Hartmut Noltemeier, Thomas Ottmann,
Derick Wood (editors):

Data Structures

Dagstuhl-Seminar-Report; 25
4.11.-8.11.91 (9145)

ISSN 0940-1121

Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

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: office@dag.uni-sb.de

Report of the Dagstuhl Seminar on

DATA STRUCTURES

November, 4–8, 1991

Organized by:

Hartmut Noltemeier (Universität Würzburg)
Thomas Ottmann (Universität Freiburg)
Derick Wood (University of Waterloo, Ontario)

Overview

The first Dagstuhl seminar on Data Structures was organized by Hartmut Noltemeier (Universität Würzburg), Thomas Ottmann (Universität Freiburg) and Derick Wood (University of Waterloo, Ontario).
It was a pleasure to recognize the international acclaim the seminar has found; the 30 participants came from 10 countries.
29 contributions were given: 27 talks on various topics within the field of data structures,
1 software demonstration and a very lively open-problems-session on Tuesday night.

Although the field of data structures is by now a classic one and at least as old as Computer Science itself, it just happened to have a renaissance and a rapidly growing number of challenging questions. This is partially due to challenges coming from distributed environments, object oriented programming paradigm and the need for persistent data structures as well as from problems of knowledge representation, but especially from new advanced applications too.

This seminar brought together a lot of researchers to focus upon these topics and to exchange new results and questions. Everybody was impressed by the quality of the presentations, a major part of them with exciting new results. They acknowledged the importance of such research exchange at Schloß Dagstuhl.

Abstracts

Uniquely Represented Dictionaries: Lower Bound
Arne Andersson
University of Lund, Sweden

We give a tight lower bound for uniquely respresented dictionaries. For any unique representation of a set in a graph of bounded outdegree, one of the operations search or update must require a cost of $\Omega(n^{1/3})$.

Lexicographic lists for parallel string manipulations
Alberto Apostolico
Purdue University, West Lafayette, Ind., USA/ Universita di Padua

Lexicographic lists are introduced as a tool for achieving fast parallel solutions to a number of problems on strings. A string given together with its associated lexicographic list is considered to be in standard form. Putting a string w in standard form regards O(|w| log |w|) work and O(log |w|) time on a CRCW PRAM. The following applications are considered in particular.
1) Given a string x, it is possible to test whether or not x is square-free in overall O(|x| log |x|) (resp., O(|x|)) work and O(log |x|) time if the alphabet of x is unbounded (resp., bounded). It is also possible to detect all the squares in x in overall O(|x| log |x|) work and O(log |x|) time. All bounds are optimal.
2) Let W be a collection of strings individually given in standard form. Let w be an arbitrary string in W, w' an arbitrary substring of w, and $\{\overline{w}_1, \overline{w}_2, \ldots \overline{w}_t\}$ an arbitrary set of substrings of strings in W. Then a CRCW PRAM with $O(\overline{n} = \sum_{h=1}^{t} |w_h| + |w'|)$ processors will find all the occurences of w' in $\{\overline{w}_1, \overline{w}_2, \ldots \overline{w}_t\}$ in constant time.
The problems discussed in connection with lexicographic lists do not assume or imply any notion of alphabetic order. The proposed solutions show that, for some problems in this class, assuming or imposing an arbitrary order on the input alphabet leads to more efficient solutions than could be achieved otherwise.

Some results on approximate string matching
Ricardo Baeza-Yates
Universidad de Chile, Santiago

We present several results on how to search for a pattern p in a set of strings S, finding all strings $s \in S$ such that d(P,s) ≤ k, where d is a distance function over strings and k is the maximal error allowed. These algorithms are based in simple ideas and they are fast in practice.
In particular we describe:

(i) A O(cn) expected time algorithm to search in a text of length n for $k \leq {}^{m}/_{\log|\epsilon|}({}^{m}/_{c})$ where c is a constant.
(ii) If we preprocess S, building a suffix array, we can simulate the standard dynamic programming algorithm achieving $O(n^{\alpha})$ expected time ($\alpha < 1$). The same idea can be applied to match a text against itself (joint work with G. Gonnet, ETH Zurich).
(iii) If S is a set of words (dictionary), a possible index is a Burkhard-Keller tree. Expected analysis of this structure show that the expected height & search time is O(log n), and partial match searches (approximate search) and nearest neighbor search requires $O(n^{\alpha})$ time ($\alpha < 1$) (joint work with Walter Cunto, IBM Venezuela).

Lower bounds for Heap construction
Svante Carlsson
Lulea Technical University, Sweden
(joint work with Jingsen Chen, Lund University, Sweden)

We study the lower bound for the heap construction problem. We show, after discussing previous lower bounds, the first adversary based lower bound, that does not give a trivial result. Our technique gives a lower bound of 1,5 (n + 1) — log (n + 1) − 2 comparisons.

Efficient use of randomness in perfect hashing schemes
Martin Dietzfelbinger
Universität (GH) Paderborn

We consider the problem of universal hashing (as introduced by Carter and Wegman), using few random bits, and constructing perfect hash functions by the scheme of Fredman, Komlós, Szemerédi, again using few random bits.

(i) Universal hashing. Let U be a finite set (the universe), m a natural number (the table size), $c > 0$. A class $\mathcal{H}$ of functions from U to [m] is c-universal if for distinct $x,y \in U$, if $h \in \mathcal{H}$ is chosen randomly, then $Pr(h(x) = h(y)) \leq c/m$. Mehlhorn's construction of a c-universal class that uses the minimum possible number of random bits ($O(\log\log|U| + \log m)$ used random bits. We argue against the use of random primes in practical algorithms and describe a new, very simple class of hash functions that has the same performance as Mehlhorn's class but does not use random primes: Given $U = \{0,1\}^k$, let $l = \log k + \log (m/\epsilon)$, $r = k/l$, and let $p > 2^l$ be a fixed prime. Split $x \in U$ into substrings $x_0, ..., x_{r-1}$ of l bits each. Choose $a, \alpha, \beta \in \{0, ..., p-1\}$ randomly. Let
$h_{\alpha, \beta, a}(x) = ((\alpha \times \sum_{0 \leq i < r} x_i a^i + \beta) \bmod p) \bmod m$.
Claim: If $x = y$ then $Pr(h_{\alpha, \beta, a}(x) = h_{\alpha, \beta, a}(y) \leq (1+\epsilon)/m$.

(ii) Perfect hashing. We show how to use the 2–independent sampling technique by Chor and Goldreich to reduce the number of random bits needed by a probabilistic algorithm for constructing a perfect hash function for a set $S \subseteq U$, $|S| = n$. We show that $O(\log\log|U| + \log n)$ random bits suffice to obtain an algorithm with expected running time O(n).

Randomized Adaptive Sorting
Vladimir Estivill-Castro
York University, Ontario, Canada
(joint work with Derick Wood, University of Waterloo)

Our goal is to design practical sorting algorithms that require time proportional not only to the size of the input but also to the disorder in the input. Such sorting algorithms are said to be *adaptive*. We introduce randomization to achieve this goal; the first time randomization has been used to obtain adaptive sorting algorithms. We investigate three randomized algorithms:

- *Randomized Merge Sort,* which is expected *Runs*-optimal;
- *Randomized Quicksort,* which is *Exchange*-sensitive; that is, it takes $\Theta(|X|[1 + \log(Exc(X) + 1)])$ time in the expected case; and
- *Skip Sort,* which is *Inversions-*, *Runs-*, and *Exchange*-optimal.

The three sorting algorithms are simple and practical, in contrast to previous adaptive sorting algorithms that used complex data structures. Moreover, previous claims about the performance of adaptive variants of *Quicksort* were based only on simulation results, our claims are based on a formal analysis.

Hierarchical Computation of Spatial Joins
Oliver Günther
FAW Ulm

Spatial joins are join operations in the relational sense, typically denoted by $R \bowtie_{i\Theta j} S$, where at least one of the colums involved is of some spatial data type, and Θ is a spatial operator. Typical spatial data types are point, line, or polygon, and spatial operations may include "containment" or "overlap". We discussed in our talk, which join processing strategies common in standard (non-spatial) relational databases can be used to compute spatial joins as well. The main problem here is that many of these strategies are based on some kind of sorting the rows of each relation, which is obviously somewhat more difficult in the presence of spatial data. In particular, the common sort-merge-strategy is not applicable in this context. Index-based join strategies are much more promising: we presented a class of tree structures that can be used to compute spatial joins in a hierarchical manner. This class includes common spatial database indexes as the R-tree as well as application-specific spatial hierarchies.
Finally, we can enhance this hierarchical join strategy by so-called local join indices that take advantage of the proximity properties of many common spatial join operators. A simulation model and preliminary performance analysis concluded our talk.

A plane-sweep algorithm for finding a closest pair among convex planar objects
Klaus Hinrichs
Universität Münster
(joint work with Frank Bartling, Universität Siegen)

Given a set of geometric objects a closest pair is a pair of objects whose mutual distance is smallest. We present a plane-sweep algorithm which finds a closest pair with respect to any L_p—metric, $1 \leq p < \infty$, for planar configurations of n (possibly intersecting) compact convex objects such as line segments, circular discs and convex polygons. For configurations of line segments or discs the algorithm runs in asymptotically optimal time $O(n \log n)$. For a configuration of n convex m-gons given in a suitable representation it finds a closest pair with respect to the Euclidean metric L_2 in time $O(n \log(n \times m))$.

Stable in-place quicksort
Jyrki Katajainen
University of Copenhagen and Turku
(joint work with Tomi Pasanen, University of Turku, Finland)

In the stable 0–1 sorting problem the task is to sort an array of n keys with two distinct key values such that equal keys are output in the same relative order as they were input. Recently, Munro, Raman and Salowe [BIT, 1990] gave an algorithm which solves this problem in O(n log*n) time and constant extra space. We show that by a slight modification of their method the stable 0–1 sorting is possible in O(n) time and O(1) extra space. This will immediately give us a stable, in-place implementation for quicksort.

"The Big Sweep"
Rolf Klein
Fernuniversität Hagen

The first sweep line algorithm for computing the Euclidean Voronoi diagram of n points in optimal O(n log n) time was invented by S. Fortune. He used a bending transformation in order to make sure that each site is the first point of its Voronoi region to be hit by the sweepline. Later, R. Seidel and R. Cole suggested a simplification. The transformation can be avoided if one maintains that part of the Voronoi diagram which does not change anymore, as new sites are detected; this is just the Voronoi diagram of the sweepline itself, and of all points "behind" it.
We show that the same approach — with some modifications — works for all nice metrices in the plane. Here, a metric d is called nice if convergency with respect to d is equivalent to Euclidean convergency, if for any two points a third point "between" them exists, and if the boundary of all bisectors consists of lines, up to homeomorphism. This result is based on joint work with F. Dehne, Carleton University.

Fast, time-processor optimal shared memory simulations
Friedhelm Meyer auf der Heide
Universität (GH) Paderborn
(joint work with Richard Karp and Michael Luby, Berkeley, CA)

We present a randomized simulation of a n loglog (n)log* (n)-processor shared memory machine (PRAM) on a n-processor distributed machine with n memory modules. The simulation guarantees optimal expected delay O(loglog (n)log* (n)). This time bound is very reliable, it holds with overwhelming probability. The algorithm is based on a novel hashing strategy using a parallel perfect hashing scheme for intermediate storage of unsatisfied write requests.
The well known direct implementations of hashing-based simulations are much slower: the previously most efficient simulations have delay Θ(log (n)/loglog (n)), if an n-processor-PRAM is simulated on an n-processor DMM, and the best known time-processor optimal simulation simulates an n log (n)-processor PRAM on an n-processor DMM with expected delay O(log (n)).

Selection and Related Problems
Ian Munro
Princeton University and University of Waterloo, Canada

Selection, or finding the median, is a problem in comparison based complexity that has been seriously studied for more than 30 years. We survey algorithms and lower bounds on the problem, for both the worst case and the average case. Attention will then turn to the related problem of finding elements in an unordered array that are of rank immediately above and below a given value.

LEDA
A Library of Efficient Data Types and Algorithms
Stefan Näher
Max-Planck-Institut Informatik, Saarbrücken

LEDA is a library of C++ software components that makes the results (data structures and algorithms) from the area of combintorial computing available to non-expert users, teaching and experiments. The main features of the library are

- precise and readable specifications by ADTs
- efficient and up-to-date implementations
- very comfortable graph and network data types.

LEDA is designed to narrow the gap between algorithms research and implementation.

Similarity and Proximity in Sets
Hartmut Noltemeier
Universität Würzburg
(joint work with K. Verbarg and C. Zirkelbach, Universität Würzburg)

The talk centers around the representation of sets (of complex objects) including their similarity- and/or proximity properties.
The goal is a compact representation as well as the support of various kinds of operations. We shortly review bisector trees, Voronoi trees and introduce monotone bisector trees, showing some major advantages of this class of trees. Variants of this class (MB*T) are efficient tools f.e. for the compact representation and efficient partitioning of complex scenes of geometric objects, and we are showing additionally that by this single data structure a lot of different kinds of operations can be supported very well. We report on extensive experimental results and point to some topics of further interest.

Uncoupling Updating and Rebalancing in Chromatic Binary Search Trees

Otto Nurmi
University of Helsinki, Finland
(joint work with Eljas Soisalon-Soininen, University of Helsinki)

In order to gain maximal efficiency of concurrent use of search trees the number of nodes to be locked at a time schould be as small as possible, and the locks should be released as soon as possible. We propose a new rebalancing method for binary search trees that allows rebalancing to be uncoupled from updating, so as to make updating faster. The trees we use are obtained by relaxing the balance conditions of red-black trees. When not involved with updating, the rebalancing task can be performed as a shadow process being active all the time, or it can be performed outside rush hours, at night, for example.

Uniquely Represented Dictionaries: Upper Bounds

Thomas Ottmann
Universität Freiburg

The dictionary problem asks for a family of data structures to store a set of items and for algorithms to carry out search, insert and delete operations efficiently. We consider graphs storing the items in its nodes. We call a dictionary set-uniquely represented, if each set of items is represented by a unique graph. A dictionary is represented size-uniquely, if each set of the same size is represented by the same structure. We also assume that the values stored in the nodes of a graph representing a dictionary are constraint by a fixed (for any graph) total order. We present two families of structures which are size- and order-unique representations of dictionaries. The first family — the k-level jump-lists — are an extension of the jelly-fish structure by L. Snyder. It allows to carry out searches in time $O(k \times n^{1/k})$ and updates in time$O(k \times n^{(k-2)/k})$; it requires space $O(k \times n)$. The second structure — the shared search tree — allows to carry out searches in time O(log n) and updates in time $O(\sqrt{n})$; it requires space O(n log n).
Our upper bound constructions "beat" some previously stated lower bounds for the uniquely represented dictionaries. That is , the computational models of the upper and lower bounds appearing in the literature do not fit together.
Our results also shed some new light on the general question of how many structures are required for a dictionary of a given size in order to allow fast searches and updates.

Data Structures in a new model of computation

Linda Pagli
Università di Pisa
(joint work with Fabrizio Luccio, Pisa)

We introduce a new sequential model of computation called Logarithmic Pipelined Model (LPM), in which a RAM processor of fixed size has pipelined access to a memory of m cells in time log m. Our motivation is that the usual assumption that a memory can be accessed in constant time becomes theoretically unacceptable as m increases, while an access time of log m is consistent with VLSI technologies. For a problem π of size n, $\pi \in P$, we denote by S(n) the time required by the fastest sequential algorithm and by T(n) the time required by the fastest sequential algorithm solving π in LPM. Letting O(log m) = O (log n), we define the classes:
$LP_0 = \{\pi \in P: T(n) \in O(S(n)\}$, $LP_\infty = \{\pi \in P: T(n) \in O(S(n)\log n)\}$.
LP_0 is the class of problems for which LPM is as efficient as the standard sequential model, while problems in LP_∞ are less adequate to be solved in the new model. We discuss several problems belonging to the above classes and derive the relative lower bound. We discuss important relations between parallel and LPM algorithm and finally we derive an optimal PRAM algorithm for the first ranking problem for a list of n elements stored by pointers in m memory cells, $m \geq n$. This problem was previously solved in parallel only for the case m = n.

Implementing Insertion Sort

Ola Petersson
Lund University, Sweden
(joint work with Alistair Moffat, Melbourne University, Australia)

A sorting algorithm is adaptive if sequences that are initially "close" to sorted are processed faster than those that are not, where the "distance" from totally sorted is quantified by some measure of presortedness.
In this talk we investigate the adaptivity of sorting algorithms resulting from various implementations of the well known sorting by insertion paradigm. Starting from textbook variants we gradually employ more sophisticated data structures. The final algorithm is something of a "swiss army knife" for adaptive sorting, in that it adapts to all measures of presortedness proposed in the literature.

String searching algorithms under Markovian dependencies
Mireille Regnier
INRIA, France

We presented a general framework to derive average performaces of string searching algorithms such as Knuth-Morris-Pratt and Boyer-Moore. It relies mainly on languages and combinatorics on words, joined to probabilistic tools. We assumed a Markovian distribution, suitable for applications such as natural languages or biological data bases searching. This allows to prove expected linearity and derive the linearity constant.

Dynamic Voronoi Diagrams and Applications
Thomas Roos
Universität Würzburg

In the last three years, the development of a new data structure in dynamic computational geometry attracted an increasing number of researchers: the dynamic Voronoi diagram. To give an intuition of this notion, consider a finite set of points in the Euclidean plane each of which is continuously moving along a given trajectory. At each instant of time, the points define a Voronoi diagram. As the points move, the Voronoi diagram changes continuously, but at certain critical instants, topological events occur that cause a change in the topological structure.
We summarize various results which have been obtained on this topic presenting also some most recent generalizations with respect to the dimension, the order of the Voronoi diagram, and the underlying objects. In addition, the "methodology of dynamization" can be applied to many other geometric data structures.
Application areas include motion planning and region location, as well as pattern recognition and placement problems in dynamic scenes.

Implementing plane-sweep: a case study
Peter Schorn
ETH Zürich

We consider as an example the Bentley-Ottmann plane sweep for finding all intersections among line-segments and study three implementation problems:
1) Access to the y-table is difficult in the presence of entries with the same key-value. As a solution we use the reference concept which attaches a unique reference to each entry. This idea separates key-based operations ("find") from data structure manipulations (e.g. "swap", "delete") and increases efficiency.
2) A priority queue with delete is needed. We show how a heap in its standard array representation can be modified to support an efficient "delete" operation.
3) High precision requirements for exact arithmetic. We present a topological argument to reduce the required precision from five-fold to three-fold. Experimental results, obtained with the XYZ GeoBench, show the superiority of the heap with delete over a balanced tree implementation of a priority queue.

A Data Structure For Shortest Rectilinear Path Queries in a Simple Polygon
Sven Schuierer
Universität Freiburg

We present a data structure that allows to preprocess a simple rectilinear polygon in the plane such that shortest path queries in the rectilinear link metric for any two query points can be answered efficiently. Efficiently meaning that if the two query points are vertices of the polygon, then the rectilinear link distance can be computed in time O(1) and a shortest path can be reported in time O(1+l), where l is the distance between the two vertices; if the two points are in some arbitrary location in the polygon, then O(log n) time is needed to find their distance and a shortet path can then again be constructed in time proportional to its length. The data structure needs linear time preprocessing and linear space.
The main idea in its construction is to partition the polygon into histograms and associate a tree T with this partition such that shortest path queries can be transformed into nearest common ancestor queries in T.

The Transformation Technique for Spatial Objects Revisited
Hans-Werner Six
Fernuniversität Hagen
(joint work with B.-U. Pagel, Hagen)

The transformation technique was one of the first approaches for storing a set of iso-oriented rectangles such that insertion, deletion and proximity queries could be carried out with reasonable performance. The basic idea is to transform rectangles into points in higher dimensional space in order to apply data structures for points which are better understood and easier to handle. When this technique came up it was regarded as less appropriate for proximity queries because proximity is lost in the image space. Furthermore, the distributions of the resulting points tend to be highly biased and point data structures of that time could not cope with that efficiently.
We show that the transformation technique can be made competitive to any other data structure maintaining set of rectangles. One reason is due the LSD tree, a point data structure which is able to handle the image point distribution efficiently. Second reason is that important geometric properties of domain space are carried over and drive the actions in the image space. Furthermore, we show that new kinds of transformations which take other than pure location parameters into account, e.g. the volume, the longer side length of rectangles, etc, lead to further improvements and provide the transformation technique with new quality.

Still another suffix-tree construction algorithm
Esko Ukkonen
University of Helsinki, Finland

An on-line algorithm is presented for constructing the suffix-tree for a given string in time linear in the length of the string. The algorithm processes the string in one left-to-right-scan and has always the suffix-tree for the scanned portions of the string ready. The method can be understood as a linear-time version of a very simple algorithm for (quadratic size) suffix-tries. This latter method is similar to the position-tree algorithm of Kempf, Bayer & Güntzer (1987).
The suffix-tree algorithm is further modified to solve the adaptive dictionary matching problem as recently proposed (and solved) by Amir & Farach (1991). This problem asks for maintaining a dictionary of key-words under key-word insertions and deletions, and for finding the occurences of key-words in text strings. A solution is given such that insertions and deletions are performed in time linear in the length of the key-word. The occurences of the current key-words in a text of length n can be found in time $O(k \times n)$ where k is the length of the longest key-word. The algorithm maintains the compacted version of the Aho-Corasick string-matching automaton for all suffixes of the key-words in the dictionary.

The Knight's Hamiltonian Path Problem
Ingo Wegener
Universität Dortmund
(joint work with Axel Conrad, Tanja Hindrichs)

The 400 years old Hamiltonian path problem for a knight on an $n \times n$ chessboard is solved. A Hamiltonian path exists iff $n \geq 5$ and a Hamiltonian circuit exists iff $n \geq 5$ and n is even. If the source s and terminal t are given, an s-t Hamiltonian path exists for $n \geq 6$ iff n is odd and s and t are white squares or n is even and s and t have different colours.
For $n = 5$ one has to add an additional condition. In the affirmative case s-t-Hamiltonian paths can be computed in optimal time, e.g. $O(n^2)$ sequential time and O(1) parallel time by n^2 processors.

Global order makes spatial access faster
Peter Widmayer
Universität Freiburg
(joint work with A. Hutflesz, C. Zimmermann, Universität Freiburg)

We present a secondary storage access scheme for geometric objects in a dynamic environment that aims at minimizing disk seek operations for spatial proximity queries. This goal is achieved by preserving the natural spatial order of objects on the secondary storage medium not only within blocks, but to some extend also beyond block boundaries. It turns out that window queries can be answered extremely fast as compared with schemes that do not take disk seeks into account.

GRAIL: A manipulation system for grammars, (regular) expressions, and automata
Derick Wood
University of Waterloo, Canada

A system, GRAIL II, is currently being implemented at Waterloo as the latest successor of INR & GRAIL that were implemented over the last ten years. It is written in C++ and provides finite automata and regular expressions at present. Apart from transformation operations it includes union, catenation, intersection, minimization, dfa construction, ... Each operation is written as a single process and they are run in a loosely-coupled environment via message passing. The challenge for algorithm and data structure researchers is to design and develop real algorithms, rather than textbook algorithms, that are simple to code and are extremely efficient. As an example we discussed nfa-dfa conversion and sketched an algorithmic approach based on multiway merging. The investigation of this algorithm leads to a conjecture for random nfa; namely, if their deterministic density is outside the range 1,5 ... 2,5, there is only a small-polynomial blow-up in the number of states in the dfa, and, conversely, within the range there is a high probability, that there is an exponential blow-up. The proof of this conjecture is eagerly awaited.

Dagstuhl-Seminar 9145

Participants

Arne **Andersson**
Lund University
Dept. of Computer Science
Box 118
22100 Lund
Sweden
tel.: +46-46-108036
arne@dna.lth.se

Alberto **Apostolico**
Universita di Padova
Dipartimento di Elettronica e Informatica
Via Gradenigo 6/A
35131 Padova
Italy
tel.: +39-49-828 7500
axa@sabrina.dei.unipd.it

Ricardo **Baeza-Yates**
Universidad de Chile
Depto. de Ciencias de la Computacion
Blanco Encalada 2120
Casilla 2777
Santiago
Chile
tel.: +56-2-6892736
rbaeza@dcc.uchile.cl

Svante **Carlsson**
Luleå Technical University
Department of Computer Science
95187 Luleå,
Sweden
Svante.Carlsson@sm.luth.se

Janos **Csirik**
Joseph-Attila-University
Department of Applied Computer Science
Arpad tér 2
6720 Szeged
Hungary
h873csi@ella.hu

Martin **Dietzfelbinger**
Universität - GH - Paderborn
FB 17 - Mathematik/Informatik
Warburger Str. 100 Postfach 1621
W-4790 Paderborn
tel.: +49-5251-60 3308
M.Dietzfelbinger@uni-paderborn.de

Vladimir **Estivill-Castro**
York University
Department of Computer Science
4700 Keele Street
North York Ontario M3J-1P3
Canada
vlad@sasquatch.cs.yorku.ca

Oliver **Günther**
FAW Ulm
Helmholtzstr. 16 Postfach 2060
W-7900 Ulm
GUENTHER@DULFAW1A.bitnet

Klaus **Hinrichs**
Universität Münster
FB 15 - Informatik
Einsteinstr. 62
W-4400 Münster
hinrichs@ti.e-technik.uni-siegen.dbp.de
(until 31.3.92)

Ulrich **Huckenbeck**
Lehrstuhl für Informatik I
Universität Würzburg
Am Hubland
W-8700 Würzburg
tel.: +49-931-888 5025
hu@informatik.uni-wuerzburg.de

Jyrki **Katajainen**
University of Copenhagen
Department of Computer Science
Universitetsparken 1
2100 Copenhagen East
Denmark
jyrki@diku.dk

Rolf **Klein**
Fernuniversität Hagen
Praktische Informatik VI
Elberfelder Straße 95
W-5800 Hagen 1
tel.: +49-2331-804 8365
klein@fernuni-hagen.de

Friedhelm **Meyer auf der Heide**
Universität GH Paderborn
FB 17 - Mathematik/Informatik
Warburgerstr. 100 Postfach 1621
W-4790 Paderborn
tel.: +49-5251-6003310
fmadh@uni-paderborn.de

Ian **Munro**
Princeton University
Departement of Computer Science
35 Olden Street
Princeton NJ 08544
USA
imunro@waterloo.ca

Stefan **Näher**
Max-Planck-Institut für Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11
tel.: +49-681-302 5420

stefan@mpi-sb.mpg.de

B. **Nilsson**
Universität Freiburg
Institut für Informatik
Rheinstraße 10-12
W-7800 Freiburg

Hartmut **Noltemeier**
Universität Würzburg
Lehrstuhl für Informatik I
Am Hubland
W-8700 Würzburg
tel.: +49-931-888 5054 (5055)
noltemeier@informatik.uni-wuerzburg.de

Otto **Nurmi**
University of Helsinki
Dept. of Computer Science
Teollisuuskatu 23
00510 Helsinki
Finland
tel.: +358-0-708 4250
onurmi@cs.helsinki.fi

Thomas **Ottmann**
Universität Freiburg
Institut für Informatik
Rheinstraße 10-12
W-7800 Freiburg
tel.: +49-761-203 3890
ottmann@informatik.uni-freiburg.de

Linda **Pagli**
Universita di Pisa
Dipartimento di Informatica
Corso Italia 40
I-56125 Pisa
Italy

Ola **Petersson**
Lund University
Dept. of Computer Science
Box 118
22100 Lund
Sweden
ola@dna.lth.se

Mireille **Regnier**
INRIA
Domaine de Voluceau
Rocquencourt
B.P. 105
F-78153 Le Chesnay Cedex
France
tel.: +33-1-39 63 54 78
regnier@inria.inria.fr

Thomas **Roos**
Universität Würzburg
Lehrstuhl für Informatik I
Am Hubland
W-8700 Würzburg
tel.: +49-931-888 5056
roos@informatik.uni-wuerzburg.dbp.de

Peter **Schorn**
ETH Zürich
Institut für Theoretische Informatik
ETH-Zentrum
8092 Zürich
Switzerland
tel.: +41-1-254 7385
schorn@inf.ethz.ch

Sven **Schuierer**
Universität Freiburg
Institut für Informatik
Rheinstraße 10-12
W-7800 Freiburg
tel.: +49-761-203 3894
schuierer@informatik.uni-freiburg.de

Hans-Werner **Six**
Fernuniversität-GH-Hagen
Praktische Informatik III
Feithstraße 140 Postfach 940
W-5800 Hagen
tel.: +49-2331-804 2964
six@fernuni-hagen.de

Esko **Ukkonen**
University of Helsinki
Dept. of Computer Science
Teollisuuskatu 23
00510 Helsinki
Finland
tel.: +358-0-7084172
ukkonen@cs.Helsinki.FI

Ingo **Wegener**
Universität Dortmund
Lehrstuhl für Theoretische Informatik II
Postfach 500 500
W-4600 Dortmund 50
tel.: +49-231-755 2777
wegener@cantor.informatik.uni-dortmund.de

Peter **Widmayer**
Universität Freiburg
Institut für Informatik
Rheinstraße 10-12
W-7800 Freiburg
tel.: +49-761-203 3885 (3886)
widmayer@informatik.uni-freiburg.de

Derick **Wood**
University of Waterloo
Department of Computer Science
Waterloo Ontario N2L 3GI
Canada

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)

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 17, 1.-5.7.1991 (9127)

A. Karshmer, J. Nehmer (editors):
Operating Systems of the 1990s, Dagstuhl-Seminar-Report; 18, 8.-12.7.1991 (9128)

H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization, Dagstuhl-Seminar-Report; 19, 26.8.-30.8.91 (9135)

T. Lengauer, R. Möhring, B. Preas (editors):
Theory and Practice of Physical Design of VLSI Systems, Dagstuhl-Seminar-Report; 20, 2.9.-6.9.91 (9136)

F. Bancilhon, P. Lockemann, D. Tsichritzis (editors):
Directions of Future Database Research, Dagstuhl-Seminar-Report; 21, 9.9.-13.9.91 (9137)

H. Alt , B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 22, 07.10.-11.10.91 (9141)

F.J. Brandenburg , J. Berstel, D. Wotschke (editors):
Trends and Applications in Formal Language Theory, Dagstuhl-Seminar-Report; 23,14.10.-18.10.91 (9142)

H. Comon , H. Ganzinger, C. Kirchner, H. Kirchner, J.-L. Lassez , G. Smolka (editors):
Theorem Proving and Logic Programming with Constraints, Dagstuhl-Seminar-Report; 24, 21.10.-25.10.91 (9143)

H. Noltemeier, T. Ottmann, D. Wood (editors):
Data Structures, Dagstuhl-Seminar-Report; 25, 4.11.-8.11.91 (9145)

A. Borodin, A. Dress, M. Karpinski (editors):
Efficient Interpolation Algorithms, Dagstuhl-Seminar-Report; 26, 2.-6.12.91 (9149)

B. Buchberger, J. Davenport, F. Schwarz (editors):
Algorithms of Computeralgebra, Dagstuhl-Seminar-Report; 27, 16..-20.12.91 (9151)