Jean Berstel, Jean-Eric Pin, Wolfgang Thomas
(editors)

Automata Theory and Applications in Logic and Complexity

Dagstuhl-Seminar-Report; 5
14.-18.1.1991 (9103)

ISSN 0940-1121

Das Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloß Dagstuhl
Informatik, Bau 36
Universität des Saarlandes
W - 6600 Saarbrücken
Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: dagstuhl@dag.uni-sb.de

Report on the Dagstuhl-Seminar

"Automata Theory with Applications in Logic and Complexity"

(14.1. - 18.1.1991)

Organizers: J. Berstel (Paris), J.E. Pin (Paris), W. Thomas (Kiel)

The aim of this seminar was to work on and further develop promising connections between automata and formal language theory to neighbour fields as well as to application areas. Since (project-) research in theoretical computer science today is split into many rather special topics and methods, it seemed worthwile to invite scientists working in the intersection of automata theory, logic, and complexity theory for an exchange of questions and results.

Despite of some difficulties due to the rather short preparation of the seminar, the reaction to the invitations was very positive. The date early in January was difficult to meet especially by scientists in the US, Canada, and Israel, which had an effect on the planned seminar topic "circuit complexity". But the 27 participants who finally could come covered quite a wide range of themes, which nevertheless was so specific that the invited scientists had an interest in the entire program of talks (and not just part of it). All talks were attended by virtually all participants, and - as hoped - the discussions resulted in stimulations across the borders of more specialized fields.

It is difficult to classify the contributions into the sections automata theory, theory of formal languages, logic, and complexity. In many talks these aspects were so much intertwined that such a division would be artificial. It might be more appropriate to refer to an orthogonal classification distinguishing by means of the types of objects under consideration (words, trees, graphs, etc.). Here the original objective of automata and language theory, the analysis of word properties or word languages, was extended to several more general types of languages: ω-languages, tree languages, trace languages, picture languages, as well as graph languages. All these fields were represented by talks during the seminar, with different methodological emphasis (towards complexity, logic, automata, combinatorics, or algebra). The following explanations give a rough summary; more informations are found in the abstracts of the individual talks.

Automata and formal languages in the narrower sense were treated in the talks of J.

Karhumäki, A. Restivo, and D. Perrin, mainly aiming at efficient algorithms for decision problems: In particular, the topics were test sets and their efficient computation, effective decision of language properties (e.g. having a given star-height), and representability of automata by means of given transition graphs.

Several talks were concerned with different issues in complexity, part of them connected with combinatorial problems: The distinguishability of words by automata with a given number of states was analyzed (M. Robson), a new classification of problems solvable in polynomial time (in terms of counting words of given length) was presented (M. Goldwurm), and a connection between cardinality conditions and regularity for word sets was established (D. Krob). The talk of M. Goldwurm already overlapped with "structure in complexity", a field which was also represented by the contributions of R. Book and K. Compton. While R. Book gave intrinsic characterizations of some central complexity classes, K. Compton presented a description of the circuit complexity class AC^0 in terms of language theory and first-order logic.

A central subject of the seminar was the theory of tree languages (with talks by G. Hotz, A. Podelski, H. Seidl, M. Steinby, W. Thomas, and S. Tison). Here we saw a nice combination of the algebraic approach with automata theory and mathematical logic. M. Steinby presented a framework for developing a "variety theory of tree languages". H. Seidl investigated a generalized model of tree automaton (in which cost functions in semirings are evaluated). The talks of G. Hotz and A. Podelski dealt with the links between word and tree languages (e.g. a set-up for introducing tree monoids). The contributions of W. Thomas and S. Tison treated connections between tree automata and a logical definition of tree properties (decidability of the theory of ground term rewriting systems, and automata for path logics over trees).

Logical notions also furnished the background for the talks on ω-languages and graph properties. L. Priese classified different fairness notions by recursion theoretic (and topological) conditions on nonterminating computations (represented by "ω-traces"). L. Staiger compared several possibilities to pass from word languages to ω-languages, and worked out the differences for the cases of regular, context-free, and recursive languages. The contributions of B. Courcelle and M. Mosbah showed how the definability of graph properties in monadic second-order logic is linked with existence of polynomial algorithms deciding these properties.

Two generalizations of language theory that have recently attracted much attention are the theory of picture languages and that of trace languages in the sense of Mazurkiewicz (or subsets of partially commutative monoids). Six lectures were devoted to this area. P. Séébold and F.J. Brandenburg presented new results on the simplification of picture description words by suitable transformation rules. C. Choffrut analyzed conjugation and commutation laws in monoids. S. Varricchio used methods from the theory of formal power series to solve equivalence problems on sets in partially commutative monoids. Sets of infinite traces were treated by L. Priese (as mentioned above) and by V. Diekert. Diekert's contribution solved the intriguing problem how sets of infinite traces could be equipped with the common laws of concatenation.

This short summary already shows that the seminar integrated rather different approaches and produced a unified view on many questions arising in automata theory, logic, and complexity theory. Especially remarkable was the contribution by young researchers (V. Diekert, M. Goldwurm, D. Krob, M. Mosbah, A. Podelski, P. Séébold, H. Seidl, S. Tison, S. Varricchio).

Besides the "official" program of talks, many discussions and smaller meetings took place: For example, J.E. Pin reported in a long evening session on a new (algebraic) construction of deterministic ω-automata and thus proposed a new approach to the core of the theory of ω-languages. Also new cooperations between participants were started. As a small example it can be mentioned that a ten year old question by the present writer (concerning an example for the generalized star-height problem) was now settled by M. Robson in an e-mail correspondence following the seminar.

Summarizing, the seminar has provided, by a varied program of talks of high quality as well as by the informal discussions accompanying it, a valuable stimulus to the current work of the participants. This was supported considerably by the pleasant atmosphere in the house and and by the fact that the staff, which proved to be a most friendly and competent team, worked already very smoothly and efficiently. We also got good hints for a nice Wednesday afternoon excursion (even with sunny January weather). The organizers would like to take this opportunity to thank the staff, as well as the members of the Dagstuhl office in Saarbrücken, for their efficient and pleasant cooperation.

Kiel, 4.5.91

W. Thomas

(Prof. Dr. W. Thomas)

Vortragsprogramm

Montag, 14.1.91

R. Book (St. Barbara/Würzburg): Intrinsic Characterizations of Certain Classes of Languages
L. Priese (Paderborn): Fairness and Traces
M. Steinby (Turku): Varieties of Regular Tree Languages
D. Perrin (Paris): Synchronizing Words and Automata, and the Road Coloring Problem
B. Courcelle (Bordeaux): Expressing Graph Properties in Monadic Second-Order Logic. Applications to the Theory of Graph Grammars and to the Complexity of Graph Algorithms
M. Mosbah (Bordeaux): Monadic Second-Order Evaluations on Tree-Decomposable Graphs

Dienstag, 15.1.91

J. Karhumäki (Turku): Efficient Constructions of Test Sets for Regular and Context-Free Languages
H. Seidl (Saarbrücken): Tree Automata with Cost Functions
S. Tison (Lille): Tree Automata and Rewrite Systems
J.M. Robson (Canberra): Short Strings not Distinguished by Small Automata
P. Séébold (Paris): Minimizing Picture Words
F.J. Brandenburg (Passau): Cycles in Picture Words: Elimination and Generation

Mittwoch, 16.1.91

A. Restivo (Palermo): FTR Languages of Star-Height One
D. Krob (Rouen): On Length Distributions of Rational Languages
S. Varricchio (L'Aquila): Some Decidability Results in Formal Series Theory

Donnerstag, 17.1.91

K. Compton (Ann Arbor): A Characterization of Regular Languages Recognized by Constant Depth Circuits
M. Goldwurm (Mailand): On Counting and Instance Problems Defined Over Formal Languages
V. Diekert (München): Recognizability of Complex Trace Languages
G. Hotz (Saarbrücken): Defining Languages by Interpreting Tree Languages
A. Podelski (Paris): Monoids of - and Automata Over - Finite and Infinite Trees

Freitag, 18.1.91

Ch. Choffrut (Paris): Conjugacy
L. Staiger (Aachen): ω-Power and δ-Limes of Formal Languages
W. Thomas (Kiel): Stratified Tree Automata

Zusammenfassungen der Vorträge

Intrinsic Characterizations of Certain Classes of Languages

R. V. Book (St. Barbara)

A characterization of a class of languages is intrinsic if it does not use the technical machinery involved in the definition of the class. For example, L is context-free iff there exist a regular set R and homomorphisms h_1, h_2 such that $L = h_1 (h_2^{-1} (\Delta_2) \cap R)$. More pertinent here is the following:

$$L \in BPP \text{ iff } \mathrm{Prob}_A (\{ A \mid L \leq_T^P A \}) = 1 .$$

We have the following:

Theorem: For an arbitrary language L,

(a) $L \in P$ iff $\mathrm{Prob}_A (\{ A \mid L \leq_{logn-T}^{P} A \}) = 1$;

(b) $L \in NP$ iff $\mathrm{Prob}_A (\{ A \mid L \leq_{logn-tt}^{NP} A \}) = 1$.

Corollary:

(a) $P = NP$ iff $\mathrm{Prob}_A (\{ A \mid SAT \leq_{logn-T}^{P} A \}) = 1$;

(b) $P = NP$ iff $\mathrm{Prob}_A (\{ A \mid P_{logn-T} (A) = NP_{logn-tt} (A) \}) = 1$.

In contrast to Corollary (b), it is known (5th IEEE Conf. on Structure in Complexity Theory) that Bennett and Gill's separation of P(A) und NP(A) for random A does not speak to the question of whether SAT is in P.

Fairness and Traces

L. Priese, D. Nolte (Paderborn)

We show how to transform some fairness results for transition systems (the standard model for operational interleaving semantic) to models with true concurrency. As such a model we use traces here. Three different notions of infinite traces are introduced (following P. Gastin, B. Rozoy and V. Diekert):

$$T_1(\Sigma, D) := \Sigma^\infty /_{\approx 1} , \; T_2(\Sigma, D) := T_1(\Sigma, D) \cup \{ \perp \} , \; T_3(\Sigma, D) := \Sigma_\delta^\infty /_{\approx 3}$$

where in Σ we also operate with 'blocked' letters, δ_a, telling that $a \in \Sigma$ is blocked. We are interested in formal systems, P, with a true concurrency semantic presented by an interpretation of P into sets of i-traces: $J(P) \subseteq T_i(\Sigma, D)$, $1 \leq i \leq 3$.

Thus, $t \in J(P) \cap T_i (\Sigma, D)$ reads that t is an i-trace "which forms a computation of P". While $t \in J(P) \cap T_i (\Sigma, P)$ is canonically a Π_1^0 predicate for i = 1 and reasonable interpretations J and formal systems P, this isn't true for i = 2 or 3. However, we only need $t \in J(P) \wedge T_i (\Sigma, D)$ to be a Π_3^0 - predicate to apply our general fairness results (P. Darondeau, D. Nolte, L. Priese, S. Yoccoz):

$\forall M \subseteq N^\omega$ there are equivalent:

- M is Π_3^0
- M contains only fair computations (of some transition system)
- $f \in M \Leftrightarrow f = \lim^d{}_{n \to \omega} f \mid n$ for some Π_1^0 ultrametric d.

Thus, being a 'fair' computation of P adds to a trace t just another Π_3^0 -condition. As a result there are equivalent:

- t is a fair i-trace in J(P) ($t \in J^{fair}(P)$)
- $J^{fair}(P)$ is Π_3^0
- $t = \lim^d{}_{n \to \omega} t[n]$ for some Π_1^0 ultrametric d.

Varieties of Regular Tree Languages

M. Steinby (Turku)

The theory of varieties of tree languages (VTLs) is proposed as a general framework for the study of families of regular tree languages. As a counterpart to an extended version of Eilenberg's variety theorem, it can be shown that each VTL $\mathcal{V}$ is matched by a variety $\mathcal{V}^a$ of finite algebras and a variety $\mathcal{V}^c$ of finite congruences (when these notions are suitably defind). The congruences are convenient for defining a VTL, while the variety $\mathcal{V}^a$, being generated by the syntactic algebras of the members of $\mathcal{V}$, gives a useful description of $\mathcal{V}$. As examples of VTLs we mention the families $\mathcal{Nil}$ (finite and cofinite tree languages). $\mathcal{D}$ (definite tree languages), $\mathcal{RD}$ (reverse definite tree languages), $\mathcal{GD}$ (generalied definite tree languages) and $\mathcal{Loc}$ (local tree languages).

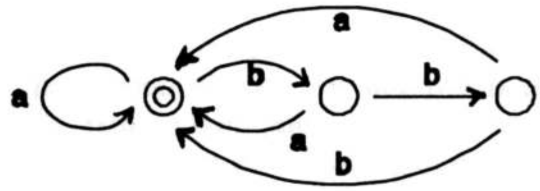
Synchronizing Words and Automata, and the Road Coloring Problem

D. Perrin (Paris)

[joint work with M.P. Schützenberger]

The road coloring problem, posed by Adler, Goodwin and Weiss in 1977, asks whether it is always possible to label the edges of an aperiodic connected graph with constant outdegree to make a synchronizing automaton.

We solve the particular case of a graph with all vertices except one having indegree 1 (see example below)



The proof uses a theorem of C. Reutenauer (1985) which gives a factorization of the noncommutative polynomial of a finite maximal code. When the code is not synchronizing, the factorization is not trivial.

Expressing Graph Properties in Monadic Second-Order Logic

B. Courcelle (Bordeaux)

Considering a graph as a logical structure makes possible to express its properties by logical formulas. Monadic second-order logic is powerful enough to express a large number of interesting graph properties (connectivity, planarity, Hamiltonicity) while enjoying general decidability properties (the monadic theory of a context-free HR set of graphs is decidable). Its role in the theory of graph grammars is comparable to that of regular languages in classical language theory.(1)

We have presented two uses of monadic second-order logic: A method allowing to construct a graph reduction system that decides in linear time whether a given graph belongs to a set of graphs that is both definable in monadic second-order logic and is if bounded tree-width (i.e., is a subset of some context-free HR set of graphs). This gives a generalization of the classical methods of recognition of trees by elimination of pendant edges, and of series-parallel graphs by elimination of parallel edges and series-edges.

We have also presented some graph transductions that can be specified in monadic-second order logic. They generalize rational transductions, "yields" (frontier mappings), and other transformations of words and trees that are very useful in language theory. One obtains with them intrinsic characterizations of HR and C-edNCE sets of graphs (i.e., of the two families of context-free sets of graphs). (2)

(1) Coauthors: S. Arnborg, A. Proskurowski, D. Seese
(2) Coauthor: J. Engelfriet

Works supported by the ESPRIT project 3299 "GRAGRA" (Computing by Graph Transformations).

Monadic Second-Order Evaluations on Tree-Decomposable Graphs

B. Courcelle, M. Mosbah (Bordeaux)

Every graph generated by a hyper-edge replacement graph grammar can be represented by a tree, namely the derivation tree of the derivation sequence that produced it. Certain functions on graphs can be computed recursively on the derivation trees. By using monadic second-order logic and semi-ring homomorphisms, we describe a class of such functions in a single formalism. Polynomial and even linear algorithms can be constructed for some of these functions.

Let φ be a monadic second-order formula with set w of free variables. For every graph G, we denote by sat $(\varphi)(G)$ the set of w-assignments in G that satisfy the formula φ. It is known that sat $(\varphi)(G)$ can be evaluated bottom-up on any derivation tree of G (Courcelle). If h is a homomorphism, then we define an evaluation V by $V(G) = h(sat(\varphi)(G))$. The mapping V can be evaluated directly bottom-up, without needing the costly computation of sat $(\varphi)(G)$.

We extend and unify results obtained by Takanizawa et al., Bern et al. (81), Arnborg et al. (88) and Habel et al. (89).

Efficient Constructions of Test Sets for Regular and Context-Free Languages

J. Karhumäki (Turku)

(joint work with W. Rytter and S. Jarominek (Warsaw)

A test set of a language is its finite subset such that in order to test whether abitrary two morphisms are equivalent on all words of the language it is enough to test this on this finite subset. It is that each language possesses a test set (known as Ehrenfeucht Conjecture before it was proved). For regular and CF languages such test sets can be found effectively, however the sizes of those were very large: Exponential and double exponential, respectively. Here we prove that

- For regular languages a linear size test set exists and can be found in time $O(n^2)$ where n is the number of transitions of the automaton.
- For context-free languages test sets of (essentially) exponential size can be found effectively.

Moreover, we give

- An O(n logn) - time algorithm for the problem "morphic equivalence on regular languages", and
- An $O(n^2 \log n)$ - time algorithm for the equivalence of two deterministic gsm's.

Tree Automata with Cost Functions

H. Seidl (Saarbrücken), A. Weber (Frankfurt/M.)

A cost function c for a finite tree automaton A is a mapping from the transition relation of A into $\{0,1\}$. Two different notions c_+ and $c_\sqcup$ generalize the notion of "distance" for word automata. Cost $c_+(\varphi)$ of a computation φ is the sum of costs of occurring transitions; cost $c_+(t)$ of a tree t is given by

$$c_+(t) := \sqcap\{c_+(\varphi) \mid \varphi \text{ accepting computation for } t\};$$

whereas the cost of A is

$$c_+(A) := \sqcup\{c_+(t) \mid t \in L(A)\}.$$

In contrast, $c_\sqcup(\varphi)$ of a computation φ gives the maximal sum of costs occurring on a path in φ. $c_\sqcup(t)$ and $c_\sqcup(A)$ are defined analogously to $c_+(t)$ and $c_+(A)$. In case the ambiguity of A is finite, we are able to prove:

1) $c_+(A) < \infty$ iff $c_+(A) < 3^n \sqcup L^r$

2) $c_\sqcup(A) < \infty$ iff $c_\sqcup(A) < 3^n + 2^{n(n+1)}$

where L is the maximal rank of input symbols and $r = 3^n$ where n is the maximal number of states of A.

These results are extended to the case where the costs are polynomials chosen from $N_0[X]$ or $R[X]$ of degree at most 1 where R is the semiring $R = (N_0 \cup \{-\infty\}, \sqcup, +)$.

Tree Automata and Rewrite Systems

M. Dauchet and S. Tison (Lille)

Our approach is to use tree automata technics to solve problems in term rewrite systems. We try to show its successes and its limits.

1) The main result is: The theory of ground rewrite systems is decidable: We define the first-order language for ground (i.e. without variables) rewrite systems where constants are ground terms and predicates are associated with rewrite relations defined by ground systems, and we prove the decidability of this theory. Particularly, we get decision procedures for classic properties. The proof is standard: we associate with any formula an automaton.

2) We sketch some other problems. Particularly, we ask a question: May we decide whether the set of normal forms of a rewrite system is recognizable?

3) We introduce a new class of tree automata, by adding tests of equality (and $\neq$) to usual tree automata. With some restrictions on these tests, we obtain a "good" family:

Closure under boolean operations and alphabetic morphism, good decision properties. E.g., we can use these automata for some decidability problems in term rewrite systems where the variables which occur twice occur at depth 1.

Short Strings not Distinguished by Small Automata

T. M. Robson (Lyon)

This paper considers the question of what are the shortest two strings over a binary alphabet, not separated by any finite automaton with n states. An answer to this question would answer also the converse question of how many states are required for a finite automaton to distinguish two strings of length L.

A restricted case where the state transition functions of the automaton are restricted to permutations is considered, partly in the general hope of obtaining insights useful in answering the general question, and more specially in the expectation that there will be a reduction from the general question to the restricted case.

For n from 2 to 5 the shortest strings in the restricted case have been found and, except for n = 2, they are shorter than those known to exist even in the single letter alphabet case (4, 11 and 32 instead of 6, 12 and 60).

A reduction from the general case to the restricted one is shown, such that a pair of strings with length $\leq L$ not distinguished by n-state permutation machines implies a pair with length $O(L\Phi^n)$ not distinguished by any n-state machine.

Minimizing Picture Words

P. Séébold (Paris)

[joint work with K. Slowinski (Lille)]

With any word over the alphabet $D = \{r, \bar{r}, u, \bar{u}\}$, we associate a connected picture in the following manner: The reading of each letter of this word induces a unit line: r ($\bar{r}$, u, $\bar{u}$ resp.) stands for a right (left, up, down resp.) move.

We present a rewriting system which allows to obtain, from any word over D, all the words describing the same picture.

Particularly, we give an algorithm to find, in finite time, a minimal word describing a given picture: This word represents the shortest way to draw this picture "without pen-up".

Cycles in Picture Words: Elimination[1)] and Generation[2)]

F.-J. Brandenburg (Passau)

[[1)] joint work with J. Dassow (Magdeburg), [2)] joint work with M. Chytil (Prague)]

A picture word is a string over the alphabet π = {u, d, l, r}. These symbols are interpreted as unit lines in the discrete Euclidian plane. A picture word w can be seen as a sequence of commands for a plotter pen, describing a picture p(w). In a graph theoretic view, p(w) is a graph on the grid and w describes an Euler tour of p(w). w describes a cycle, if its picture p(w) is a cycle.

In the first part we define a scheme of 16 reduction rules which reduces picture words by eliminating two out of three traversals of the same unit edge, e.g., u O d O' u $\rightarrow$ O' u O, where O and O' represent cycles. This system is strictly length decreasing, non-confluent and effective in the sense that irreducible picture words are computable in $O(m^2)$ time, where n = |w|. It does not compute minimal picture words. The regular and context-free languages are not preserved by our reductions or by considering minimal picture words.

In part II, we wish to generate cycles or simple cycles for free. A cycle grammar is a CFG with distinguished sets of ordinary and of cycle nonterminals, i.e. (N, N_c, T, P, S). For a cycle nonterminal A_c, a derivation A $\Rightarrow^*$ w is cycle respecting, if w describes a cycle. The cycle language $L_c(G)$ consists of all picture words generated by cycle respecting derivations. In a similar way we define simple cycle grammars producing simple cycles, i.e., with no crossing from their simple cycle nonterminals, when such symbols are ignored in the derivation processes. Cycle and simple cycle grammars are strictly more powerful than standard picture grammars. Particular emphasis is put on the complexity of the recognition problem, i.e. w $\in L_c(G)$, which is polynomial for cycle grammars and for unambiguous simple cycle grammars, and is NP-complete for ambiguous regular simple cycle grammars.

FTR Languages of Star Height One

A. Restivo (Palermo)

In 1982 K. Hashiguchi proved that it is decidable whether a given regular language has star height one. In this talk a different and simpler proof is given for a particular family of languages: Factorial, Transitive and Regular (FTR) languages. A language L is factorial if L = F(L), where F(L) denotes the set of factors of L. L is transitive if, for all U, V $\in$ L, there exists W $\in$ L such that U W V $\in$ L. The following theorem gives a characterization of FTR languages of star height one.

<u>Theorem</u> Let L be a FTR language and let h(L) denote its star height.

h(L) = 1 $\Leftrightarrow$ There exists a <u>finite</u> set H such that L = F (H*).

By this characterization we derive a simple algorithm to decide whether a FTR language has star height one. This is obtained by considering, for a given L, submonoids which are maximal in L. We prove, in particular, that they are regular sets and that they can be effectively constructed.

Some open problems and conjectures are also proposed. In particular: Given a FTR language L, decide whether there exists a finite code C such that L = F (C*).

On Length Distributions of Rational Languages

D. Krob (Paris, Rouen)

(joint work with G. Hansel and Chr. Michaux)

D. Niwinski raised the two following questions:

(Q1) Let L be a rational language, then

$\{n \in N, |L \cap A^n| > 2^{n-1}\} \in \mathrm{Rat}(N)$?

(when A is a two-letter alphabet).

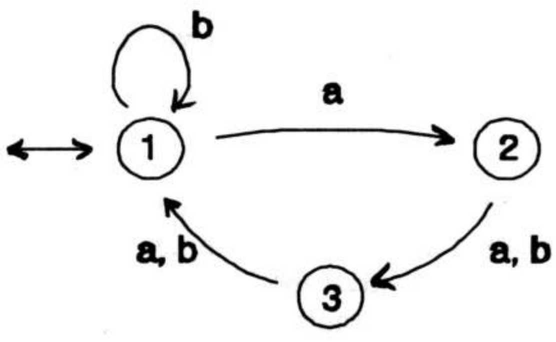
(Q2) Let L be a rational language over $\{a, b\}$, $P \in N[X]$.

$\{w \in A^*, |w^{-1} L \cap A^{P(|w|)}| > 2^{P(|w|)-1}\} \in \mathrm{Rat}(a,b)$?

We show how to transform Niwinski's questions into questions concerning pobabilistic automata. Then using Turakainen's work, we can prove that:

1) "In general", the answers to (Q1) and (Q2) are positive.

2) There are some sparse counter-examples. For instance, the automaton $\mathcal{A}$



(and L = L($\mathcal{A}$,1→1)) is the unique (and smallest) 3-states deterministic complete automaton that gives a counter-example to (Q1).

Some Decidability Results in Formal Series Theory

S. Varricchio (L'Aquila and Paris)

We prove some decidability results about rational power series on a free partially commutative monoid. In particular we show that for these series the equivalence problem is decidable, nevertheless the same problem for rational sets of the free partially commutative monoid is undecidable in the general case.

As a consequence of our result we will prove that the equivalence problem for the unambiguous rational sets of the free partially commutative monoid is decidable for any concurrence relation.

We also show a generalization of the Eilenberg Equality Theorem to rational series with coefficients in a commutative ring.

A Characterization of Regular Languages Recognized by Constant Depth Circuits

K.J. Compton (Ann Arbor)

A characterization of the regular languages in the complexity class AC^0 is given and the implications of this result for determining the structure of the complexity class NC^1 are discussed. (AC^0 is the class of languages accepted by constant depth, polynomial size families of circuits whose gates are of fan-in 2; NC^1 is the class of languages accepted by log depth, polynomial size families of circuits whose gates are of unbounded fan-in). This result appeared in the paper "Regular Languages for NC^1"by D. Barrington, K. Compton, Howard Straubing and D. Thérien, but the proof presented here is different from the proof appearing in the paper. The precise statement of the result is as follows:

For L regular, the following are equivalent.

(i) $L \in AC^0$

(ii) If L has syntactic monoid M(L) and $h: \Sigma^* \to M(L)$ is the syntactic morphism, then $h(\Sigma^n)$ contains only trivial groups for each $n \geq 1$.

(iii) L can be built from the languages $\emptyset$, $\{a\}$ ($a \in \Sigma$), and $(\Sigma^q)^*$ using the operations $\cup$, $\cdot$, and complementation.

(iv) There is a first-order sentence φ such that $L = \{w : \mathcal{A}_{w,q} \models \varphi\}$. Here $\mathcal{A}_{w,q}$ is the structure

$$< \{1, ..., |w|\}, \leq, mod_q, (R_a)_{a \in \Sigma} >$$

where $\{1, ..., |w|\}$ is the universe of the structure, $\leq$ is the usual order relation on this universe, mod_q is the unary relation given by $mod_q(x) \leftrightarrow x \equiv 0$ (mod_q),
and for each $a \in \Sigma$, $R_a(x)$ holds precisely when there is an a in w at position x.

The implication (i) $\Rightarrow$ (ii) is derived using the Furst-Saxe-Sipser-Ajtai Theorem. The implication (ii) $\Rightarrow$ (iii) is derived from a theorem of Schützenberger.

On Counting and Instance Problems Defined Over Formal Languages

M. Goldwurm (Milano)

In this talk I have presented a short survey on the complexity of counting problems on languages together with some new results.

Given a language $L \subseteq \Sigma^*$, Counting (L) is the problem of computing on input 1^n ($n \in N$) the number of strings of length n in L. Moreover, Ranking(L) is the problem of computing on input $y \in \Sigma^*$ the number of words $x \in L$ such that either $|x|<|y|$ or ($|x|=|y|$ and $x \leq_L y$), where $\leq_L$ is the lexicographic ordering over Σ^*.

It is known that, for every $L \in P$, Ranking (L) belongs to #P while Counting (L) belongs to $\#P_1$ which is the restriction of #P to functions having unary input [Valiant '79]. So a general problem is that one of determining the classes of languages in P which have easy counting and ranking problems. In particular it is known that the ranking problem for regular languages is in DIV (the class of problems NC^1 reducible to integer division) and the same result holds for Counting(L) for unambiguous context-free L. Moreover, Counting and Ranking problems for languages recognized by 1-way unambiguous log-space bounded T.M. are in DET (the class of functions NC^1-reducible to computing the determinant). The Ranking problem for unambiguous context-free languages and, more generally, for languages accepted by 1-way unambiguous log space auxiliary P.D.A., is in NC^2.

On the other hand it is also known that Counting and Ranking problems are respectively $\#P_1$-complete and #P-complete for some finitely ambiguous c.f. languages, as well as for languages accepted by one of the following machines: log-space Det. T.M., 1-way Nondet. T.M., 1-way 2 head D.F.A., 2-way Det P.D.A., Nondet. log-time (off line) T.M..

At last a new result is presented which proves that Counting(L) and Ranking(L) belong to DET for every language L accepted by 1-way Nondet. log-space. T.M. with bounded degree of ambiguity k ($k \in N$). Such a result also holds for every language accepted by a 1-way Nondet. log-space Turing Machine M such that for every n the number of accepting computations of M over any input of size n varies in a finite number of values $\{d_1(n), d_2(n), ..., d_k(n)\}$ where k is fixed and does not depend on n.

Recognizability of Complex Trace Languages

V. Diekert (München)

[joint work with P. Gastin (Paris) and A. Petit (Paris)]

The straightforward generalization of the notion of finite Mazurkiewicz trace to infinite trace does not allow any good definition of concatenation. This drawback leads to the consideration of so-called complex traces which were introduced by one of the authors at STACS '91. A complex trace over a dependence alphabet (X, D) is a pair (r, D(A)) where r is a real trace, i.e., a finite or infinite trace in the sense of Mazurkiewicz, and where $D(A) = \{b \in X \mid \exists\, a \in A: (a, b) \in D\}$ for some $A \subseteq X$ with some minor extra conditions.

The concatenation is then defined by the formula

$$(r, D(A))\cdot(s, D(B)) = (r \cdot \mu_{I(A)}(s), D\,(A \cup B \cup \text{alphabet}\,(\mu_{I(A)}\,(s)^{-1}\,s)))$$

where $\mu_{I(A)}(s)$ means the maximal prefix of s containing letters which are independent of A only.

In the present talk we define the notion of recognizability for complex trace languages and we show that the Kleene-Ochmanski Theorem holds. This means that recognizable complex trace languages form a boolean algebra which is closed under concatenation, co-star and co-omega operations and that conversely any recognizable language has a co-rational description. Our results are based on previous work by P. Gastin and P. Gastin/ A. Petit/ W. Zielonka among others.

Conjugacy

Ch. Choffrut (Paris)

Given 2 elements x, y in a group G, they are said to conjugate if there exists a $z \in G$ such that $x = zyz^{-1}$, i.e.

(1) $xz = zy$

Equivalently, there exist two elements $u,v \in G$ such that:

(2) $x = uv$ and $y = vu$.

Distinguish the two conditions (1) and (2) by defining in an arbitrary monoid M the following two relations:

(3) Conj(x,y) iff $xz = zy$ for some $z \in M$
(4) Shift(x,y) iff $x = uv$ and $y = vu$ for some $u,v \in M$

The following inclusion:

$$\text{Shift}^* \subseteq \text{Conj}$$

is straightforward. We address the questions of when each of the following properties hold:

1) $\text{Shift}^* = \text{Conj}$
2) $\text{Shift}^n = \text{Shift}^*$ for some $n > 0$
3) $\text{Conj}\,(x^n, y^n) \Rightarrow \text{Conj}(x,y)$.

A few examples are considered.

i) The bicyclic monoid presented by $< A \cup A;\ \bar{a}a = 1$ for all $a \in A >$

Then $\text{Shift}^2 = \text{Conj}$.

ii) The monoids of traces presented by: $< A;\ ab = ba$ for all $(a,b) \in \Theta>$, where

$$\Theta = \Theta^{-1} \subseteq A\times A$$

Then $\text{Shift}^n = \text{Conj}$ where n is the maximal diameter of a connected companent of the graph of $A\times A - \Theta$ ((a,b) is an edge iff $(a,b) \in A\times A - \Theta$) (cf. C. Duboc, Thèse, Rouen, 1986)

iii) The monoid presented by $< a,b;\ ab = ba^2 >$.

Then $\text{Shift} \not\subseteq \text{Shift}^2 \not\subseteq \dots \not\subseteq \text{Shift}^n \not\subseteq \dots \not\subseteq \text{Shift}^* = \text{Conj}$.

Furthermore, for these three examples, condition 3 holds.

ω-Power and δ-Limit of Languages

L. Staiger (Aachen)

Two operations mapping languages to ω-languages are investigated:

$$L^\omega := \{\xi\colon \xi\in\Sigma^\omega \wedge \xi = w_1 \cdot w_2 \cdot w_3 \dots \text{ for } w_i\in L\}$$

$$L^\delta := \{\eta\colon \eta\in\Sigma^\omega \wedge \exists^\infty\, w \in L : w \text{ is a prefix of } \eta\}$$

Both operations have been proved useful for the characterizations of classes of ω-languages via classes of languages, e.g. (pr denotes projection)

1. Regular ω-languages

$$\text{REG}_\omega = \{\cup_{i=1}^n L_i' \cdot L_i^\omega : L_i, L_i' \in \text{REG}\} = \{\text{pr}\,(L^\delta) : L \in \text{REG}\}$$

2. Context-free ω-languages

$$\text{CF}_\omega = \{\cup_{i=1}^n L_i' \cdot L_i^\omega : L_i, L_i' \in \text{CF}\} \not\subseteq \{\text{pr}\,(L^\delta) : L \in \text{CF}\}$$

3. Recursive ω-languages

$$REC_\omega = \{pr\,(L^\delta) : L \in REC\} \not\supseteq \{\cup_{i=1}^{n} L_i' \cdot L_i^\omega : L_i, L_i' \in REC\}$$

Several open problems concerning an alternative characterization of L^ω, e.g. as δ-limit, in terms of the Borel hierarchy and as a solution of the linear equation $T = L \cdot T$ are considered.

Finally, the solution of the linear equation $T = L \cdot T \cup E$ ($E \subseteq \Sigma^\omega$) is used to derive the identity $(L^*)^\delta = L^\omega \cup L^* \cdot L^\delta$.

Stratified Tree Automata

Wolfgang Thomas (Kiel)

We consider chain logic over the infinite k-ary branching tree, which is obtained from monadic second-order logic SkS by restricting the set quantifiers to chains (i.e., subsets of paths). Many systems of modal logics for programs can be interpreted in chain logic. We introduce an automaton model which characterizes chain logic. These "stratified tree automata" are Muller tree automata equipped with a ranking (or stratification) of the state set, such that (1) in automaton transitions $(q,a,q_1,\dots,q_k)$ we have $\text{rank}(q) \geq \text{rank}(q_k)$, (2) any deterministic state q has even rank, and (3) otherwise rank(q) is odd and equals rank (q_i) for at most one q_i (in a transition as above). The equivalence to chain logic is shown by an easier proof than for Muller tree automata and SkS. Also there a correspondence between rank of initial automaton states and quantifier depth of chain logic formulas.

Participants:

Jean **Berstel**
Univ. Pierre et Marie Curie
LITP
4, Place Jussieu
5252 Paris Cedex 05
Frankreich
berstel@litp.ibp.fr

Ronald **Book**
Dept. of Mathematics
UC Santa Barbara
Santa Barbara, CA 93106
USA
book%henri@hub.ucsb.edu

Franz J. **Brandenburg**
Lehrstuhl für Informatik
Universität Passau
Postfach 2540
W-8390 Passau
brandenb@fmi.uni-passau.de

Véronique **Bruyère**
Université de Mons-Hainaut
Faculté des Sciences
Avenue Maistriau 15
B-7000 Mons
Belgien
SBRUYERE@BMSUEM11.BITNET

Christian **Choffrut**
Université Paris Nord
CSP
Av. J.B. Clément
93430 Villetaneuse
Frankreich
cc@litp.ibp.fr

Kevin **Compton**
EECS Dept.
University of Michigan
Ann Arbor, Michigan 48109-2122
USA
kjc@eecs.umich.edu

Bruno **Courcelle**
Université Bordeaux 1,
Laboratoire Bordelais
de Recherche en Informatique
351, Cours de Libération
33405 Talence Cedex
Frankreich

Volker **Diekert**
TU München
Institut für Informatik
Arcisstr. 21
W-8000 München 2
diekert@lan.informatik.tu-muenchen.dbp.de

Massimiliano **Goldwurm**
Università degli Studi di Milano
Dip. Scienze dell'Informazione
Via Moretto da Brescia 9
20133 Milano
Italien
goldwurm@imucca.unimi.it

Günther **Hotz**
Universität des Saarlandes
Fachbereich 10
Angewandte Mathematik und Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11
hotz@cs.uni-sb.de

Juhani **Karhumäki**
University of Turku
Dept. of Mathematics
20500 Turku 50
Finnland
karhumak@utu.cs.fi

Daniel **Krob**
Laboratoire d'Informatique de Rouen
Université de Rouen
Faculté des Sciences
76134 Mont Saint Aignan Cedex
Frankreich
dk@litp.ibp.fr

M. **Mosbah**
Université Bordeaux 1,
Laboratoire Bordelais
de Recherche en Informatique
351, Cours de Libération
33405 Talence Cedex
Frankreich
mosbah@geocub.greco-prog.fr

Dominique **Perrin**
Université Paris VII
LITP,
2, Place Jussieu
75251 Paris Cedex 05
Frankreich
dp@litp.ibp.fr

Jean-Eric **Pin**
Université Paris VI
LITP
4, Place Jussieu
75252 Paris Cedex 05
Frankreich
pin@litp.ibp.fr

Andreas **Podelski**
LITP
Univertité Paris VII
3, place Jussieu
75251 Paris Cedex 05
Frankreich
podelski@litp.ibp.fr

Lutz **Priese**
Gesamthochschule Paderborn
Universität für Mathematik und Informatik
Fachbereich 17
Warburger Str. 100
W-4790 Paderborn
lutz@pbinfo.de

Antonio **Restivo**
Università di Palermo
Instituto di Matematica
Via Archirafi 34
Palermo
Italien

J. Mike **Robson**
Laboratoire de l'Informatique du Parallélisme
Ecole Normale Supérieure de Lyon
46 Allée d'Italie
Lyon 69364 Cedex 07
Frankreich
robson@ensl.ens-lyon.fr

Patrice **Seebold**
LITP - UFR d'Informatique
Université Paris VII
2, Place Jussieu
75251 Paris Cedex 05
Frankreich
seebold@litp.ibp.fr

Helmut **Seidl**
Fachbereich 14 - Informatik
Universität des Saarlandes
Im Stadtwald
W-6600 Saarbrücken 11
seidl@cs.uni-sb.de

Ludwig **Staiger**
v. Humboldt Str. 1
W-5024 Pulheim

Magnus **Steinby**
University of Turku
Dept. of Mathematics
20500 Turku 50
Finnland
steinby@kontu.utu.fi

Wolfgang **Thomas**
Christian-Albrechts-Universität
Institut für Informatik und
Praktische Mathematik
Olshausenstr. 40
W-2300 Kiel 1
wt@informatik.uni-kiel.dbp.de

Sophie **Tison**
Université de Lille 1
LIFL
Bâtiment M 3
59655 Villeneuve d'Ascq Cedex
Frankreich
tison@lifl.lifl.fr

Stefano **Varricchio**
Universita dell'Aquila
Dip. di Matematica Pura e Applicata
Aquila
Italien

Pascal **Weil**
Université de Paris VI
LITP
Place Jussieu
7525 Paris Cedex 05
Frankreich
weil@litp.ibp.fr

Detlef **Wotschke**
Universität Frankfurt
FB 20 Informatik
Robert-Mayer-Str. 11-15
W-6000 Frankfurt
wotschke@vax1.rz.uni-frankfurt.dbp.de

Bisher erschienene und geplante Titel:

W. Gentzsch, W.J. Paul (editors):
Architecture and Performance, Dagstuhl-Seminar-Report; 1, 18.-20.6.1990; (9025)

K. Harbusch, W. Wahlster (editors):
Tree Adjoining Grammars, 1st. International Worshop on TAGs: Formal Theory and Application, Dagstuhl-Seminar-Report; 2, 15.-17.8.1990 (9033)

Ch. Hankin, R. Wilhelm (editors):
Functional Languages: Optimization for Parallelism, Dagstuhl-Seminar-Report; 3, 3.-7.9.1990 (9036)

H. Alt, E. Welzl (editors):
Algorithmic Geometry, Dagstuhl-Seminar-Report; 4, 8.-12.10.1990 (9041)

J. Berstel , J.E. Pin, W. Thomas (editors):
Automata Theory and Applications in Logic and Complexity, Dagstuhl-Seminar-Report; 5, 14.-18.1.1991 (9103)

B. Becker, Ch. Meinel (editors):
Entwerfen, Prüfen, Testen, Dagstuhl-Seminar-Report; 6, 18.-22.2.1991 (9108)

J. P. Finance, S. Jähnichen, J. Loeckx, M. Wirsing (editors):
Logical Theory for Program Construction, Dagstuhl-Seminar-Report; 7, 25.2.-1.3.1991 (9109)

E. W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 8, 4.-8.3.1991 (9110)

M. Broy, P. Deussen, E.-R. Olderog, W.P. de Roever (editors):
Concurrent Systems: Semantics, Specification, and Synthesis, Dagstuhl-Seminar-Report; 9, 11.-15.3.1991 (9111)

K. Apt, K. Indermark, M. Rodriguez-Artalejo (editors):
Integration of Functional and Logic Programming, Dagstuhl-Seminar-Report; 10, 18.-22.3.1991 (9112)

E. Novak, J. Traub, H. Wozniakowski (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 11, 15-19.4.1991 (9116)

B. Nebel, C. Peltason, K. v. Luck (editors):
Terminological Logics, Dagstuhl-Seminar-Report; 12, 6.5.-18.5.1991 (9119)

R. Giegerich, S. Graham (editors):
Code Generation - Concepts, Tools, Techniques, Dagstuhl-Seminar-Report; 13, , 20.-24.5.1991 (9121)

M. Karpinski, M. Luby, U. Vazirani (editors):
Randomized Algorithms, Dagstuhl-Seminar-Report; 14, 10.-14.6.1991 (9124)

J. Ch. Freytag, D. Maier, G. Vossen (editors):
Query Processing in Object-Oriented, Complex Object, and Nested Relation Databases, Dagstuhl-Seminar-Report; 15, 17.-21.6.1991 (9125)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Model Theory, Dagstuhl-Seminar-Report; 16, 24.-28.6.1991 (9126)