MATHEMATISCHES FORSCHUNGSINSTITUT OBERWOLFACH

T a g u n g s b e r i c h t 47/1983

Complexity Theory

30.10. - 4.11.1983

The 6th Oberwolfach Conference on Complexity Theory was organized as before by C.P.Schnorr (Frankfurt), A.Schönhage (Tübingen), and V.Strassen (Zürich). The 41 participants came from 11 countries, 16 participants came from North and South America and Israel.

42 lectures were given at the conference covering distinct areas of complexity theory. They dealt with subjects of algebraic, numerical, number theoretical, geometric, and combinatorial nature and their applications including problems of concrete computer implementation.
Lectures were given on bilinear complexity, polynomial complexity, solving algebraic equations numerically by fast algorithms, problems connected with number theory as cryptography and primality testing, computer implementation of fast algorithms, Boolean complexity, VLSI and Computer Design, complexity classes, and in particular on the complexity classification of problems from graph theory, sorting, and coding.

Participants

Atkinson, M., Ottawa
Baur, W., Konstanz
Beth, Th., Erlangen
Bini, D., Pisa
Borodin, A., Toronto
Clausen, M., Zürich
Collins, G.E., Madison
Cook, S.A., Toronto
van Emde Boas, P., Amsterdam
Fürer, M., Zürich
Galil, Z., New York
von zur Gathen, J., Toronto
de Groote, F., Frankfurt
Heintz, J., Frankfurt
Hopcroft, J., Ithaca
Klawe, Maria, San José
Lenstra, H.W., Amsterdam
Leeuwen, J. van, Utrecht
Lickteig, Th., Tübingen
Loos, R., Karlsruhe
Mehlhorn, K., Saarbrücken
Meyer auf der Heide, F., Frankfurt
Monien, B., Paderborn
Morgenstern, J., Nizza
Ong, Heidrun, Frankfurt
Pan, V., Albany
Paterson, M.S., Coventry
Paul, W.J., San José
Pippenger, N., San José
Reischuk, R., Bielefeld
Schnorr, C.P., Frankfurt
Schönhage, A., Tübingen
Shamir, A., Rehovot
Sieveking, M., Frankfurt
Stoss, H.J., Konstanz
Strassen, V., Zürich
Valiant, L.G., Cambridge, Ma.
Volger, H., Tübingen
Wegener, I., Frankfurt
Yao, A.C., Stanford
Yao, F.F., Palo Alto

Vortragsauszüge

M.D. ATKINSON : The complexity of euclidean congruence

Let S, T be two n-point sets in real 3-dimensional space and let Cong(S,T) be the set of Euclidean congruences from S onto T . An algorithm of time complexity $n \log n$ is given. The algorithm applies a number of symmetry tests to S and T which reveal either that S is not congruent to T or produce two sets S^*, T^* of bounded size such that $\mathrm{Cong}(S,T) \subseteq \mathrm{Cong}(S^*,T^*)$.
The algorithm makes use of an algorithm of Shamos and Bentley for finding shortest distances, a fast pattern recognition algorithm, and the Euler formula relating the edges and faces in a planar graph. It is optimal to within a constant factor.

T. BETH : On the complexity of group algebras

The investigation of fast regular algorithms (e.g. FFT, WFTA etc.) leads to the study of certain G-modules associated with certain suitable groups (e.g. groups of the roots of unity, Galois groups etc.). Generalization to a much wider class of problems leads to the study of complexity of the "universal" G-module $\mathbb{F}G$, the group algebra of G over $\mathbb{F}$. Very rough estimates show that the usual "detour" of computing in semisimple group algebras $\mathbb{F}G$ by computing in the isomorphic Wedderburn-algebra, can only guarantee an expected reduction of complexity, if a fast transformation method for carrying out this isomorphism can be derived.
It is shown that for solvable groups G whose order n is only divided by primes of bounded size, the complexity of the transformation algorithm can be reduced to $O(n^{\omega/2})$, where $O(n^{\omega})$ denotes the complexity of multiplying matrices of order n .
The proof is based on the iterative use of Clifford's theorem. The method seems to be applicable to wider classes of groups that shall be studied subsequently.

D. BINI : Some computational problems concerning linear spaces of matrices

Linear spaces of matrices occur in several problems of Numerial Analysis. Main computational issues are the construction of concrete fast algorithms for matrix inversion, computation of the determinant, approximation of the eigenvalues and eigenvectors of a matrix belonging to a linear space. We show that the concepts of tensor rank and border rank provide a valuable tool for this investigation. We prove that if the linear space $\hat{A}$ of $n \times n$-matrices is such that tensor rank $(\hat{A}) = n + k$ (border rank $(\hat{A}) = n + k$), where $k < n$ is a constant, then for any $A \in \hat{A}$ the computation (approximation) of $\det A$ costs $n(k^2+1) +$ constant multiplications or $\lceil \log_2 n \rceil +$ constant parallel steps. Similar results hold for achieving one step of the bisection method or Newton method applied to the characteristic polynomial of A, yielding concrete fast algorithms for approximating the eigenvalues of A. The inverse of any non singular matrix $A \in \hat{A}$ can be computed in $\lceil \log_2 n \rceil +$ constant parallel steps. Moreover, in the case of approximate algorithms, switching from approximate to exact algorithm increases the number of processors, leaving almost unchanged the number of parallel steps.
Applications to some important classes of matrices are given.

G.E. COLLINS : A close look at Karatsuba integer multiplication

The critical value of the Karatsuba algorithm is the operand size at which the algorithm becomes faster than the classical algorithm. An existing implementation in the SAC-2 computer algebra system has a critical value of 38 beta-digits (about 38o decimal digits) when running on the University of Wisconsin Madison's Univac 1100 computer. In this implementation integers are kept in linked lists; a proposed implementation which would perform multiplications in a large scratchpad array would have a critical value of about 21 beta-digits. At an operand size of about 250 beta-digits this Karatsuba algorithm implementation would become twice as fast as

the classical algorithm, at about 1300 beta-digits four times as fast, and at about 8000 beta-digits ten times as fast. On a computer of the future executing instructions at a rate of 100 per microsecond, this last multiplication would take 2 seconds using the Karatsuba algorithm.

St.A. COOK : The parallel complexity of the Abelian permutation group membership problem
(joint work with Pierre McKenzie)

The Abelian permutation group membership problem (APGM) is in RNC^3 (i.e. it can be solved by Boolean circuits of depth $O(\log^3 n)$ and polynomial size, provided the circuit is allowed coin toss inputs). In fact, this problem is reducible to the problem of solving a (singular) system of linear diophantine equations modulo a number m, all of whose prime power divisors are small. We show that the latter problem is in RNC^3. Also APGM is in NC^1 (solvable by deterministic $O(\log n)$ depth circuits) in case the number of generators of the group is bounded. Finally, APGM is hard for nondeterministic log space.

P. VAN EMDE BOAS : On tape versus Core - an application of space efficient hashing functions to the invariance of space
(joint work with C. Slot)

Complexity classes like P and NP are well defined based on the fact that within the family of "reasonable" machine models, each model can simulate each other model with a polynomially bounded overhead in time. Similarly, in order that a class like LOGSPACE is well defined, one needs to establish that these models simulate each other with a constant factor overhead in space. It seems that the standard definition for the space measure on RAM's, with respect to this issue, is not the correct one. We provide an alternative definition which is correct, and show that for the case on on-line computations the two definitions indeed are

different. Our case would be much stronger if we could provide an off-line counterexample as well, but an attempted counterexample fails to separate tape and core. The simulation which succeeds in accepting this language on a Turing machine in extremely little space is based upon an improvement with respect to space consumption of the perfect hashing functions described by Fredman, Komlos & Szémerédi in their 1982 FOCS paper. We show that it is possible to obtain perfect hash functions for n-element subsets of an u-element universe requiring space $O(\log(u) + n)$ bits for being designed, described and evaluated. A generalisation of our simulation shows that the counterexample looked for does not exist - the two measures lead for deterministic RAM's to the same space-complexity classes.

M. FÜRER : A fast algorithm for the roots of complex polynomials

An algorithm is presented whose input consists of a positive integer s and the coefficients of a complex polynomial of degree n with sufficiently high precision (say given in floating-point representation with an exponent of bounded length). The algorithm computes the roots of the polynomial with precision 2^{-s} in time $O(p(n) \cdot s^{1+\varepsilon})$ for some polynomial p and every $\varepsilon > 0$. The degree of p is higher than in Schönhage's algorithm, but the method is much simpler. Therefore this algorithm might be faster for practical values of s and n. Furthermore it seems that the actual running time is faster (in the average and in the worst case) than the proven running time. The algorithm compares favorably with all the other algorithms I know, because these have one of the following disadvantages:

- they don't work for all polynomials or for unlucky choices of starting values
- the running time is quadratic in s (only linear convergence in the presence of clusters of zeros)
- the running time is exponential in n.

The algorithm is motivated by a geometric mechanical model of Newton's method. But when the Newton correction is bad, because

some "gravitational" forces are small, the algorithm switches discontinuously to higher forces. And accelerated Newton corrections are used in order to handle multiple zeros and clusters of zeros correctly and fast.

J. v.z. GATHEN : Factoring sparse multivariate polynomials

This talk presents probabilistic algorithms for testing irrecibility and computing the factorization of sparse multivariate polynomials. The running time for the first algorithm is polynomial in the input size, and for the factoring algorithm polynomial in input plus output size. Both algorithms work over algebraic number fields and over finite fields. They are based on an effective version of Hilbert's irreducibility theorem.

H.F. DE GROOTE : Complexity of Lie algebras
(joint work with Joos Heintz)

In the following $\mathfrak{g}$ denotes a (real or complex) Lie algebra (finite dimensional), $\mathfrak{h} \subset \mathfrak{g}$ a Cartan subalgebra, $L(\mathfrak{g})$ the multiplicative complexity of the bracket operation $(X,Y) \longmapsto [X,Y]$ and $R(\mathfrak{g})$ the bilinear multiplicative complexity of the bracket. $\mathfrak{g}$ is always supposed to be semisimple. The following results were presented :

Prop. 1 Let $[X,Y] = \sum_{\rho=1}^{R} u_\rho(X)\, v_\rho(Y)\, w_\rho$ be a bilinear computation of $[X,Y]$. W.l.o.g. let $\{w_1,\ldots,w_n\}$ be a basis of $\mathfrak{g}$. Then $H := \{u_{n+1},\ldots,u_R\}^{\perp}$ is an abelian subalgebra of $\mathfrak{g}$.

Corollary Let $\mathfrak{g}$ be compact. Then $R(\mathfrak{g}) \geq 2 \dim_{\mathbb{R}} \mathfrak{g} - \dim_{\mathbb{R}} \mathfrak{h}$.

Prop. 2 Let $\mathfrak{g}$ be compact and $\mathfrak{g}^{\mathbb{C}} := \mathbb{C} \otimes_{\mathbb{R}} \mathfrak{g}$ its complexification. If $R(\mathfrak{g}) = 2 \dim_{\mathbb{R}} \mathfrak{g} - \dim_{\mathbb{R}} \mathfrak{h}$ then

$$\mathfrak{g}^{\mathbb{C}} \cong \prod \mathfrak{sl}(2,\mathbb{C}) \ .$$

Remark $\dim_{\mathbb{C}} \mathfrak{a} \leq 2$ for all abelian subalgebras of $\mathfrak{sl}(3,\mathbb{C})$.
Hence $R(\mathfrak{sl}(3,\mathbb{C})) \geq 15$.

Prop. 3 $R(sl(n,\mathbb{C})) \geq L(sl(n,\mathbb{C})) \geq 2 \dim_{\mathbb{C}} sl(n,\mathbb{C}) - \dim_{\mathbb{C}} h \quad (= 2n^2 - 2n)$.

Prop. 4 Let g be a complex semisimple Lie algebra, $\Gamma(g)$ its isotropy group and $\Gamma^0(g) \subseteq \Gamma(g)$ the small isotropy group. Then

$$\Gamma^0(g) \cong \mathbb{C}^\times \times \mathbb{C}^\times \times \mathrm{Aut}(g) \ ,$$
$$\Gamma(g)/\Gamma^0(g) \cong \gamma_3 \ ,$$

where $\mathrm{Aut}(g)$ is the automorphism group of g and γ_3 the permutation group of three elements.

J. HEINTZ : Polynomials with symmetric Galois group which are easy to compute

Let k be a Hilbertian field (e.g. a field of finite type over its prime field with transcendence degree ≥ 1 in case characteristic > 0) and let X be an indeterminate over k.
We construct a sequence $(F_d)_{d \in \mathbb{N}}$ of polynomials $F_d \in k[X]$ of degree $\deg F_d = d$ and nonscalar complexity $L(F_d) \leq 7 + 2 \log_2 d$ with symmetric Galois group. This has two consequences :

1. Let $v, d \in \mathbb{N}$ such that $7 + 2 \log_2 d \leq v$. The set of all polynomials $F \in k[X]$ with $\deg F \leq d$ and $L(F) \leq v$ contains a Zariski-dense subset of polynomials with symmetric Galois group. Since the polynomials $F \in k[X]$ with $L(F) \approx \log_2 \deg F$ can be considered as those polynomials which are easy to compute, we can rephrase this result as follows: Almost all polynomials, in particular those which are easy to compute, have maximal, i.e. symmetric, Galois group.
2. Let $F \in k[X]$ with $d = \deg F$ and symmetric Galois group. Then all factors G of F (with coefficients in the splitting field of F) with $\deg G$ not too close to d are hard to compute (i.e. $L(G) \approx \sqrt{\deg G}$). So 1. implies that almost all polynomials have factors which are hard to compute. In particular, there exist polynomials which are easy to compute with all factors of not too high degree hard to compute.

J. HOPCROFT : Complexity problems in robotics

In this talk we introduce some of the algorithm problems in robotics. In particular, we formulate a general frame work for the multiple object motion planning and provide a heuristic solution. The planning of motion for rectangles in a rectangular box is shown PSPACE-hard. Motion of non-rigid objects such as linkages is considered, and the complexity of various problems is classified. In particular, it is shown motion planning without boundaries is as hard as with boundaries and is PSPACE-hard.

M. KLAWE : A tight bound for black and white pebbles on the pyramid

Lengauer and Tarjan proved that the number of black and white pebbles needed to pebble the root of a tree is at least 1/2 the number of black pebbles needed to pebble the root. We extend this result to a class of acyclic directed graphs which includes pyramid graphs.

J. VAN LEEUWEN : Data organization for parallel computing (joint work with H.A.G. Wijshoff)

Modern vector- and arrayprocessing computers have one or more highly pipelined processing units and a (large) number of memory banks that can be accessed independently in parallel. A skewing scheme is any storage scheme S that assigns the elements of a matrix to an address in the M memory banks available such that any "template" of data items of interest (rows, blocks, etc.) can be retrieved conflictfree. Skewing schemes were introduced in the nineteen sixties, in the systems programming efforts for the ILLIAC IV, and continue to be of interest to designers of large computers. The common linear skewing schemes are an instance of the larger class of periodic schemes defined by Shapiro in terms of tables of "bounded size". We show that periodic skewing schemes are best analysed using their connection to (classical) integer lattices.

A number of new results are presented for estimating the minimum number of memory banks required for common types of conflictfree access, and necessary and sufficient conditions are derived in terms of the basis of the generating lattice for a periodic skewing scheme to be essentially linear.

H.W. LENSTRA, JR. : Primality testing and Galois theory

It will be shown how Galois theory, both for algebraic numbers and for finite rings, can be used to present several primality testing algorithms from a unified point of view. A central role is played by the notion of an Artin symbol for extensions with an abelian Galois group. This theory makes it possible to combine the recent Jacobi sum tests of Adleman et al. with the older tests of Lucas, Lehmer, Brillhart and Selfridge, as generalized by Williams. It is expected that this combination will lead to an important improvement in practice.

R. LOOS : An analysis of the improved Kronecker algorithm for factoring polynomials over $Q(\alpha)$

Let A be the minimal polynomial of α over Q of degree m. Let f be a squarefree polynomial over $Q(\alpha)$ of degree n. Following Trager $f(\alpha,x)$ can be translated to $f(\alpha,x+s\alpha)$ for the integer $0 \leq s \leq m^2n^2$ such that the norm of f over $Q(\alpha)$ is squarefree too. Then Kronecker's algorithm ("Grundzüge einer arithmetischen Theorie algebraischer Grössen" 1882) has the computing time $O(m^2n^2H^2) + t_{\mathbb{Z}}(k,d)$, where $H = m\cdot n + n\log|A|_1 + m\cdot\log|f|_1$ and $t_{\mathbb{Z}}(k,d)$ is the time for factorization of an integral polynomial of degree k and coefficient length d. With Schönhage's new algorithm for this purpose (see this conference) the time for the algorithm over $Q(\alpha)$ becomes $O(k^{14} + k^{12}(\log d)^2)$ with $k = \max(m,n)$ and $d = \max(|A|_1, |f|_1)$. Empirically, however, the gcd over $Q(\alpha)$ dominates the total computing time.

K. MEHLHORN : Area-time optimal VLSI integer multiplier with minimum computation time

According to VLSI theory $[\log n, \sqrt{n}]$ is the range of computation times for which there may exist AT^2-optimal multiplier for n-bit integers. Such networks were previously known for the time range $[\Omega((\log n)^2), O(\sqrt{n})]$; we settle this question by exhibiting a class of AT^2-optimal multipliers with computation times $[\Omega(\log n), O(n^{2/5})]$.

F. MEYER AUF DER HEIDE : A polynomial algorithm for the n-dimensional Knapsack problem

A linear search algorithm is presented which recognizes the n-dimensional knapsack problem in $2n^4\log(n) + O(n^3)$ steps. This algorithm works for inputs consisting of n numbers for some arbitrary but fixed integer n. This result solves an open problem posed by Dobkin/Lipton and A.C.C. Yao among others. It destroys the hope of proving nonpolynomial lower bounds for this NP-complete problem in the model of linear search algorithms. A generalization to the integer programming problem is presented, and some lower bounds for this problem are established.

B. MONIEN : How to find long paths efficiently

We study the complexity of finding long paths in directed or undirected graphs. Given a graph $G = (V,E)$ and a number k our algorithm decides within time $O(k! \cdot |V| \cdot |E|)$ for all $u,v \in V$ whether there exists some path of length k from u to v. The complexity of this algorithm has to be compared with $O(|V|^{k-1} \cdot |E|)$ which is the worst case behaviour of the algorithms described up to now in the literature. We get similar results for the problems of finding a longest path, a cycle of length k or a longest cycle, respectively.
Our approach is based on the idea of representing certain families of sets by subfamilies of small cardinality. We also discuss the border lines of this idea.

J. MORGENSTERN : Implementation of two dimensional Fourier transform via polynomial transform (joint work with M. Lhomme and M. Quandalle)

Polynomial transforms were discovered by Nussbaumer and Quandalle in 1979. It is a fast way to compute convolutions of sequences of polynomials $\mathrm{mod}(Z^N-1)$ by avoiding certain multiplications since the root of the transformation (analogous to the ordinary Discrete Fourier Transform) is a power of Z which results in shifts in words.
M. Truong and Reed have implemented such algorithms enabling to compute two dimensional convolutions and saving 20% of the time. In a joint work with M. Lhomme and M. Quandalle we implemented a two dimensional Fourier transform using that tool which requires less operations but is not faster on different machines. In particular this FORTRAN Program cannot beat a parallel program even on a scalar machine.

H. ONG : Signatures through approximate representations by quadratic forms (joint work with C.P. Schnorr)

We propose a signature scheme where the private key is a random (n,n)-matrix T with coefficients in $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$, m a product of two large primes. The corresponding public key is A,m with $A = T^tT$. A signature y of a message $z \in \mathbb{Z}_m$ is any $y \in (\mathbb{Z}_m)^n$ such that y^tAy approximates z, e.g. $|z-y^tAy| < 4m^{2^{-n}}$. Messages z can be efficiently signed using the private key T and by approximating z as a sum of squares. Heuristical arguments show that forging signatures is not easier than factoring m. The prime decomposition of m is not needed for signing messages, however knowledge of this prime decomposition enables forging signatures. Distinct participants of the system may share the same modul m provided that its prime decomposition is unknown. Our signature scheme is faster than the RSA-scheme.

M.S. PATERSON : An improved depth $O(\log n)$ comparator network for sorting

The recent $O(\log n)$ network devized by Ajtai, Komlós, and Szemerédi is simplified and improved. For the new network we provide closer estimates of the parameter values need. The principal simplification has been the elimination of their Zig and Zag steps with separate register reassignment stages in favour of a purely rhythmical 'pumping action' combining the functions of those steps.

N. PIPPENGER : The explicit construction of highly expanding graphs

We extend the methods of G.A. Margulis, D. Gabber, and Z. Galil to the construction of highly expanding graphs. We construct bipartite graphs that expand sets of size αn out of n to sets of size βn out of n with degree

$$O\left(\left(\frac{1}{\alpha(1-\beta)}\right)^c \left(\log \frac{1}{\alpha(1-\beta)}\right)^{2c}\right) ,$$

where $c = \log_3(1 + \sqrt{2} + 2\sqrt{3}) = 1.62\ldots$. We describe applications to the construction of superconcentrators with limited depth and sorting schemes with few rounds.

R. REISCHUK : Coding strings by pairs of strings (joint work with Chung, Paul and Tarjan)

Let $X, Y \subset \{0,1\}^*$. We say Y codes X if every $x \in X$ can be obtained by applying a short program to some $y \in Y$. We are interested in sets Y that code X robustly in the sense that even if we delete an arbitrary subset $Y' \subset Y$ of size k , say, the remaining set of strings $Y \setminus Y'$ still codes X . In general, this can only be achieved by making in some sense more than k copies of each $x \in X$ and distributing these copies on different strings Y . Thus if the strings in X and Y have the same length, then $\#Y \geq (k+1)\#X$.

If we allow coding of X by Y in a way that every $x \in X$ is obtained from strings $x, z \in Y$ by application of a short program, then we can do better.
Let $Y = \{\oplus x \mid S \subset X\}$ where $\oplus$ denotes bitwise sum mod 2. Then $\#Y = 2^{\#X}$, yet Y codes X robustly for $k = 2^{\#X-1}-1$. We explore the limitations of coding schemes of this nature.

C.P. SCHNORR : The complexity of quadratic equations modulo n (joint work with H. Ong and A. Shamir)

We study the complexity of solving for given k, m, $n \in \mathbb{N}$, n a product of two large primes, the equation

(1) $\quad x^2 + ky^2 = m \pmod n$.

This equation, with $-k$ a square modulo n, is the base of the OSS-signature scheme presented by A. Shamir. Solving (1) is not easier than solving a general, binary quadratic equation modulo n. Equation (1) has uniform complexity which indicates that the corresponding signature scheme is cryptographically strong.

Theorem Any $T(n)$-time algorithm which for fixed n solves equation (1) for an ε-fraction of the messages m and multipliers k, $-k$ a square modulo n, yields a probabilistic $\frac{1}{\varepsilon}\, T(n)$-time algorithm for solving equation (1) with probability $\geq 1/2$ for arbitrary m and k. Equation (1) is a special case of the equation

(2) $\quad ax^2 + bxy + cy^2 = [x,y] \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = m \pmod n$.

Let $G(\Delta)$ be the class group of $SL_2(\mathbb{Z})$-equivalence classes of primitive, positive quadratic forms $ax^2 + bxy + cy^2$ with negative discriminant $\Delta = b^2 - 4ac$. Equation (2) can be solved by exploiting the group structure of $G(\Delta)$.

Lemma Let the order $|G(\Delta)|$ be odd with prime decomposition $\prod_{i=1}^{t} p_i^{e_i}$ then equation (2) can be solved in $O(\max_i p_i^{e_i})$ multiplications in $G(\Delta)$.

Using some heuristic assumptions on the behaviour of class groups this yields an $o(e^{\sqrt{\ln k \ln\ln k}})$-time algorithm for solving equation (1).

Zum Bericht von C.P. SCHNORR :

A Prize of $ 100 will be paid to the first person who

1. finds a provably polynomial time algorithm for solving the equation $m \equiv s_1^2 + k s_2^2 \pmod n$ for arbitrary k,n,m , or
2. solves a challenge equation with a 1000 bit modulus n that will be sent to him upon request.

A. SCHÖNHAGE : Factorization of univariate polynomials

Factorization over $\mathbb{C}$ means, for any given polynomial $P(z) = a_n z^n + \ldots + a_0$ with $a_j \in \mathbb{C}$ and $|P| = \sum_\nu |a_\nu| \leq 1$, and for given $s \in \mathbb{N}$, to compute approximate linear factors $L_j(z) = u_j z + v_j$ $(1 \leq j \leq n)$ such that $|P - L_1 L_2 \ldots L_n| < 2^{-s}$ holds.

Theorem 1. There exists a multitape TM doing this in time
$$O(n^3 \log n + n^2 s) \lg(ns) \lg\lg(ns) .$$
The underlying splitting circle method combines numerical techniques like Newton's method, Fourier transforms, and Graeffe's method with fast algorithms from complexity theory.

Factorization over $\mathbb{Z}$ can be accomplished by computing a root z of the given integer polynomial $F(x)$ and then finding its minimal polynomial by diophantine approximation of $1, z, z^2, z^3, \ldots$ via the basis reduction algorithm of Lenstra, Lenstra and Lovasz. The time analysis yields

Theorem 2. Factorization of $F(x)$ is possible in time
$$O(n^{7+\varepsilon} + n^{5+\varepsilon} \cdot (\log|F|)^{2+\varepsilon}) .$$

A. SHAMIR : An efficient signature scheme based on quadratic equations
(joint work with H. Ong and C.P. Schnorr)

In this talk I present a new signature scheme which is based on the binary quadratic equation $m \equiv s_1^2 + k s_2^2 \pmod n$, where m is the message, (s_1, s_2) is the signature, and (n,k) is the public key. For messages $m \in \mathbb{Z}_n^*$ the set of signatures is characterized by $s_1 \equiv (m/r + r)/2 \pmod n$,

$s_2 \equiv (m/r - r)\cdot u/2 \pmod n$, where r ranges over $\mathbb{Z}_n^*$ and $u \in \mathbb{Z}_n^*$ represents the secret key. The complexity of signature generation is one modular multiplication and one modular division, and the complexity of signature verification is three modular multiplications. The security of the scheme depends (but is not provably equivalent to the difficulty of factoring, and thus it is recommended to use a composite modulus with at least 1000 bits. The signature scheme has uniform complexity with respect to k and m, and the secret signature key cannot be obtained by analyzing message-signature triplets if factoring is difficult.

M. SIEVEKING : The set of functions defined by a sequence of nonscalar operations

Given a sequence β of multiplications/divisions $o_1,\ldots,o_k$ and indeterminates $x_1,\ldots,x_n$, define a sequence of functions in $\mathbb{C}[x_1\ldots x_n]$ by

$$r_{-i} = x_i \qquad (1 \le i \le n)$$

$$r_0 = 1$$

$$r_j = s_j\, o_j\, t_j\,, \quad s_j = \sum_{s=-n}^{j-1} a_{js}\, r_s \qquad t_j = \sum_{s=-n}^{j-1} b_{js} r_s \qquad (1 \le j \le k)$$

$$s^\beta = \sum_{s=-n}^{k} a_{k+1s}\, r_s$$

We study the set V_β of all s^β with varying parameters a_{ij}, b_{lm}. Bounds for the dimension of V_β are applied to the question 1) How long is the shortest possible correct test sequence for V_β ?

2) How complex are multiples of functions?

3) How complex is the "f(x) = 0" test problem compared with the f(x) evaluation problem?

L.G. VALIANT : Short monotone formulae for the majority function

It is shown that for the n-input Boolean majority functions there exist monotone formulae of size $O(n^{5.3})$.

H. VOLGER : Some remarks on the evaluation of powers (joint work with T. Lickteig)

We consider the nonscalar complexity $L(x^n)$ of the rational function x^n (i.e. minimum number of multiplications and divisions) and the discrete complexity $l(x^n)$ which is the length of the shortest addition/subtraction chain for n. The following is known: $\lceil \log n \rceil \leq L(x^n) \leq l(x^n) \leq \lfloor \log n \rfloor + \bar{s}(n) - 1$.

Question 1: $\exists n : L(x^n) < l(x^n)$?

Question 2: $\limsup(L(x^n) - \lceil \log n \rceil) = \infty$?

We have the following partial results:

(1) $L(x^{2^k-1} + \ldots) \geq k+1$ for $k \geq 3$

$L(x^{2^k-1}) = l(x^{2^k-1}) = k+1$ for $k \geq 3$

(2) $L(x^{-2^k+1} + \ldots) \geq k+1$ for $k \geq 2$

$L(x^{-2^k+1}) = l(x^{-2^k+1}) = k+1$

Let L_* resp. l_+ be the divisionfree analogues of L resp. l.

(3) $L_*(x^{2^k-1} + \ldots) \geq k+2$ for $k \geq 5$

in particular $L_*(x^{31}) = l_+(x^{31}) = 7 > L(x^{31}) = l(x^{31}) = 6$

(4) $x^7 = \lim_{\varepsilon \to 0} p_\varepsilon$, $L(p_\varepsilon) = 3$, $L(x^7) = 4$

$x^{-3} = \lim_{\varepsilon \to 0} q_\varepsilon$, $L(q_\varepsilon) = 2$, $L(x^{-3}) = 3$

I. WEGENER : Decision trees and restricted branching programs for the computation of Boolean functions

Decision trees for Boolean functions are labelled binary trees where the leaves are labelled by 0 or 1 and the inner nodes by Boolean variables. The computation starts at the root. At each node we test the appropriate Boolean variable and follow the left (0) or right (1) edge until we reach a leaf where we read the value of the function. In branching programs we drop the assumption that the in degree of the nodes is 1. We consider those restricted branching programs, where we are allowed to test each variable on each path only once. We give optimal algorithms for the computation of optimal decision trees and restricted branching programs for symmetric functions. For the k-clique function we prove large (for non constant k even exponential) lower bounds.

A.C. YAO : Lower bounds on restricted boolean computations

As strong lower bounds to problems in NP have so far been elusive in the general computational models, it is of interest to study the complexity of such problems for restricted classes of Boolean circuits, in the hope that new techniques might be developed. In this talk we show that to compute the majority function of n variables, any monotone circuit of depth 3 must have size at least $2^{n^{\varepsilon}}$, and any branching program of width 2 must have superpolynomial size. We also show that, any depth-4 circuit with "Exclusive-OR" gates at the lowest level, must have superpolynomial size in order to compute the majority function.

Berichterstatter: J. Heintz

Adressen der
Tagungsteilnehmer

Dr. E. George Collins
Computer Science Department
University of Wisconsin
1210 W. Dayton St.

Madison, Wis. 53706
USA

Prof. Michael Atkinson
School of Computer Science
Carleton University

Ottawa
Canada K1S 5B6

Dr. St. Cook
Department of Computer Science
University of Toronto

Toronto, Canada M5S 1A7

Prof. Dr. Walter Baur
Fachbereich Mathematik
Universität Konstanz

7750 Konstanz

Dr. Peter Van Emde Boas
ITW/VPW
Universität Amsterdam
Roeterstraat 15

NL - 1018 WB Amsterdam

Dr. Thomas Beth
Informatik 1
Martensstr. 3

852 Erlangen

Dr. Martin Fürer
Institut für Angewandte Mathematik
Universität Zürich

CH - 8001 Zürich

Dr. Dario Bini
Department of Mathematics
University of Pisa

I - 56100 Pisa

Prof. Dr. Zvi Galil
Columbia University in the City
of New York
Department of Computer Science

New York, N.Y. 10027
USA

Dr. Allan Borodin
Department of Computer Science
University of Toronto

Toronto M5S 1A7
Canada

Dr. Joachim von zur Gathen
Department of Computer Science
University of Toronto

Toronto MS5 1A7

Dr. Michael Clausen
Institut für Angewandte Mathemat.
Universität Zürich
Rämistr. 74

CH - Zürich

Prof. Dr. H.F. de Groote
FB Mathematik
Universität Frankfurt
Robert-Mayer-Str. 6-10

6000 Frankfurt/Main

Dr. Joos Heintz
FB Mathematik
Universität Frankfurt
Robert-Mayer-Str. 6-10

6000 Frankfurt/Main

Prof. John E. Hopcroft
Department of Computer Science
Cornell University

Ithaca, N.Y. 14850
USA

Prof. Maria Klawe
IBM Research

San Jose, CA 95193
USA

Prof. Dr. H.W. Lenstra, Jr.
Mathematisch Instituut
Universiteit van Amsterdam
Roetersstraat 15

NL 1018 WB Amsterdam

Dr. Jan van Leeuwen
Dept. of Computer Science
Universität von Utrecht
P.O.Box 80.012

NL - 3508 TA Utrecht

Dr. Thomas Lickteig
Mathematisches Institut
Universität Tübingen
Auf der Morgenstelle 10

7400 Tübingen

Prof. Dr. R. Loos
Fachbereich Informatik
Universität Karlsruhe

7500 Karlsruhe

Prof. Dr. K. Mehlhorn
FB 10
Universität

6600 Saarbrücken

Dr. F. Meyer auf der Heide
Fachbereich Informatik
der Universität Frankfurt
Dantestr. 5

6000 Frankfurt/Main

Prof. Dr. B. Monien
FB Mathematik/Informatik
Universität

4790 Paderborn

Prof. Jacques Morgenstern
Département de Mathématiques
Université de Nice

Nizza
Frankreich

Frau
Heidrun Ong
Mathematisches Seminar
Robert-Mayer-Str. 6-10
Universität Frankfurt

6000 Frankfurt/Main

Dr. V. Pan
Computer Science Department
SUNY

Albany, N.Y. 12222
USA

Prof. Mike Paterson
Department of Computer Science
University of Warwick

Coventry CV4 7 AL
Great Britain

Dr. Wolfgang Paul
IBM Research Laboratory
5600 Cottle Road

San José, California 95193
USA

Prof. N. Pippenger
IBM Research Laboratory
5600 Cottle Road

San José, California 95193
USA

Prof. Dr. R. Reischuk
Fakultät für Mathematik
Universitätsstraße

4800 Bielefeld

Prof. Dr. C.P. Schnorr
FB Mathematik
Universität Frankfurt
Robert-Mayer-Str. 6-10

6000 Frankfurt/Main

Prof. Dr. A. Schönhage
Mathematisches Institut
Universität Tübingen
Auf der Morgenstelle 10

7400 Tübingen

Prof. A. Shamir
Department of Mathematics
The Weizman Institut of Science

Rehovot
Israel

Prof. Dr. M. Sieveking
FB Mathematik
Universität Frankfurt
Robert-Mayer-Str. 6-10

6000 Frankfurt/Main

Dr. H.J. Stoß
Fakultät f. Mathematik
Universität Konstanz

7750 Konstanz

Prof. Dr. V. Strassen
Institut f. Angewandte Mathem.
Universität Zürich
Rämistr. 74

CH - Zürich

Prof. L. Valiant
Aiken Computation Lab.
Harvard University

Cambridge, Mass. 02138
USA

Dr. Hugo Volger
Math. Institut der Universität
Auf der Morgenstelle 10

7400 Tübingen

Prof. Dr. Ingo Wegener
Fachbereich Informatik
J.W.G. Universität Frankfurt
Postfach 11 19 32

6000 Frankfurt/Main

Prof. A. Yao
Stanford University
Computer Science Department

Stanford, CA 94305
USA

Prof. F. F. Yao
Xerox Research Lab.

Palo Alto, Calif. 94304
USA