)$$

gilt.

Als Keller-Automat bezeichnet man einen Automat, für den folgendes 6-Tupel gilt:

$$K = (A,B,H,d,M,S)$$

Es bedeuten:

A ... Alphabet von Zuständen

B ... Alphabet von Eingabezeichen

H ... Alphabet von Keller-Zeichen

M ... ausgezeichnetes Element aus H (das "Startzeichen")

S ... ausgezeichnetes Element aus A (der "Startzustand")

d ... Überführungsfunktion.

Es gilt dann der

Hauptsatz für Kellerautomaten:

Eine Menge L von Worten über einem Alphabet B ist genau dann eine umgebungsunabhängige Sprache, wenn ein Keller-Automat K mit L = T(K) existiert.

Zum Abschluß sei noch angegeben der

Hauptsatz für Turingmaschinen:

Eine Menge L von Worten über einem Alphabet B ist genau dann eine Regelsprache, wenn es eine Turingmaschine C mit T(C) = L gibt.

X. ZUSAMMENFASSUNG

In dem vorliegenden Paper wurden - ausgehend von natürlichen Sprachen - besondere algebraische Eigenschaften von formalen Sprachen untersucht.
Es sei an dieser Stelle aber mit Nachdruck darauf hingewiesen, daß linguistische Fragestellungen nur durch eine entsprechende Interpretation der Menge V des Abschnittes II, nämlich als Alphabet im herkömmlichen Sinne, entstehen.

Aus dieser Tatsache läßt sich leicht ableiten, daß durch andere Interpretationen von V auch andere Fragestellungen resultieren; wesentlich ist dabei aber, daß die formalen Gegebenheiten sich nicht ändern: anders ausgedrückt, gelten auch bei anderen semantischen Belegungen von V alle hier abgeleiteten Sätze.

Es ist daher naheliegend, zu versuchen, der Menge V auch eine verhaltenswissenschaftlich relevante Bedeutung zu geben. Man konstruiert dann damit ein System, dessen Struktur durch die Theorie der formalen Sprachen näher untersucht werden kann. Diese Systemstruktur ist zwar axiomatisch determiniert, es ist aber darauf hinzuweisen, daß über nicht-deterministische Automaten ein Zugang zur wahrscheinlichkeitstheoretischen Betrachtungsweise des Systems möglich wird. Über diese Bemühungen soll in einem anderen Paper berichtet werden.

XI. EINIGE LITERATURHINWEISE

Burstall, R.M. und Laudin, P.J.: Programs and their Proofs: an Algebraic Approach. In: Machine Intelligence, Vol.4 (ed.Meltzer,Michie), Edinburgh at the University Press,1969,S.17-43

Chomsky, N.: Formal properties of grammars. In: Handbook of Mathematical psychology,Vol.II (ed.Luce,Bush,Galanter), John Wiley & Sons, 1963, S.323-418

Chomsky, N. und Schützenberger,M.P.: The algebraic theory of context-free languages. In: Computer programming and formal systems (ed. Braffort, Hirschberg), North Holland Publ. Co.,1963, S.118 - 161.

Deussen,P.:Halbgruppen und Automaten. Springer-Verlag, 1971, Heidelberger Taschenbücher, Bd.99.

Dörr, J. und Hotz, G.: Automatentheorie und formale Sprachen. Bericht einer Tagung des Mathematischen Forschungsinstituts Oberwolfach. Bibliographisches Institut, Mannheim, 1970. Berichte aus dem Mathematischen Forschungsinstitut Oberwolfach, Heft 3.

Ginsburg, S.: A survey of Algol-like and context-free language theory. In: Formal language description languages for computer programming (ed. Steel), North Holland Publ.Co., 1966, S. 86 - 99.

Maurer, H.: Theoretische Grundlagen der Programmiersprachen - Theorie der Syntax. Bibliographisches Institut, Mannheim, 1969, Bd.404/404a*

Parikh, R.J.: Language generating devices. M.I.T. Res.Lab.Electron.Quart.Prog.Rept.60 1961, S. 199 - 212.

Post, E.L.: A variant of a recursively unsolvable problem. Bull.Am.Math.Soc. 52 (1946), S. 264 - 268.

Vigor, D.B., Urquhart D., Wilkinson A.: PROSE-Parsing Recogniser Outpulling Sentences in English. In : Machine Intelligence, Vol4, (ed. Meltzer, Michie) Edinburgh at the University Press, 1969, S. 271 - 284.

Symbole mit fester Bedeutung

G .... Regelgrammatik

L(G).. von der Regelgrammatik G erzeugte Regelsprache

A .... Hilfssymbolmenge
B .... Basissymbolmenge
P .... Produktionsregeln
} einer Regelgrammatik G

S .... Ausgezeichnetes Element der Hilfssymbolmenge A

G_1.... Vereinigungsmenge von A und B, als "Gesamtalphabet" bezeichnet

B^*.... freies Monoid über B

b_i.... Elemente von B^*

r .... Abbildung von Ketten b_i in Z

Z .... Menge der ganzen Zahlen

N .... Menge der natürlichen Zahlen

R .... formale Potenzreihe

Sup(R) Stütze von R