MATHEMATISCHES FORSCHUNGSINSTITUT OBERWOLFACH

Tagungsbericht 33/1991

EFFIZIENTE ALGORITHMEN

4.8. bis 10.8.1991

Die Tagung fand unter der Leitung von Kurt Mehlhorn (Saarbrücken) und Robert E. Tarjan (Princeton) statt. Im Mittelpunkt des Interesses standen Fragen aus dem Bereich der effizienten diskreten Algorithmen. Behandelte Spezialgebiete waren u.a. Graphenalgorithmen, algorithmische Geometrie, Datenstrukturen und String Matching. Viele Ergebnisse waren auf dem Gebiet der dynamischen Algorithmen. Beispiele sind dynamische 2- und 3-edge-connectivity Probleme auf Graphen, dynamic point location Probleme und dynamic closest pair Probleme. Auch wurde Ergebnisse presentiert auf dem relativ neuen Gebiet der randomisierten Algorithmen. Solche Algorithmen haben den Vorteil daß sie einfacher sind im Vergleich zu deterministischen Algorithmen, sowohl was Implementierbarkeit als was Klarheit betrifft. Vor allem die algorithmische Geometrie und die Theorie der Datenstrukturen benutzen die Technik der Randomisierung intensiv. Es wurden aber auch klassische Probleme behandelt, wie die Berechnung der konvexen Hülle einer Punktmenge im d-dimensionalen Raum (ein optimaler Algorithmus wurde presentiert), die Analyse von Quicksort, usw. Für weitere Details der Ergebnisse wird auf die Vortragsauszüge verwiesen.

VORTRAGSAUSZÜGE

H. ALT

Measuring the Resemblance of Curves

A standard problem in shape or pattern recognition is to measure how much two given shapes "resemble each other". The Hausdorff-distance is appropriate in many cases but sometimes gives a poor estimate of the resemblance of curves. We therefore consider the Fréchet-metric, which is for curves P, Q defined as

$$\delta_F(P,Q) = \inf_{\alpha,\beta} \sup_t d(\alpha(t), \beta(t)),$$

where $\alpha, \beta : [0,1] \to R^2$ (or R^3) are monotone parametrizations of P, Q respectively. For P, Q being polygonal chains, we give an algorithm of runtime $O(pq)$ for solving the decision problem, whether for given P, Q, ϵ, it holds that $\delta_F(P,Q) \leq \epsilon$. ($p, q$ are the lengths of P, Q, respectively.) The problem of actually computing $\delta_F(P,Q)$ can be solved in time $O(pq \log^2(pq))$ using parametric search. The problem becomes somewhat easier, if the monotonicity condition in the definition of δ_F is dropped. (Joint work with Michael Godan, FU Berlin.)

T. ASANO

Walking on an Arrangement Topologically and Applications

Given a set of n lines and a convex s-gon R, it is not so hard to devise an algorithm for enumerating all the intersections among those lines within R in $O(K+s+n\log(n+s))$ time and $O(n+s)$ space, where K is the number of intersections. Here we are interested in a bit different problem, that is, we are required to find a tour which visits all the regions within R formed by those lines. The length of the tour must be linear in the number of regions. An algorithm which has the same computational complexity as above is presented. This is an improvement or extension of the well-known Topological Sweep Algorithm which sweeps the entire arrangement of lines in $O(n^2)$ time and $O(n)$ space. Our Topological Walk Algorithm can be applied to most of the problems which can be solved by Topological Sweep.

N. BLUM

On locally optimal alignments in genetic sequences

A substring $\tilde{x}$ of a text string x has c–locally minimal distance from a pattern string $y, c \in N \cup \{0\}$, if no other substring x' of x with smaller edit distance to y exists which overlaps $\tilde{x}$ more than c characters. We show how to compute all substrings of x which have c–locally minimal distance from y and all corresponding alignments in $O(m \cdot n)$ time where n is the length of x and m is the length of y.

B. CHAZELLE

Computing convex hulls optimally

I will present a deterministic algorithm for computing the convex hull of n points in d–space in $O(n^{\lfloor d/2 \rfloor})$ time, which is optimal.

J. CHERIYAN

Digraph K–vertex connectivity and directed rubber bands

Let $G = (V, E)$ be a directed graph and n denote $|V|$. We show that G is k–vertex connected iff for every subset X of V with $|X| = k$, there is an embedding of G in the $(k-1)$–dimensional space $R^{k-1}, f : V \to R^{k-1}$, such that no hyperplane contains k points of $\{f(v)|c \in V\}$, and for each $v \in V - X, f(v)$ is in the convex hull of $\{f(w)|(v,w) \in E\}$. This result generalizes to directed graphs the notion of convex embeddings of undirected graphs introduced by Linial, Lovász and Widgerson in "Rubber bands, convex embeddings and graph connectivity", Combinatorica 8 (1988) 91–102.

Using this characterization, a directed graph can be tested for k–vertex connectivity by a Monte Carlo algorithm in time $O((M(n)+nM(k))(\log n)^2)$ with error probability $< 1/n$, and by a Las Vegas algorithm in expected time $O((M(n)+nM(k))k\log n)$, where $M(n)$ denotes the number of arithmetic steps for multiplying two $n \times n$ matrices $(M(n) = O(n^{2.376}))$. Our Monte Carlo algorithm improves on the best previous deter-

ministic and randomized time complexities for $k > n^{0.19}$; e.g., for $k = \sqrt{n}$, the factor of improvement is $> n^{0.62}$. Let ϵ be any real > 0. Both algorithms have processor efficient parallel versions that run in $O((\log n)^2)$ time on the EREW PRAM model of computation, using a number of processors equal to $n^{0.5+\epsilon}$ times the respective sequential time complexities. Our Monte Carlo parallel algorithm improves on the number of processors used by the best previous (Monte Carlo) parallel algorithm by a factor of at least $n^{1.5-2\epsilon}$ while having the same running time.
Generalizing the notion of s-t numberings, we give a combinatorial construction of a directed s-t numbering for any 2-vertex connected directed graph.
(Joint work with John H. Reif.)

K. CLARKSON

Simpler algorithms and analysis for convex hulls

We give an algorithm for maintaining the convex hull of finite $R \subseteq S \subseteq E^d$. We analyze the performance of this algorithm when inserting and deleting points from R, when R is a random subset of S, and points are chosen for insertion or deletion from $S \setminus R$ or R at random. If conv(R) has expected f_r facets when $|R| = r$, insertion requires expected $\sum_{j \leq i} \frac{d(d-1)}{j(j-1)} f_j$ time and similar for deletion.
(Joint work with K. Mehlhorn and R. Seidel.)

P. F. DIETZ

Persistence and Randomization

I describe a family of combinatorial games arising in persistent data structures, in deamortization of data structures, and in some buffer-server problems. The games involve two players, A and B, who increment (decrement) a collection of n nonnegative real valued variables. Player A attempts to maximize the largest variable, B attempts to minimize it. In the simplest version of the game, in which A increments by a total of 1 and B may zero one variable, a bound of $H_{n-1} + 1$ is optimal.
For this game, I show that if player A is oblivious and B randomizes, the score is bounded by $\log \log n$ with high probability, with optimal play. I also describe a game on directed graphs with in and out-degrees bounded by constants. I show how randomization may be used to make parallel linked data structures persistent in constant slowdown with high probability.

M. DIETZFELBINGER

Some aspects of perfect hashing algorithms: simplicity, efficiency, reliability

We present a refined analysis of a universal hash class $\mathcal{R}$ (which was presented already at ICALP'90). Apart from its use for constructing a reliable probabilistic algorithm for a real-time dictionary, this class has some nice properties that makes it suitable for a variety of other applications that use universal hashing but require a certain performance *with high probability* rather than in the expected case. In contrast, we note that the popular class of linear hash functions has a performance that is probably inferior to that of class $\mathcal{R}$ and even to that of higher degree polynomials.

T. FEDER

Stability in Nonexpansive Metric Networks

A network is a collection of gates, each with many inputs and many outputs, where links join individual outputs to individual inputs of gates; the unlinked inputs and outputs of gates are viewed as inputs and outputs of the network. A stable configuration assigns values to inputs, outputs, and links in a network, so as to ensure that the gate equations are satisfied.

The problem of finding stable configurations in a network is computationally hard, even when all values are boolean and all input values are specified in advance; in general, the difficulty of a stability problem seems to depend on the kinds of gates present in the network. The study can be restricted to gates that satisfy a nonexpansiveness condition requiring small perturbations at the inputs of a gate to have only a small effect at the outputs of the gate. The stability question on the class of networks satisfying this local nonexpansiveness condition contains stable matching as a main example, and defines the boundary between tractable and intractable versions of network stability.

The structural and algorithmic study of stability in nonexpansive networks is based on a representation of the possible assignments of boolean values for a network as vertices in a boolean hypercube under the associated Hamming metric. This global view takes advantage of the median properties of the hypercube, and extends to metric networks, where individual values are now chosen from finite metric spaces and combined by means of an additive product operation. The relationship between products of metric spaces and products of graphs then establishes a connection between isometric representation in graphs and nonexpansiveness in metric networks.

G. FREDERICKSON

Ambivalent Data Structures for Dynamic 2–Edge–Connectivity and k Smallest Spanning Trees

A data structure possesses ambivalence if at each of many locations it keeps track of several alternatives, even when a global examination of the data structure would identify for each location the alternative (or valence) that is in fact valid. An example of such alternatives might be whether a path between vertices x and y in a spanning tree of a graph goes through a vertex w or through a vertex w'. We design data structures that are ambivalent with respect to the structure of a spanning tree as that spanning tree is being updated, and that yield faster algorithms for two problems. We give an algorithm to find the k smallest spanning trees of a weighted undirected graph in $O(m \log \beta(m,n) + min\{k^{3/2}, km^{1/2}\})$ time, where m is the number of edges and n is the number of vertices. For a planar graph, we give an algorithm that uses $O(n + k(\log n)^3)$ time. We also give data structures for maintaining an undirected graph under the operations of inserting and deleting edges and vertices, and answering queries about whether two given vertices are in the same 2–edge–connected component. We achieve an update time of $O(m^{1/2})$ and a query time of $O(\log n)$. For an embedded planar graph, these times are $O((\log n)^3)$ and $O(\log n)$, resp.

L. GUIBAS

Constructing Parts of Arrangements

Arrangements are ubiquitous structures in geometric computing. They are interesting in their own right in many applications, such as motion planning, but often also arise as intermediate structures in other problems. Invariably arrangements take a lot space to store so, whenever possible, it pays to construct only the relevant cell or cells (such as the cell containing a specific point, or the zone of a manifold). In this talk we show how to adapt the currently popular Randomized Incremental Constructions in geometry to deal with computing only parts of arrangements.

As examples we show how to compute a single face in an arrangement of n segments in the plane in time $O(n\alpha(n)\log n)$, and the zone of a line in a line arrangement in time $O(n \log n)$. These times are within $\log n$ of the space required by the corresponding structures. We also show how to deal effectively with parts of arrangements in divide and conquer applications. Often arrangements of manifolds related to the data defining a problem are used to partition the problem into subproblems of balanced complexity. In some applications we can establish a priori that we need to recurse down only a very small subset of the resulting cubes. As an example of these ideas, we show a deterministic $O(n^{1+\epsilon})$ algorithm for computing the diameter of n points in $\mathcal{R}^3$.

T. HAGERUP

Parallel Integer Sorting without Concurrent Writing

M general objects can be sorted in $O(M \log M)$ sequential time, and this performance has been matched in the realm of parallel computing, first by the celebrated algorithm of Ajtai, Komlós and Szemerédi (1983), which uses $O(\log M)$ time and $O(M)$ processors, i.e. $O(M \log M)$ operations. M integers of $O(\log M)$ bits each can be sorted in $O(M)$ sequential time (bucket sort + radix sort), but this has not been matched by any fast parallel algorithm. In particular, Cole and Vishkin (1986), Kruskal, Rudolph and Snir (1986) and Wagner and Han (1986) describe algorithms that work in polylogarithmic time on the exclusive—read exclusive—write (EREW) PRAM using $O(M \log M/ \log\log M)$ operations. We improve this result by showing that M integers of $O(\log M)$ bits each can be sorted in $O((\log M)^{3/2})$ time on an EREW PRAM using $O(M\sqrt{\log M}\log\log M)$ operations. Our algorithm is simple and uses the idea of storing $\sqrt{\log M}$ (small) integers in a single word.
(Joint work with Susanne Albers.)

M. JERRUM

An Analysis of a Monte Carlo Algorithm for Estimating the Permanent

Karmarkar, Karp, Lipton, Lovász, and Luby proposed a Monte Carlo algorithm for approximating the permanent of a non-negative $n \times n$ matrix, which is based on an easily computed, unbiased estimator. It is not difficult to construct 0,1-matrices for which the variance of this estimator is very large, so that an exponential number of trials are necessary to obtain a reliable approximation that is within a constant factor of the correct value. Nevertheless, the same authors conjectured that for almost every 0,1-matrix the variance of the estimator is small. The conjecture is shown to be true; indeed for almost every 0,1-matrix, $O(n\omega(n)\epsilon^{-2})$ trials suffice to obtain a reliable approximation that is within a factor $(1+\epsilon)$ of the correct value. Here $\omega(n)$ is any function tending to infinity as $n \to \infty$.

M. KAUFMANN

Edge-disjoint paths in planar graphs

An efficient algorithm for the edge-disjoint path problem in planar graphs is presented. Using G. Frederickson's decomposition method for planar graphs we can improve the best bound for the running time of $O(n^2)$ (due to Becker/Mehlhorn, Matsumoto/Nishozelzi/Saito) for the edge-disjoint path problem to $O(n^{5/3}(\log\log n)^{1/3})$. (Joint work with Gerhard Klär.)

R. KLEIN

Computing Voronoi diagrams for simple polygons

Let P be a simple polygon in the plane, and let S denote a set of vertices of P. Then the bounded Voronoi diagram of S in P is defined by saying that each point p in P belongs to the region of the closest vertex of S that is visible from p. If S includes all reflex vertices of P then the bounded Voronoi diagram equals the geodesic Voronoi diagram of S. This is assumed in the following. We show that in the Manhattan metric the bounded diagram of a rectilinear polygon can be computed in linear time. For the Euclidean metric we give a linear reduction to the case of rectilinear histograms.
(Joint work with Andrzej Lingas.)

H. LA POUTRÉ

Maintenance of Triconnected Components of Graphs

We consider the problem of maintaining 3-vertex-connected components of a graph under insertions of edges (and vertices), where k-vertex-connectivity is defined as follows. Let G be an undirected graph. Two nodes x and y are k-vertex-connected ($k \geq 1$) if there exist k different vertex-disjoint paths between them. We present a data structure with algorithms for maintaining the 3-vertex-connectivity relation of a graph. The algorithms start from a graph of n nodes and no edges in which edges are inserted one by one, and where at any time for any two nodes, the query whether these nodes are 3–vertex–connected can be answered. The solution has a total running time of $O(n + m\alpha(m,n))$, where m is the number of edge insertions and queries. The data structure allows for insertions of nodes also (in the same time bounds, taking n as the final number of nodes).

T. LENGAUER

Path Problems With General Cost Criteria

We consider single–source and all–pairs problems on directed graphs with edge labels that come from general cost structures. We generalize the Ford method to cost structures in which paths are compared on the basis of a partial (instead of a total) ordering. An axiomatic study shows, that this generalization applies to cost structures that are left monotonic and fulfill a certain requirement related to negative cycles. Then we extend the algorithmic framework to both nonmonotonic cost structures and cost structures with irregular behavior w.r.t. cycles. The latter generalization can be viewed as a generalization of the classical solution method for the algebraic path problem on closed semirings: We show how to eliminate the need for the associativity of $*$ and the distributivity of $*$ over $+$ in the closed semiring. Our results provide structured parameters of the cost structure that govern the efficiency of the path algorithms. We give efficient path algorithms for several example cost structures.

J. MATOUŠEK

Ham-sandwich cuts in 3 dimensions

For an n–point set $A \subseteq E^d$ and a hyperplane h we say that h bisects A if each of the open halfspaces defined by h contains $\leq \frac{n}{2}$ points. Given sets $A_1, \ldots, A_d \in E^d$ (d a small fixed integer) with n points in total, we seek a hyperplane h simultaneously bisecting all the A_i's. A linear–time algorithm for this problem in dimension 2 due to Lo and Steiger is outlined, and its generalization to dimension 3 with time complexity $O(n^{3/2}\log^2 n)$ is sketched, with empasis on topological aspects.

E. MAYR

Spanning Trees in Weighted Graphs

Let A_0 be the incidence matrix of an undirected graph with weighted edges and let B_0 be the same matrix, with the $\pm i$ entries replaced by $\pm x^{\omega(e_i)}$, where e_i is the corresponding edge and x an indeterminate. Then $\det(B_0^T A_0)$ is the generating function for the number of spanning trees by weight. This yields several pseudopolynomial algorithms for constructing a spanning tree of a given weight (if it exists), and a polynomial algorithm for counting the number of minimum spanning trees. We also study how one weight class of spanning trees can be reached from another by performing edge swaps. We show that from any MST some member of the i–th weight class can be reached by doing at most $i - 1$ swaps.
(Joint work with C. G. Plaxton.)

K. MEHLHORN

Dynamic Planar Point Location

We present a solution for the dynamic planar point location problem with query and insertion time $O(\log n \log\log n)$, deletion time $O((\log n)^2)$ and space $O(n)$.
(Joint work with H. Baumgarten and H. Jung.)

F. MEYER AUF DER HEIDE

Fast, time–processor optimal shared memory simulations on complete networks

We present a new randomized method for simulating a shared memory machine, i.e. a PRAM, on a distributed memory machine, i.e. a complete network. Previous methods, based on a direct hashing strategy, have the drawback that, for simulating an n–processor PRAM on an n–processor network an expected delay of $\Theta(\log(n)/\log\log(n))$ is inherent, for time–processor optimal simulation even expected delay $\Theta(\log(n))$ is inherent (time–processor optimal means that time $O(\log(n))$ is sufficient even to simulate an $n\log(n)$–processor PRAM). Our new method yields the following results:

- An n–processor PRAM can be simulated with expected delay $O(\log\log(n))$,
- An $n\log\log(n)\log^*(n)$ PRAM can be simulated with (time–processor optimal) expected delay $O(\log\log(n)\log^*(n))$

(Joint work with Richard Karp and Michael Luby.)

J. I. MUNRO

Deterministic Skip Lists

We develop a form of skip list that guarantees $\Theta(\lg n)$ search, insert and delete costs. The basic notion is to insist that between any pair of elements above a given height are a small number of elements of precisely that height. To guarantee the desired behaviour, we need only add the constraint that the physical sizes of nodes be exponentially increasing, a feature ideally suited to a buddy system of memory allocation. The technique is competitive in terms of time and space with balanced tree schemes, and, we feel, inherently simpler when taken from first principles.
(Joint work with Thomas Papadakis.)

Th. OTTMANN

Uniquely represented dictionaries

The dictionary problem asks for a family of data structures to store the sets of items and for algorithms to carry out the dictionary operations (search, insert, delete) efficiently. We call a dictionary set–uniquely represented, if each set of items is represented by a unique data structure. We call a dictionary size–uniquely represented, if each set of the same size is represented by the same structure. The unique representation problem for dictionaries asks for efficient algorithms for maintaining a set– or size–unique representation of dictionaries. We present two families of structures which lead to size–unique representations of dictionaries: The k–level jump lists and the shared search trees. The 3–level jump lists allow to carry out search, insert and delete operations in $O(n^{1/3})$ time in the worst case. We also state a general lower bound stating that for any unique representation of a set in a graph of bounded outdegree, one of the operations search or update must require a cost $\Omega(n^{1/3})$. These results shed some new light on previously claimed lower bounds for unique binary search tree representations of dictionaries.

M. H. OVERMARS

Hidden surface Removal

An important algorithmic problem in Computer Graphics is the hidden surface removal problem. Here we ask to compute the view of a scene in space as seen from a particular viewpoint. Most solutions used in Computer Graphics are so–called *image space* solutions, where for each pixel in the view image one computes the object in the scene that is visible at this pixel. Such techniques have efficient hardware implementations. They are though limited in their use. For example, they cannot be used very well when displaying a three–dimensional scene on a laser printer. The resolution of the laser printer tends to be too large to compute for each pixel the visible object. Hence, there is a need for so–called *object–space* solutions that compute a combinatorial representation of the visible scene. In this talk we review a number of recent object space hidden surface removal techniques.

R. REISCHUK

The Complexity of Fault–Tolerant Decision Trees

An ϵ–noisy Boolean decision tree is a decision tree with the property that each question of a Boolean variable might be answered incorrectly. This happens independently with probability ϵ. Such a tree computes a function $f : \{0,1\}^n \to \{0,1\}$ δ–reliably if for all inputs x the correct output $f(x)$ is produced with probability at least $1-\delta$. We show that for $0 < \epsilon, \delta < \frac{1}{2}$ almost all n–input functions require ϵ–noisy δ–reliable decision trees of depth at least $c_{\epsilon,\delta}\, n \log n$ for a fixed constant $c_{\epsilon,\delta} > 0$.
(Joint work with Bernd Schmeltz.)

U. RÖSLER

A limit theorem for 'Quicksort'

Let us consider an algorithm with enough randomisation within the algorithm. The analysis of the run time X_n may depend on the following structure: At level n we do some operations and then call the algorithm again several times at some lower level, the level may be random. In a first analysis using the above structure we obtain the asymptotic behavior of the expectation EX_n of X_n. More refined is the question of the derivation of X_n from EX_n. In this talk we give a general method, examplified by 'Quicksort', to derive the asymptotic distribution of X_n correctly normalized. A highlight is the statement 'Quicksort' is reliable, i.e.

$$\text{Prob}(X_n - EX_n > EX_n) \le \frac{\text{const}(\lambda)}{n^\lambda}$$

for all $\lambda > 0$ as a consequence of the above.

D. SLEATOR

Link Grammars

A link grammer is a new (contextfree equivalent) formal grammatical system in which it is convenient to encode natural language grammars. We have developed efficient algorithms for link parsing, written a program for general link parsing, and written a link grammar for the English language. We are encouraged by the speed and breadth of English phenomena captured by our system. (It currently can correctly parse most NY Times sentences.)
(Joint work with Davy Temperley.)

M. SMID

AN $O(n \log n \log\log n)$ algorithm for the on–line closest pair problem

Let V be a set of n points in k–dimensional space. It is shown how the closest pair in V can be maintained under insertions in $O(\log n \log\log n)$ amortized time, using $O(n)$ space. Distances are measured in the L_t–metric, where $1 \le t \le \infty$. This gives an $O(n \log n \log\log n)$ time on-line algorithm for computing the closest pair. The algorithm is based on Bentley's logarithmic method for decomposable searching problems. It uses a non–trivial extension of fractional cascading to k–dimensional space. It is also shown how to extend the method to maintain the closest pair during semi–online updates. Then, the update time becomes $O((\log n)^2)$, even in the worst case.
(Joint work with Christian Schwarz.)

G. STEIDL

Discrete Vandermone Transforms

Let $w(x)$ be a weight function, $P_n(x) \in \Pi_n$ $(n = 0, 1, \ldots)$ orthogonal polynomials with $\int_a^b w(x)P_n(x)P_m(x)dx = c_n v_{mn}$ and x_k $(k = 0, \ldots, N-1)$ the zeros of $P_N(x_k)$. Then the generalized Vandermonde–matrix $A := (P_n(x))_{n,k=0}^{N-1}$ is almost orthogonal, i.e. $A^{-1} = D_1 A^T D_2$ with regular diagonal matrices $D_{1/2}$. The corresponding transform $\hat{a} := Aa$ $(a \in \mathcal{R}^N)$ is called discrete Vandermonde transform (DVT). The most important DVT is the discrete cosine transform (DCT), which has found applications in digital signal processing and numerical analysis. Using this polynomial approach a new numerically stable $O(N \log N)$–algorithm to perform the DCT was presented. It can be generalized to other DVT's also in multidimensonal case.

E. UKKONEN

On approximate string matching in static texts

The problem of finding all approximate occurences P' of a pattern string P in a text string T such that the edit distance between P and P' is at most k is considered. We concentrate on a scheme in which T is first preprocessed to make the subsequent searches with different P fast. It will be shown how the standard dynamic programming procedure for edit distances can be applied on T represented as a suffix tree or suffix automaton. This leads to an algorithm with running time $O(mq+n)$ and to another one with running time $O(mq \log q)$, where n is the length of T and m is the length of P and $1 \leq q \leq n$ is a parameter dependent on the problem instance. A combination of these algorithms leads finally to a method with running time $O(mn)$ in the worst case and $O(m)$ in the best case.

P. VAN EMDE BOAS

Associative Storage modification machines

During the last 17 years a large variety of machine models belonging to the so–called second machine class have been proposed. These models are characterized by the property that their deterministic and nondeterministic polynomial time classes coincide and are equal to the class PSPACE from the sequential world. These models seem to embody the right combination of exponential growth power and uniformity. Examples are the Vector Machines of Pratt and Stockmeyer, the MRAM of Hartmanis and Simon, Goldschlagers SIMDAG, and our EDITRAM.
What seems to be nonexistent in the literature is a clean parallel version of the storage modification machine. Parallel versions proposed (going back to a model proposed by Barzdin already around 1965!) are too powerful; the nondeterministic version recogizes NEXPTIME in polynomial time, thus resembling an earlier machine model of Fortune and Wyllie.
We introduce the Associative Storage Modification Machine, which has the required behavior of the second machine class; it is a sequential device which obtains its power from the potential to traverse reverse edges in two key instructions from the standard SMM–instruction code. In this way the model can generate and manipulate sets of exponential size. The proof that the device is a true second machine class member is fully classic: a direct but contrieved P-time solution of QBF on an ASMM machine, combined with a PSPACE simulation of a nondeterministic ASMM computation on the basis of a recursive guess and verify strategy involving a computation trace.
(Joint work with J. Tromp.)

I. WEGENER

A solution and an optimal algorithm for the knight's Hamiltonian path problem on a chessboard

Is it possible for a knight to visit all squares of an $n \times n$ chessboard on an admissible path exactly once? The answer is yes if and only if $n \geq 5$. For $n \geq 5$ optimal sequential

and parallel algorithms for the computation of a Hamiltonian path for the knight are presented. A Hamiltonian path from a given source s to a given terminal t exists for $n \geq 6$ if and only if some easily testable color criterion is fulfilled. Hamiltonian circuits exist if and if $n \geq 6$ and n is even.
(Joint work with A. Conrad, T. Hindrichs and H. Morsy.)

E. Welzl

Linear Programming and Related Problems

A simple randomized algorithm is presented that solves Linear Programs with n constraints and d variables in expected $O(d2^d n)$ time. The algorithm is described in an abstract framework which makes it applicable to a large class of problems like smallest enclosing balls (or ellipsoids) for n points in d–space, largest ball (or ellipsoid) in a convex polytope in d–space etc.

D. E. WILLARD

Fusion Trees for Sorting, Searching and other applications

We have devised an $O(N \log N/\log\log N)$ worst–case time sorting algorithm and an $O(\log N/\log\log N)$ procedure for balance–tree search, insert and delete operations. These results are interesting because the Information Theoretic Lower Bound states that any algorithm employing only comparison operations will not run so quickly. Our procedures improve upon the Information Theoretic Lower Bound by utilizing the additional power of a computer's arithmetic and bitwise AND operations. These results are interesting because we do not cheat by using words of excessive bit–length or make any other unreasonable assumptions. Our algorithm also provides even greater $O(\sqrt{\log N})$ improvements when one either uses a randomized measure of time or slightly expands the memory space beyond $O(N)$. It also provides an $O(1)$ operation time for sets of polylog(N) size, a result that leads to improved algorithms for many problems, including most notably a linear algorithm for constructing the minimum spanning tree.
(Joint work with M. Fredman.)

Berichterstatter: Michiel Smid

Tagungsteilnehmer

Prof.Dr. Helmut Alt
Institut für Informatik (WE3)
Freie Universität Berlin
Arnimallee 2-6

W-1000 Berlin 33
GERMANY

Prof.Dr. Tetsuo Asano
Information Science Center
Osaka Electro-Commun. University
18-8 Hatsu-cho, Neyagawa

Osaka 572
JAPAN

Dr. Norbert Blum
Institut für Informatik IV
Universität Bonn
Römerstr. 164

W-5300 Bonn 1
GERMANY

Prof.Dr. Bernard Chazelle
Department of Computer Science
Princeton University

Princeton , NJ 08544
USA

Prof.Dr. Joseph Cheriyan
c/o Prof. R. G. Bland
School of OR + Industr. Engineer.
Cornell University
Upson Hall

Ithaca , NY 14853-7501
USA

Dr. Ken Clarkson
AT & T
Bell Laboratories
Room 2C-455
600 Mountain Avenue

Murray Hill , NJ 07974-2070
USA

Prof.Dr. Paul Dietz
Dept. of Computer Science
University of Rochester

Rochester , NY 14627
USA

Dr. Martin Dietzfelbinger
FB17, Mathematik/Informatik
Universität Paderborn
Warburger Str. 100
Postfach 1621

W-4790 Paderborn
GERMANY

Prof.Dr. Peter van Emde Boas
Fakulteit Wiskunde en Informatica
Universiteit van Amsterdam
Plantage Muidergracht 24

NL-1018 TV Amsterdam

Prof.Dr. Tomas Feder
Bell Communications Research Center
Room 2M-347
445, South Street

Morristown , NJ 07960-1910
USA

Prof.Dr. Greg N. Frederickson
Department of Computer Sciences
224, Computer Science Building
Purdue University

West Lafayette , IN 47907
USA

Prof.Dr. Rolf-Ulrich Klein
Fachbereich Mathematik und
Informatik - Dekanat -
Fernuniversität Hagen
Postfach 940

W-5800 Hagen
GERMANY

Prof.Dr. Leo J. Guibas
Digital Equipment Corporation
Systems Research Center
130 Lytton Avenue

Palo Alto , CA 94301
USA

J.Han A. La Poutré
Department of Computer Science
University of Utrecht
P. O. Box 80. 089
Padualaan 14

NL-3508 TB Utrecht

Dr. Torben Hagerup
Max-Planck-Institut für Informatik
Im Stadtwald

W-6600 Saarbrücken
GERMANY

Prof.Dr. Thomas Lengauer
FB17, Mathematik/Informatik
Universität Paderborn
Warburger Str. 100
Postfach 1621

W-4790 Paderborn
GERMANY

Prof.Dr. Mark R. Jerrum
Department of Computer Science
University of Edinburgh
Mayfield Road

GB- Edinburgh EH9 3JZ

Prof.Dr. Jiri Matousek
Department of Applied Mathematics
Charles University
Malostranske nam. 25

118 00 Praha 1
CZECHOSLOVAKIA

Dr. Michael Kaufmann
Max-Planck-Institut für Informatik
Im Stadtwald

W-6600 Saarbrücken
GERMANY

Prof.Dr. Ernst W. Mayr
Fachbereich 20 - Informatik
Universität Frankfurt
Postfach 11 19 32

W-6000 Frankfurt 1
GERMANY

Prof.Dr. Kurt Mehlhorn
Fachbereich Informatik
Universität des Saarlandes
Im Stadtwald 15

W-6600 Saarbrücken 11
GERMANY

Prof. Michael S. Paterson
Department of Computer Science
University of Warwick

GB- Coventry CV4 7AL

Prof.Dr. Friedhelm Meyer auf der Heide
FB 17: Mathematik - Informatik

Universität Paderborn
Warburger Str. 100

W-4790 Paderborn
GERMANY

Dr. Rüdiger Reischuk
Institut für Theoretische Informatik
Technische Hochschule Darmstadt
Alexanderstr. 10

W-6100 Darmstadt
GERMANY

Prof.Dr. Ian Munro
Dept. of Computer Science
University of Waterloo

Waterloo, Ontario N2L 3G1
CANADA

Prof.Dr. Uwe Rösler
FB 6 - Mathematik/Informatik
Universität Oldenburg
Carl-von-Ossietzky-Str.
Postfach 2503

W-2900 Oldenburg
GERMANY

Prof.Dr. Thomas Ottmann
Institut für Informatik
Universität Freiburg
Rheinstr. 10

W-7800 Freiburg
GERMANY

Prof.Dr. Daniel D. Sleator
School of Computer Science
Carnegie Mellon University

Pittsburgh , PA 15213
USA

Prof.Dr. Mark H. Overmars
Department of Computer Science
University of Utrecht
P. O. Box 80. 089
Padualaan 14

NL-3508 TB Utrecht

Dr. Michiel Smid
Max-Planck-Institut für Informatik
Im Stadtwald

W-6600 Saarbrücken
GERMANY

Dr. Gabriele Steidl
Fachbereich Mathematik
Universität Rostock
Universitätsplatz 1

O-2500 Rostock
GERMANY

Prof.Dr. Ingo Wegener
Institut für Informatik II
Universität Dortmund
Postfach 50 05 00

W-4600 Dortmund 50
GERMANY

Prof.Dr. Robert E. Tarjan
Department of Computer Science
Princeton University

Princeton , NJ 08544
USA

Prof.Dr. Emo Welzl
Institut für Informatik (WE3)
Freie Universität Berlin
Arnimallee 2-6

W-1000 Berlin 33
GERMANY

Prof.Dr. Esko Ukkonen
Dept. of Computer Science
University of Helsinki
Teollisuuskatu 23

SF-00510 Helsinki

Prof.Dr. Dan E. Willard
LI 67-A, Dept of Computer Science
State University of New York
at Albany
1400 Washington Avenue

Albany , NY 12222
USA