MATHEMATISCHES FORSCHUNGSINSTITUT OBERWOLFACH

T A G U N G S B E R I C H T 35/1978

FORMALE SPRACHEN

21.8. BIS 25.8.1978

Die diesjährige Tagung "Formale Sprachen" stand unter der Leitung von R.V. Book (University of California, Santa Barbara), G. Hotz (Saarbrücken) und H. Walter (Darmstadt). Sie fand zum ersten Mal unter diesem verkürzten Titel statt (der bisherige war:"Automatentheorie und formale Sprachen"), und beschränkte sich auch auf dieses spezielle Gebiet.

Von den 34 Teilnehmern kamen 17 aus dem Ausland, davon 9 aus außereuropäischen Ländern.

Die 26 Vorträge behandelten hauptsächlich die Themen: Grammatiken und deren Transformationen, algebraische Sprachtheorie, Analysealgorithmen und Komplexitätsklassen. Einzelne Vorträge befaßten sich darüberhinaus mit Invarianzeigenschaften von Sprachen, Darstellungs- und Iterationstheoremenund Datenstrukturen.

Am Mittwoch Nachmittag fand bei sommerlichem Wetter die nun schon traditionelle Wanderung statt, und zwar dieses Mal nach St. Roman.

Teilnehmer

H. Alt, Saarbrücken
G. Barth, Pennsylvania, USA
E. Bertsch, Saarbrücken
R.V. Book, Santa Barbara, USA
W. Brauer, Hamburg
P. Deussen, Karlsruhe
H. Ehrig, Berlin
K.Estenfeld, Saarbrücken
S. Ginsburg, Los Angeles, USA
J.Goldstine, Pennsylvania, USA
M.A. Harrison, Berkeley, USA
G. Hotz, St.Ingbert
H. Huwig, Dortmund
M. Jantzen, Hamburg
A.K. Joshi, Philadelphia, USA
H. Jürgensen, Darmstadt
R. Kemp, Saarbrücken
H. Maurer, Graz, Österreich
O. Mayer, Kaiserslautern
B. Monien, Paderborn
A. Nijholt, Amsterdam,Niederlande
M. Nivat, Paris, Frankreich
I. Peak, Budapest, Ungarn
J.F. Perrot, Paris, Frankreich
J. Pflüger, Darmstadt
Ch. Reutenauer, Paris, Frankreich
J. Sakarovitch, Paris, Frankreich
I.H. Sudborough, Evanston, USA
Urbanek, Wien, Österreich
E. Valkema, Kiel
H.K.G. Walter, Darmstadt
D. Wood, Hamilton, Kanada
D. Wotschke, Pennsylvania, USA
C. Wrathall, Santa Barbara, USA

Vortragsauszüge

P. DEUSSEN: Nine Classes of Parsable Languages obtained by Refining one Abstract Accepting Algorithm

Consider Turing-, L.b.-, Stack or F.S.-Acceptors as Semi-Thue systems Π_Q (with states) and construct an abstract accepting algorithm $\mathfrak{A}(\mathcal{P})$ which works iteratively as follows: while the situation is not yet the final one, $\mathcal{P}$ selects a set of admissible productions in Π_Q from which one is chosen to be applied to the current situation.

Let, furthermore, Π_Q be Π_{LR}, the shift-reduce stack-acceptor, or be Π_{LL} the top-down stack acceptor, which, respectively, are obtained from some context-free grammar Π.

- if $\mathcal{P}$ cuts 'blind alleys' then $\mathfrak{A}(\mathcal{P})$ is partially correct, and determinism holds iff Π is unambiguos
- if $\mathcal{P}$ selects on the basis of k-lookahead thenthe requirement that $\mathfrak{A}(\mathcal{P})$ be deterministic leads directly to the properties of Π being LR(k), SLR(k), LALR(k), LL(k), or simple L(k).

The recursion formulation of $\mathfrak{A}(\mathcal{P})$ leads to the (recursive) backtracking algorithm ARIADNE($\mathcal{P}$) which is partially correct and deterministic under very weak assumptions on $\mathcal{P}$.

- if, additionally, $\mathcal{P}$ cuts 'blind alleys' then ARIADNE($\mathcal{P}$) persues no useless trials
- if $\mathcal{P}$ is such that $\mathfrak{A}(\mathcal{P})$ would be deterministic, then ARIADNE($\mathcal{P}$) does nowhere trace back. Thus ARIADNE($\mathcal{P}$) in this case is the origin of all kinds of "recursively descending" algorithms.

E. BERTSCH: Representation of precedence grammars

We show that precedence functions in the sense of Wirth and Weber are special cases of a general scheme of size reduction for precedence tables.

Applying this notion, we give necessary conditions for compressibility of tables. They are based on inherent redundancies of precedence relations in terms of their corresponding production systems.

Some minor results on the existence or non-existence of particular tables are also presented.

O. MAYER: On deterministic canonical Bottom-up Parsing

A general framework for producing deterministic canonical bottom-up parsers is described which contains all these parsers as special cases. Then conditions on the means of construction are presented which guarantee that the parsing methods work correctly. These conditions cover all known types of deterministic canonical bottom-up parsers.

Finally the problem is regarded to construct for a given grammar a bottom-up parser which meets the conditions mentioned above and is optimal in a certain sense.

G. HOTZ: Eine neue Invariante kontextfreier Sprachen

$G = (V,T,P,S)$ sei eine kontextfreie Grammatik. Eine Abbildung $f: \{G\} \to M$, ist eine Invariante, falls aus der Gleichheit der von G und G' erzeugten Sprachen folgt $f(G) = f(G')$.

Wir fügen zu der bekannten Invariante von Parikh,die leicht berechenbar ist, und der schwer berechenbaren Invarianten "Syntaktisches Monoid" eine weitere von beiden unabhängige Invariante hinzu.

Sei F(V ∪ T) die freie Gruppe, die durch V ∪ T erzeugt wird. Das Produktionssystem P fassen wir als Relationensystem auf und bilden die Gruppe $F(V \cup T)/P =: \mathcal{G}(G)$.
Wir zeigen den Satz: Aus L(G) = L(G') folgt $\mathcal{G}(G) = \mathcal{G}(G')$ der Beweis verwendet die Tietze-Transformationen und ist direkt.

Damit stehen als Invarianten kontextfreier Sprachen alle Invarianten von endlichen Gruppendarstellungen zur Verfügung. Eine Charakterisierung der kontextfreien Sprachen über diese Gruppen ist nicht möglich, da jede endlich darstellbare Gruppe als $\mathcal{G}(G)$ auftreten kann. Es wurde weiter auf den freien Differential-Kalkül von Fox hingewiesen, der für kontextfreie Sprachen schärfere Invarianten zu liefern verspricht.

D. WOOD: A survey of grammar and L form theory

After introducing forms via the unifying concept (c-f) rewriting systems, which specialize under ⇒ (sequential) and ⇛(parallel) rewriting to give cf Chomsky grammars and EOL systems, respectively, the following topics were discussed.Forms under both s(strict)- and g(general)-interpretations were studied.

I. Let τ be a grammatical transformation. For G a gr. form [or L form] is $\mathcal{L}_x(G,\Rightarrow) = \mathcal{L}_x(\tau(G),\Rightarrow)$ [or $\mathcal{L}_x(G,\Rrightarrow) = \mathcal{L}_x(\tau(G),\Rrightarrow)$], where x = s or g.

II. Let $\mathcal{L}$ be a family, define $\mathcal{G}_x(\mathcal{L},\circ) = \{G: \mathcal{L}_x(G,\circ) = \mathcal{L}\}$, where x = s or g $\circ = \Rightarrow$ or $\Rrightarrow$. It can be shown that $\mathcal{G}_g(\mathcal{L}(FIN),\Rrightarrow)$ contains G, a synch. EOL form iff G is finite + non empty.

Gefördert durch DFG Deutsche Forschungsgemeinschaft

III. Equivalence questions. It is still open whether for G_1, G_2 gr. forms, $\mathcal{L}_s(G_1,\Rightarrow) = \mathcal{L}_s(G_2,\Rightarrow)$. On the other hand under g-ints. Ginburg + Spainer have recently shown it to be decidable. For EOL forms, in general, nothing is known.

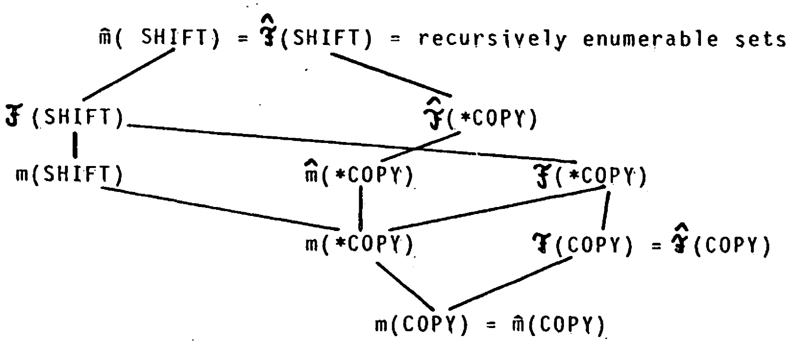
C. WRATHALL: Unbounded Resets on One Tape

Drawing for motivation on the 1978 ICALP paper of R. Book, S. Greibach and C. Wrathall (Udine), we study the semi-AFL, full-semi-AFL, AFL and full-AFL generated by the languages:

$COPY = \{ww \mid w \in \{a,b\}^*\}$, $*COPY = \{(wc)^m \mid m \geq 0,\ w \in \{a,b\}^*\}$ and

$$SHIFT = \{ \langle w_1,w_2\rangle c \langle w_2,w_3\rangle c \cdots c \langle w_m,w_{m+1}\rangle \mid m \geq 0,\ \text{each } w_i \in \{a,b\}^*,\ |w_1| = \ldots = |w_{m+1}|\}$$ where $\langle , \rangle$ is a string-pairing function.

The relationships among the classes are given in the following inclusion diagram



All inclusions shown are proper; if inclusion is not shown the classes are not comparable.
The results represent joint work with S. Greibach.

D. WOTSCHKE: Economy of Description for Probabilistic Automata and Degree Automata

[This is a report on some ongoing research done jointly by Chandra M.R. Kintala, University of Southern California, and Detlef Wotschke, Penn State].

There is a sequence of regular languages $(L_n)_{n\geq 2}$ such that

1) L_n can be accepted by a probabilistic finite automaton (with isolated cutpoints) having only two states,
2) every nondeterministic finite automaton for L_n needs n states.

There is a sequence of regular languages $(L_n)_{n\geq 2}$ such that

1) L_n can be accepted by a n-state Degree-Automaton with cutpoint 1/2,
2) every nondeterministic finite automaton accepting L_n needs at least n^2 (n^k) states.

H. ALT: On the Product of Space and Time Complexity of Turing Machines

The product complexity of a Turing machine is defined as the product of space and time complexity. It is shown that a multitape Tm of product complexity P(n) is simulatable by a 1-tape Tm of time complexity P(n). So lower bounds on the second complexity are lower bounds on the first one. Especially recognition of any nonregular language requires product complexity $\Omega(n \log n)$ and recognition of the language PAL of Palindromes requires product complexity $\Omega(n^2)$. The second result delivers a complexity argument in Algebraic language theory: Beauquier showed that there exists a special generator E_1 of the cone of contextfree languages such that PAL is easily reducable to E_1 and E_1 is easily reducable to any generator. So a language recognizable with product complexity $o(n^2)$ cannot be a generator.

R. BOOK: A New Representation Theorem For PSPACE

The author has shown (in JACM, Jan. 1978) that NP is the smallest class of languages containing all of the regular sets and closed under intersection and polynomial erasing homomorphic replication. Here PSPACE is considered. By considering length-preserving binary relations on strings, the transitive closure of such relations, and the encodings of such relations as languages, we refine the notion of a class of languages being "weakly transitively closed". Thus, PSPACE is the smallest class of languages containing all of the regular sets, this is closed under intersection and polynomial erasing homomorphic replication, and is weakly transitively closed.

B. MONIEN: Connections between the LBA problem and the knapsack problem

It is wellknown that the knapsack problem is NP complete. We will show here that some natural subproblem is NTAPE (log n) complete.

Let us consider the following problem:

input: $a_1,\dots,a_n,b_1,\dots,b_n$, $c \in N$; $a_r \leq n$, $\forall r = 1,\dots,n$

question: Do there exist $r \in N$ and $1 \leq i_1 < i_2 < \dots < i_r \leq n$

such that

$$\sum_{\nu=1}^{r} a_{i_\nu} = c \quad \text{and} \quad \sum_{i_\nu \leq j} a_{i_\nu} \geq b_j \quad \forall j = 1,\dots,n$$

Let us denote by K the set of all binary encodings of knapsack problems of this type having a solution. Then we get:

Theorem 1: K is NTAPE(log n) complete.

(especially $K \in TAPE((\log n)^\alpha)$

$\iff NTAPE(\log n) \subset TAPE((\log n)^\alpha)$)

From Theorem 1 we can easily derive as a corollary

Theorem 2: There exists a language L_0 which is NTAPE(log n) complete and which is accepted by a nondeterministic 1-way 1-counter automaton which tests for every input string during its whole computation at most once whether its counter is equal to zero.

Theorem 2 improves a former result of Z. Galil, H. Sudborough and myself which states the same conclusion for ordinary n.d. 1-way 1-counter automata (allowing an unbounded number of tests).

M. JANTZEN: Properties of Petri net languages

Let $\mathcal{L}_0, \mathcal{L}_0^\lambda$ be the families of Petri net languages in the sence of Hack. Let D_1 be the semi Dyck language over $\{a,\bar{a}\}$. Let $\mathcal{CSS}$ be the family of computation sequence sets (Peterson) then

$$\mathcal{L}_0 = \mathcal{M}_\cap(D_1-\{\lambda\}), \quad \mathcal{CSS} = \mathcal{M}_\cap(D_1), \quad \mathcal{L}_0^\lambda = \hat{\mathcal{M}}_\cap(D_1)$$

We show

$$\mathcal{M}_\cap(D_1) \subsetneq \hat{\mathcal{M}}_\cap(D_1)$$

An we define a family $\mathcal{K} \subsetneq \mathcal{M}_n(D_1)$ such that

$\mathcal{L}_0 = \mathcal{H}^{-1}(\mathcal{K})$, i.e. $\mathcal{K}$ is a family of "hardest" languages.

We discuss some of the open problems in the field of Petri net languages.

A.K. JOSHI: Some Results in Mathematical Linguistics

We will describe some of our work in mathematical linguistics. In particular, we will discuss two topics and some of their applications.

1. Local transformations: A well-known result of Peters and Ritchie is that if context-sensitive rules are used for "analysis" only, then the string language of the set of trees is still context free. We have generalized this result by considering context-free rules constrained by Boolean combinations contextual predicates (called "local transformations"). It can be shown that the Peters-Ritchie results can be extended to local transformations. Formal language applications and linguistic applications will be discussed.

2. Skeletal sets: We explore the idea of characterizing sentences by the shapes of their structural descriptions only, e.g. in the case of context-free grammars by the shapes of the derivation trees only (with no labels for the nodes). Such structures (called "skeletons") exhibit all the grouping structure without naming the syntactic categories. We will discuss mathematical properties of skeletons relating them to local sets, and skeletal rewriting systems.

Some results relating the above two topics to the syntax and semantics of programming languages will also be presented.

M.A. HARRISON: Iteration Theorems for Various Classes of Deterministic Contextfree Languages

A number of iteration theorems have been obtained for important families of deterministic languages. For example, for the families of strict deterministic languages, simple languages, LL(k) languages, etc. A key part of the method is to develop "Left-Part Theorems" which are characterizations of the trees of the grammar class. In the same way the main theorem proven here is obtained:

Theorem: Let L be a strict deterministic language of degree n. There exists an integer p such that, for each $w \in L$ and each set K of p or more positions in w, there is a factorization $\varphi = (w_1,w_2,w_3,w_4,w_5)$ of w such that

1. $w_2 \neq \Omega$
2. if $K/\varphi = \{K_1,\ldots,K_5\}$ then
   (i) either $K_1,K_2,K_3 \neq \emptyset$ or $K_3, K_4, K_5 \neq \emptyset$,
   (ii) $|K_2 \cup K_3 \cup K_4| \leq p$
3. for each $k, m \geq 0$, $u \in \Sigma^*$, $w_1w_2^{k+m}w_3w_4^k u \in L$ if and only if $w_1w_2^m w_3 u \in L$
4. for each $u_1,\ldots,u_{n+1} \in \Sigma^*$, if $w_1w_2^{n_i}u_i \in L$ for i, $1 \leq i \leq n+1$ where each $n_i \geq n$, then there exist $1 \leq i < j \leq n+1$, $1 \leq r \leq n_i$, $1 \leq r' \leq n_j$ and factorizations $\xi = (v,x,y,z)$ and $\xi' = (v',x',y',z')$ of u_j such that
   (i) for all $m \geq 0$, the following are all in L

$$w_1w_2^{(n_i-r)+mr}vxy^m z, \qquad w_1w_2^{(n_j-r')+mr'}v'x'y'^m z'$$

$$w_1w_2^{(n_i-r)+mr}v'xy^m z, \qquad w_1w_2^{(n_j-r')+mr'}vx'y'^m z'$$

   (ii) none of w_3,v,v' is a proper prefix of any of w_3,v,v'.

R. KEMP: Analysis of algorithms recognizing Dycklanguages

We regard two algorithms for the recognition of a Dyckword w of length 2N and show the following results:

Algorithm 1: The recognition by a stack (implemented on a pda)

(a) The average time complexity is: 4N

(b) The average space complexity is $\sqrt{\pi N} - 0.5 + O(\frac{\log(N)}{\sqrt{N}})$

(= the average maximum size of the stack used in algorithms that traverse a binary tree with N+1 leaves from left to right)

Algorithm 2: The recognition by an algorithm due to G. Hotz and J. Messerschmidt

i) Implementation on a two-way-one-counter-automaton

(a) The average time complexity is: $4N\sqrt{\pi N} - 6N + 4.5\sqrt{\pi N} + O(\frac{1}{\sqrt{N}})$

(b) The average space complexity is for all $\alpha > 0$:

$$\mathrm{ld}(\sqrt{N}) + C + F(N) + O(\frac{\log(N)}{N^{\alpha}})$$

where C=1.82574... is a constant and F(N) is an oscillating function with F(N)=F(4N) and -0.574 F(N) -0.492.
(= the average number of registers needed to evaluate a binary tree with N+1 leaves optimally)

ii) Implementation on a two-tape off-line Turing machine

(a) The average time complexity if: $(12N+13.5)\sqrt{\pi N} - 8(N+1)\log(4\pi\sqrt{N}) + O(N)$ (all terms in O(N) up to $O(N^{-\varepsilon})$ are computed ($\varepsilon > 0$))

(b) The average space complexity is comp. i) (b)

H. EHRIG: Parallelism and Concurrency of Graph Derivations and Applications to a very small Data Base System

Given a derivation $G \overset{p}{\Rightarrow} H \overset{p'}{\Rightarrow} X$ of graphs in a graph grammar we ask for conditions such that productions p and p' can also be applied in opposite order $G \overset{p'}{\Rightarrow} H' \overset{p}{\Rightarrow} X$. Actually there is a non-trivial condition, called independence of $G \overset{p}{\Rightarrow} H$ and $H \overset{p'}{\Rightarrow} X$, to solve this problem. Moreover in this (and only this case) we can apply the "parallel" production p + p' to G leading to a direct derivation $G \overset{p+p'}{\Rightarrow} X$ (parallelism theorem). But also in the case that $G \overset{p}{\Rightarrow} H$ and $H \overset{p'}{\Rightarrow} X$ are not independent it is possible to define a "concurrent" production $p *_R p'$ (R is the relation between both derivations) such that we obtain a direct derivation $G \xrightarrow{p *_R p'} X$ and vice versa leading to a bijective correspondence. (concurrency theorem). Both results are obtained using pushout techniques for a 3 resp. 4-dim. cube, finally applications to synchronisation in a very small data base system are discussed.

J.-F. PERROT: Varietäten von rationalen Sprachen

Varietäten sind hier im Sinne von Eilenberg zu verstehen und werden vom Standpunkt der Operationen Stern (*), Konkatenation und "Shuffle" untersucht.

Hauptergebnis: Die einzige Varietät, die bezüglich Stern abgeschlossen ist, ist die Varietät sämtlicher rationalen Sprachen. (J.-F. Perrot, J.-E. Pin: erscheint in Theoretical Comp. Sc. 1978)

Probleme: - Ob eine Varietät existiert, die keine "Gruppenvarietät" wäre, aber bezüglich Konkatenation abgeschlossen.
- Ob eine nicht kommutative Varietät bezüglich "Shuffle" abgeschlossen sein kann.

H. JORGENSEN: Unscharfe Dominanz

Die üblichen Präzisierungen der linguistischen Begriffe Dominanz und syntaktische Äquivalenz sind aus den folgenden Gründen oft nicht adäquat:

(a) Die Relationen sind häufig nur schwer oder gar nicht entscheidbar.

(b) Eine Sprache ist im allgemeinen nicht als die Menge aller korrekten Sätze gegeben, sondern z.B. als ein Textkorpus, das in Wirklichkeit nur eine Stichprobe darstellt.

(c) Die syntaktischen Phänomene, insbesondere die erlaubten Kontexte von syntaktischen Einheiten, sind eventuell von unterschiedlicher Bedeutung für die Bestimmung der syntaktischen Funktionen.

Um diese Einwände berücksichtigen zu können, wird eine Modifikation der üblichen Präzisierungen der genannten Begriffe untersucht.

J. SAKAROVITCH: Algebraische Methoden für Kellerautomaten

Soit $A = \langle X, Q, \Gamma, \delta, (q_0, y_0)\rangle$ un automate à pile classique, mais sans états terminaux. Pour chaque état q soit L_q un langage sur l'alphabet de pile Γ. Le language reconnu par l'automate A et la famille $\{L_q\}$ est alors ainsi définie:

$$L(A, \{L_q\}) = \{ w \mid (w, q_0, y_0) \overset{*}{\underset{A}{\vdash}} (1, q, w),\ w \in L_q \}$$

Si $L_q = \Gamma^*$ pour les états q dans un certain sous ensemble Q' de Q et $L_q = \emptyset$ pour les q dans le complémentaire de Q', on retrouve le processus classique d'acception par état terminal.

Il est bien connu que si L_q est un langage rationnel pour chaque q, L(A,{L_q}) est algébrique et que si, de plus, A est déterministe , L(A,{L_q}) est algébrique déterministe (Ginsburg & Greibach, 1966).

On montre alors

Théorème 1: Si L_q est algébrique pour chaque q, alors L(A,{L_q}) est un langage algebrique.

Théorème 2: Si A est un automate à pile deterministe et si L_q est algebrique non-ambigu pour chaque q, alors L(A,{L_q}) est un langage algébrique non ambigu.

Ces deux théorèmes ne font appel ni à la construction d'automates, ni à celle de grammaires ad hoc, mais sont fondés sur la représentation, donnée par Nivat (1966), des automates à pile par des transduction du monoide libre dans le monoide polycyclique.

G. BARTH: Dynamic Syntax Specification

Grammars with dynamic control sets, called recording grammars (rgs, for short),will be introduced. Like grammars with rule modification [Hanford/Jones] and grammar forms with varying interpretation [Ginsburg/Rounds], rgs adjust their generative behaviour to specific inputs. Rgs can be viewed as combinations of state grammars [Kasai] and grammars with static control sets [Ginsburg/Spanier, Salomee]. Like state grammars they act during derivations in one of three possible states, 'n' (normal), 'r' (recording) or 'd' (directed). Rules applied in state r are remembered by concatenating their labels, thereby dynamically producing control strings, which in state d are used to direct derivations. Results on generative capacity of rgs, an interesting 'substring-relationship-property' and on closure and decidability properties of rgs will be given in the lecture. Furthermore it will be demonstrated that rgs are a convenient tool to formalize context-sensitive features in programming languages.

H. WALTER/J. PFLÜGER: Affinity of contextfree grammars

The lecture give a report on the work about affinity of contextfree grammars, done by J. Pflüger, P. Ochsenschläger and H. Walter.

A grammar morphism $\varphi: G_1 \to G_2$ is a pair of mappings

$\varphi_A : A(G_1)^* \to A(G_2)^*$, $\varphi_P : P(G_1) \to \underline{S}(G_2)$

satisfying certain compatibility conditions.

Certain special types of grammar morphismus are considered:

$\varphi : G_1 \to G_2$ is a transformation iff

$\varphi D(G_1) = D(G_2)$ & $\varphi(t) = t \quad (t \in T(G_1))$.

$\varphi: G_1 \to G_2$ transformation.

φ is a reduction iff $\varphi P(G_1) \leq P(G_2)$.

A simulation is a special transformation which is the identity on the variables and has uniquely identifiable variables for the "simulation" of every rule.

Let $\mathcal{P}$ a property of (contextfree) grammars, $\mathfrak{N}$ a property of grammar morphismus.

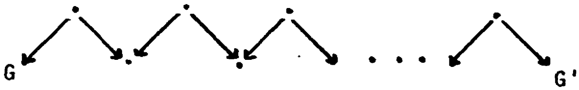
$(\mathcal{P},\mathfrak{N})$ defines an affinity iff

(1) $\varphi_1, \varphi_2 \in \mathfrak{N} \Rightarrow \varphi_1 \circ \varphi_2 \in \mathfrak{N}$

(2) $G \in \mathcal{P}$, $\varphi: G \to G'$ internal isomorphism $\Rightarrow G' \in \mathcal{P}$ & $\varphi \in \mathfrak{N}$

Then define

$G \sim G'(\underline{\mathcal{P}}_{\mathfrak{N}})$ iff there exists a diagram



where all "." $\in \mathcal{P}$ & all "→" $\in \mathcal{O}$.

Notation:

1. $G \sim G' \iff \mathcal{L}(G) = \mathcal{L}(G')$

2. $G \approx G' \iff \mathcal{L}(G^{()}) = \mathcal{L}(G'^{()})$ (parenthesized languages)

<u>Theorem 1:</u>

$G \approx G' \iff G \sim G'$ ($\underline{cf}_{\underline{Reduction}}$)

<u>Theorem 2:</u>

$G \sim G'$($\underline{cf}_{\underline{Reduction}}$) $\Rightarrow$ ind(G) = ind(G')

(ind G = total Brainerd index)

<u>Corollary:</u>

$G \approx G' \Rightarrow$ ind(G) = ind(G')

The further investigation of affinity is based on the following Lemma:

<u>Lemma:</u>

Every transformation φ can be factorised in a simulation and a reduction φ_2, that is $\varphi = \varphi_2 \circ \varphi_1$.

It delivers some other properties of transformations and simulations and we get as

<u>Corollary:</u>

$G \sim G'$ ($\underline{cf}_{\underline{sim}}$) is decidable.

As an application to show the capability of transformation may serve the following theorem:

<u>Theorem 5:</u>

Let G_B with $L_B = \mathcal{L}(G_B)$ a bracketed grammar (as in Ginsburg, Harrison, 1967) (or a parenthesized grammar), and let G be an arbitrary, reduced contextfree grammar.

Then we have:

$G_B \sim G \iff G_B \sim G\ (\underline{cf}_{trans})$.

From that we get an easy decidability procedure.

C. REUTENAUER: Rationale Potenzreihen und endlich-dimensionale Algebren

Analog zum Beispiel vom syntaktischen Monoid einer formalen Sprache ist der Begriff der syntaktischen Algebra einer Potenzreihe. So läßt sich der Satz von Kleene-Schützenberger in folgender Weise formulieren: Eine Potenzreihe ist genau dann rational, wenn ihre syntaktische Algebra endlich dimensional ist. Man kann dann Mannigfaltigkeiten von endlich dimensionalen Algebren und von rationalen Potenzreihen einführen und zeigen, daß eine bijektive Beziehung zwischen ihnen besteht, wie S. Eilenberg es für Mannigfaltigkeiten von rationalen Sprachen und endlichen Monoidenbewiesen hat. Andere Eigenschaften werden noch bewiesen.

J. GOLDSTINE: A New Look at Abstract Families of Languages (AFL's)

A pushdown automaton can be thought of as the rational language consisting of all sequences of moves (input-stack-transformation pairs) allowed by its finite state control. In this way pda's can be represented in many notational schemes and they can be easily manipulated.

Generalizing this idea one may define a data store by specifying the action of a set I of instructions on a set D of storage configurations. A data-store automaton is then a rational language over the alphabet of (input, instruction) pairs. A simple development of AFL-theory can be obtained, which stresses the correspondence between language operations and automata. In addition the theory of rational languages can be used to simplify the manipulation of automata since automata are defined to rational languages.

I.H. SUDBOROUGH: Languages defined by extended precedence grammars

It is known that every deterministic CFL can be generated by some uniquely invertible (2,1)-precedence grammar and that there are deterministic CFL's that cannot be generated by any UI (1,k)-precedence grammar, for $k \geq 1$. In this paper we consider extended operator precedence grammars. It is shown that: (1) there are (k+1,1)-operator precedence languages that are not (k,1)-operator precedence, (2) there are (1,k+1)-operator precedence languages that are not (1,k)-operator precedence, and (3) that (2,1)-operator precedence languages are contained in DSPACE(log n) if and only if every DCFL is in DSPACE(log n). Furthermore, the question of whether (1,k)-precedence grammars can be converted into equivalent simple precedence grammars is discussed.

M. NIVAT: Adherence of algebraic languages

Let X^{ω} denote the set of infinite words on X, $u \in X^{\omega}$ is a mapping of $N_{+} = N \setminus \{0\}$ into X. Let $X^{\infty} = X^{*} \cup X^{\omega}$.

For $f \in X^{*}$ denote f(n) the n^{th} letter of f if $|f| \geq n$ otherwise define $f(n) = \Omega$. For $n \in X^{\omega}$ u(n) is defined for all n.

On X^{∞} we consider the following metric distance

$$d(\alpha,\beta) = \frac{1}{\min\{n \mid \alpha(n) \neq \beta(n)\}} \qquad \text{if } \alpha \neq \beta$$

$$d(\alpha,\beta) = 0 \quad \text{if } \alpha = \beta.$$

For $L \subset X^*$ the adherence of L, according to the topology induced by this metric is $Adh(L) = \{ n \in X^\omega \mid \forall f \in X^* \; f < n \Rightarrow \exists g : fg \in L\}$

We prove several results about sets of infinite words, or w-sets, which are adherences of algebraic subsets of X^*. This work is motivated by semantic considerations which will be given too.

A. NIJHOLT: Transformations on Non-Left-Recursive Grammars

A new transformation from non-left-recursive to Greibach normal form (GNF) grammars is introduced. It can be shown that this transformation (the 'left part transformation') when applied to a strict deterministic grammar yields a GNF grammar which is also strict deterministic. Moreover, when it is applied to a simple chain grammar the GNF grammar is a simple deterministic grammar. Structure preserving properties, given in terms of grammar covers, are presented. Cover properties for different types of generalized left corner parses can be obtained. In the case that the input grammar is both non-left-recursive and non-right-recursive it can be shown that with this transformation grammar covers can be defined in such ways that left parses of the output grammar can be mapped on right parses of the input grammar and moreover, that right parses of the output grammars can be mapped on right parses of the input grammar.

H. Alt (Saarbrücken)