Richard Cole, Ernst W. Mayr,
Friedhelm Meyer auf der Heide (editors):

Parallel and Distributed Algorithms

Dagstuhl-Seminar-Report; 33
2.-6.3.92 (9210)

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.

Verantwortlich für das Programm:
Prof. Dr.-Ing. José Encarnaçao,
Prof. Dr. Winfried Görke,
Prof. Dr. Theo Härder,
Dr. Michael Laska,
Prof. Dr. Thomas Lengauer,
Prof. Ph. D. Walter Tichy,
Prof. Dr. Reinhard Wilhelm (wissenschaftlicher Direktor).

Gesellschafter: Universität des Saarlandes,
Universität Kaiserslautern,
Universität Karlsruhe,
Gesellschaft für Informatik e.V., Bonn

Träger: Die Bundesländer Saarland und Rheinland Pfalz.

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

INTERNATIONALES BEGEGNUNGS- UND FORSCHUNGSZENTRUM FÜR INFORMATIK

Schloß Dagstuhl

Seminar Report 9210

Parallel and Distributed Algorithms

March 2 – 6, 1992

OVERVIEW

The second Dagstuhl Seminar on *Parallel and Distributed Algorithms* was organized by Richard Cole (Courant Institute), Ernst W. Mayr (Johann-Wolfgang-Goethe-Universität Frankfurt), and Friedhelm Meyer auf der Heide (Universität Paderborn). This year, it brought together 30 participants from 7 countries, 10 of them came from overseas.

The 26 talks presented covered a wide range of topics including parallel data management and rearrangement on networks and PRAMs, both deterministic and randomized, computational geometry, parallel complexity theory, and synchronous and asynchronous computation.

The log-star revolution brought about efficient randomized PRAM algorithms e.g. for maintaining dictionaries and padded sorting. Also an overview of the ideas and techniques of such algorithms was given. Two talks provided important tools for the analysis of randomized and probabilistic algorithms.

Algorithms for integer sorting, sorting on two- and high-dimensional meshes, verification of optimal sorting networks and bounds of the performance of bus-networks demonstrate the never ending youth of parallel sorting.

Interesting algorithms for the construction of trapezoidal diagrams and reporting of intersection points of curves were presented as well as an algorithm that shows the surprising fact that merging of bit-strings on EREW PRAMs is possible in loglog-time.

Dissemination of data in butterfly and de Bruijn networks, efficient on-line routing of complicated permutations, and embedding of trees in the presence of faults were discussed in the area of hypercubic networks. The talks about networks were rounded off by developing a theory of wormhole routing.

Talks in parallel complexity theory dealt with pointers versus arithmetic in PRAMs, structural considerations of parallel algorithms, lower bound techniques, scheduling problems, time-varying data, and the parallel recognition of context-free languages.

Moreover, a look into asynchronous list traversal problems and into the future of optical computers was given.

Worth mentioning, Larry Rudolph demonstrated in his talk the differences between sequential, distributed, and parallel computing by juggling with apples (and eating one of them).

Caused by the pleasant atmosphere, the participants used the surroundings for lively discussions and recreational hiking.

Finally, we would like to express our thanks to all who contributed to the success of this seminar.

Reported by Rolf Wanka

Participants

Richard Anderson, Seattle
Richard Cole, New York
Martin Dietzfelbinger, Paderborn
Faith E. Fich, Toronto
Torben Hagerup, Saarbrücken
Marek Karpiński, Bonn
Ralf Klasing, Paderborn
Manfred Kunde, Newcastle and München
Mirosław Kutyłowski, Wrocław
Klaus-Jörn Lange, München
Fabrizio Luccio, Pisa
Reinhard Lüling, Paderborn
Yossi Matias, College Park
Ernst W. Mayr, Frankfurt
Kurt Mehlhorn, Saarbrücken
Friedhelm Meyer auf der Heide, Paderborn
Burkhard Monien, Paderborn
Ian Parberry, Denton
Prabhakar Ragde, Waterloo
Rajeev Raman, Saarbrücken
Rüdiger Reischuk, Darmstadt
Peter Rossmanith, München
Christine Rüb, Saarbrücken
Larry Rudolph, Jerusalem
Wojciech Rytter, Warszawa
Alan Siegel, New York
Eli Upfal, San Jose
Rolf Wanka, Paderborn
Ingo Wegener, Dortmund
Ralph Werchner, Frankfurt

Contents

CHRISTINE RÜB
Computing Intersections and Arrangements for Red-Blue Curve Segments

LARRY RUDOLPH
Free-Space Optical Network for Massive Parallel Processing

WOJCIECH RYTTER
Reducing Number of Processors in the Parallel Recognition of Two Subclasses of Cfl's

ALAN SIEGEL
Remarks on Analyzing Probabilistic Algorithms

ELI UPFAL
A Theory of Wormhole Routing in Parallel Computers

ROLF WANKA
Periodic Sorting on Two-Dimensional Meshes

RALPH WERCHNER
Optimal Routing of Parentheses and Dynamic Expression Evaluation on the Hypercube

Abstracts

Primitives for Asynchronous List Traversal

by RICHARD ANDERSON

We consider the problem of a collection of asynchronous processors traversing a list with path compression. The worst case performance depends upon the available primitives. If only atomic assignment is available, then the worst case performance of p processors traversing a list of length n is $\Theta(pn)$. However, if a *Splice* primitive is available, the performance improves to $\Theta(p^{1/2}n)$. This result can be generalized to show that a $Splice_k$ operation which removes k elements from the list leads to $\Theta(p^{1/k}n)$ performance. The lower bound proof relies on the introduction of *Tompa Trees*, which are a family of trees which give close to worst case performance for compression based algorithms.

Randomized Parallel Algorithms for Trapezoidal Diagrams

by RICHARD COLE (joint work with Ken Clarkson and Robert E. Tarjan)

We describe a parallel algorithm for computing the trapezoidal diagram of a set of possibly intersecting set of curves, where each curve is composed of straight line segments. If there are k curves, A intersections, and n segments, the algorithm runs in expected time $O(n \log^* n + A + k \log n)$ work and expected $O(\log n \operatorname{loglog} n \log^* n)$ time. An interesting special case is the simple polygon triangulation problem in which $A = 0$ and $k = 1$.

A Perfect Parallel Dictionary

by MARTIN DIETZFELBINGER (joint work with Holger Bast and Torben Hagerup)

We describe new randomized parallel algorithms for the problems of interval allocation, construction of static dictionaries, and maintenance of dynamic dictionaries. All of our algorithms run optimally in constant time with high probability. Our main result is the construction of what we call a *perfect dictionary*, a scheme that allows p processors implementing a set M in space proportional to $|M|$ to process batches of p *insert, delete,* and *lookup* instructions on M in constant time per batch.

Our best results are obtained for a new variant of the CRCW PRAM model of computation called the OR PRAM. For other variants of the CRCW PRAM we show slightly weaker results, with some resource bounds increased by a factor of $O(\log^{(k)} n)$, where $k \in \mathbb{N}$ is fixed but arbitrarily large.

Pointers versus Arithmetic in PRAMs

by FAITH E. FICH (joint work with P. Dymond, N. Nishimura, R. Ragde, and L. Ruzzo)

The Parallel Pointer Machine is a model of parallel computation with a restricted form of memory access and very restricted arithmetic capabilities. However, it is sufficiently powerful to recognize DSPACE($\log n$) and, hence, NC^1 in logarithmic time. We give a simple step-by-step simulation of an n processor PRAM with the same restricted form of memory access, but ordinary arithmetic capabilities, by a parallel pointer machine using $O(\log \log n)$ time per step. Furthermore, we show that even for strong nonuniform variants, $\Omega(\log \log n)$ time per step is required to perform a step-by-step simulation. Since any language recognized by a PRAM with ordinary arithmetic capabilities and a polynomial number of processors in $O(\log n)$ time is in AC^1, this lower bound gives evidence that $NC^1 \neq AC^1$.

Merging and Sorting Strings in Parallel

by TORBEN HAGERUP (joint work with Ola Petersson)

We show that strings of characters, equipped with the usual lexicographical ordering, can be merged and sorted in parallel as efficiently as integers, although with some loss in speed. Specifically:

- Two sorted lists of strings, containing altogether n characters, can be merged with a time-processor product of $O(n)$ in $O(\log n)$ time on a CRCW PRAM, and in $O((\log n)^2)$ time on an EREW PRAM.
- Suppose that n integers of size polynomial in n can be sorted in time $t(n)$ with a time-processor product of $q(n)$. Then a list of strings, containing altogether n characters drawn from an alphabet of size polynomial in n, can be sorted in time $O(t(n) \log n)$ with a time-processor product of $O(q(n) + n \operatorname{loglog} n)$, where the model of computation is either the CREW PRAM or the CRCW PRAM.

Broadcasting in Butterfly and DeBruijn Networks

by RALF KLASING (joint work with Burkhard Monien, Regine Peine, Elena Stöhr)

Broadcasting is the process of message dissemination in a communication network in which a message originated by one processor is transmitted to all processors of the network. In this talk, we present a new lower bound of $1.7417m$ for broadcasting in the butterfly network of dimension m. This improves the best known lower bound of $1.5621m$. We also describe an algorithm which improves the upper bound from $2m$ to $2m - 1$. This is shown to be optimal for small dimensions m. In addition, the presented lower bound technique is used to derive non-trivial lower bounds for broadcasting in the deBruijn network of dimension m. An upper bound of $1.5m + 1.5$ is well-known for this network. Here, we are able to improve the lower bound from $1.1374m$ to $1.3171m$.

Sorting and Routing on Grids Close to the Bisection Bound

by MANFRED KUNDE

Sorting and routing on r-dimensional $n \times \ldots \times n$ grids of processors is studied. Deterministic algorithms are presented for $h - h$ problems, $h \geq 1$, where each processor initially and finally contains h elements. We show that the classical $1 - 1$ sorting can be solved with $(2r - 1.5)n + o(n)$ transport steps, i.e. in about $2.5n$ steps for $r = 2$. The general $h - h$ sorting problem, $h \geq 4r - 4$ can be solved within a number of transport steps that asymptotically differs by a factor of at most 3 from the trivial bisection bound. Furthermore we show that the bisection bound is asymptotically tight for sequences of h permutation routing problems, $h = 4cr$, $c \geq 1$, and for the so-called off-line routing.

Fast Merging on the EREW PRAM

by Mirosław Kutyłowski (joint work with Torben Hagerup)

We investigate the complexity of merging sequences of small integers on the EREW PRAM. Our most surprising result is that two sorted sequences of n bits each can be merged in $O(\log\log n)$ time. More generally, we describe an algorithm to merge two sorted sequences of n integers drawn from the range $0\,..\,m-1$ in $O(\log\log n + \log m)$ time using an optimal number of processors. No sublogarithmic merging algorithm for this model of computation was previously known. The algorithm not only produces the merged sequence, but also computes the rank of each input element in the merged sequence.

On the other hand, we show a lower bound of $\Omega(\log(\min\{n,m\}))$ on the time needed to merge two sorted sequences of length n with elements in the range $0\,..\,m-1$, implying that our merging algorithm is as fast as possible for $m = (\log n)^{\Omega(1)}$. If, in addition, we impose a stability condition requiring the ranks of each input sequence to form an increasing sequence, then the complexity of the problem is $\Theta(\log n)$, even for $m = 2$. Stable merging is thus harder than nonstable merging.

Structural Classification of Parallel Algorithms

by Klaus-Jörn Lange

An *oblivious* PRAM is a CRCW PRAM fulfilling the following restrictions:

1. Each local memory is divided into a data part and an address part.
2. Only address registers may be used for indirect addressing.
3. Only address registers may be used to change the flow of control in conditional statements.
4. No data register may be loaded into an address register. Here, the global memory is regarded as data. There is an atomic operation loading the length of the input into an address register. There is a *conditional assignment statement*

   $$\text{IF } L_a > 0 \text{ THEN } L_b := L_c\,;$$

   affecting data registers, only.

Obviously, oblivious PRAM algorithms have both an input-independent control structure and an input-independent communication structure. We show for a certain normal-form of PRAMs, called *simple* PRAMs, that

$$\text{SIMPLE CRCW TIME}(\log^k n) = AC^k,$$

while

$$\text{SIMPLE OBLIVIOUS CRCW TIME}(\log^k n) = NC^k.$$

If we add to simple PRAMs the ability to use data registers as indices, but keep the control structure strictly input-independent, we get PRAM-representations of classes like DSPACE$(\log n)$ and LOGDCFL.

Computing with Time-Varying Data (The p-Shovelers Problem)

by Fabrizio Luccio (joint work with Linda Pagli)

We consider problems whose size N varies with time, and discuss the organization and analysis of sequential and parallel algorithms for them. We define sequential *d-algorithms* that have the same complexity as if all data were available at time $t = 0$, and *parallel-optimal* d-algorithms. In particular, we study polynomial d-algorithms with data size increasing as $N = n + knt^{\beta}$, where n is a given initial size.

In a first paradigm of computation, we are aimed to attain consistency on all the data currently arrived, before the algorithm terminates. In a second paradigm, we divide the time in consecutive slots, each consisting of an *update phase* to treat new data, and a *free phase* to run different routines. We show that upper and lower bounds on parallel speed-up must be reformulated, and that the *free-time gain* of parallel versus sequential computation may be made arbitrarily large. We give several examples.

Finally, we discuss some variations of the model.

Parallel Randomized Algorithmics

by Yossi Matias (joint work with Joseph Gil and Uzi Vishkin)

A well known lower bound due to Beame and Hastad shows that using a polynomial number of processors on a Parallel Random Access Machine (PRAM), the *parity* – an extremely simple problem – requires nearly-logarithmic time. In contrast, we see in the last two years a considerable number of new results, suggesting paradigms that lead to extremely fast (nearly-constant time), processor-efficient, and space-efficient parallel randomized algorithms. In particular, efficient algorithms with nearly-constant time were given for fundamental problems such as hashing, dictionary, approximate compaction, approximate sum, and load balancing. These algorithms can be used as effective tools in other parallel algorithms. This establishes a new sub-area in parallel algorithmics.

In this talk, I will highlight some of the new results in the theory of nearly-constant time randomized parallel algorithms, and I will illustrate some of the ideas and techniques that are used. From a wider perspective, the evolving theory has the following two informal corollaries regarding the design of parallel algorithms, in general:

- *Space indifference:* Designers of parallel algorithms need not bother themselves with designing memory space efficient algorithms, since there is an automatic way for reducing the space.
- *Processor allocation indifference:* Under rather general circumstances, it is sufficient to characterize an algorithm by only specifying the operations to be performed and without allocation of these operations to processors; this loosely defined style is a means to avoid spending many human-hours on an issue that can be solved automatically.

Embedding Complete Binary Trees in Faulty Hypercubes

by ERNST W. MAYR (joint work with Bob Cypher and Alex Wang)

We consider load one, dilation one, "optimal" expansion embeddings of complete binary trees with $n-1$ levels (and thus $2^{n-1}-1$ vertices) into n-dimensional binary hypercubes (with 2^n nodes), which contain faulty nodes. Based on Wu's bottom-up embedding technique, we present a new technique proceeding top-down. We show how to embed an $n-1$-tree into an n-cube with $O(\frac{n^2}{\log n})$ faulty nodes. The embedding proceeds in stages, based on levels in the tree. The existence of valid embeddings for the different stages is established using different techniques like root counting and an application of Turán's theorem. We also consider a special class of embeddings called *recursive* embeddings. They are characterized by the condition that the two subtrees of an internal node of the binary tree are mapped to disjoint subcubes. We show that, in the worst case, recursive embeddings of $n-1$-trees into n-cubes cannot tolerate more than $2n-3$ faults.

A Method for Obtaining Randomized Algorithms with Small Tail Probabilities

by KURT MEHLHORN (joint work with H. Alt, L. Guibas, R. Karp, A. Wigderson)

Let $X_1, X_2, \ldots$ be independent nonnegative random variables with the common distribution function f. We assume that $E[X_i] = \int_0^\infty x f(x)dx = 1$. $Y_1, Y_2, \ldots$ is another sequence of random variables. A strategy $\mathcal{S}$ is a distribution for $Y_1, Y_2, \ldots$. Let i_0 be the least i such that $X_i \leq Y_i$ and let $T = T_{\mathcal{S},f} = Y_1 + \ldots + Y_{i_0-1} + X_i$. The quality of $\mathcal{S}$ is given by

$$b_{\mathcal{S}}(t) = \sup\{\text{prob}(T_{\mathcal{S},f} \geq t) \mid \int_0^\infty x f(x)dx = 1\}.$$

Theorem 1: $\forall \mathcal{S}\ \forall t : b_{\mathcal{S}}(t) \geq e^{-t}$
Theorem 2: $\forall \mathcal{S}\ \forall t : b_{\mathcal{S}}(t) \leq e^{-t+1}$

A strategy $\mathcal{S}$ is deterministic if each Y_i can assume only a single value.

Theorem 3: $\exists$ det. $\mathcal{S}\ \forall t : b_{\mathcal{S}}(t) \leq e^{-t+O(\sqrt{t}\log t)}$

On the Performance of Networks with Multiple Busses

by FRIEDHELM MEYER AUF DER HEIDE (joint work with Hieu Thien Pham)

We deal with the following questions:

- To which extend can computations of parallel computers be sped up by the use of busses (or shared memory cells)?
- To which extend can shared memory cells of PRAMs be replaced by links?

We characterize the complexity of computing the MAXIMUM on a network with busses as a function in the "broadcast capability" of the network, and give (often matching) upper and lower bounds.

Further we consider very simple networks (planar networks) with very simple (planar) collections of local busses. We show that MAXIMUM can be computed in time $O(\log n)$ on a $\sqrt{n} \times \sqrt{n}$-grid, if it is enhanced by a very simple, regular, planar system of busses of length 4.

On the other hand, SORTING needs time $\Omega(n)$ on planar networks, even if they are enhanced by an arbitrary planar bus system.

The Computational Complexity of Optimal Sorting Network Verification

by IAN PARBERRY

A *sorting network* is a combinational circuit for sorting, constructed from comparison-swap units. The depth of such a circuit is a measure of its running time. It is reasonable to hypothesize that only the fastest (that is, the shallowest) sorting networks are likely to be fabricated. It is shown that the verification of shallow sorting networks is computationally intractable. More precisely, it is shown that the problem of verifying sorting networks with depth very close to the optimum is Co-NP complete, and further that any deterministic or randomized algorithm that has access only to the inputs and outputs of the sorting network must take exponential time to verify it. Despite the computational intractability of the shallow sorting network verification problem, we describe a construction and verification algorithm which was recently executed on a supercomputer to demonstrate that there is no 9-input sorting network of depth 6, a question which had remained open for over 15 years.

Towards Lower Bounds for Parallel Computation over Moderate-Sized Domains

by PRABHAKAR RAGDE

Many lower bounds on parallel models of computation use powerful results in Ramsey theory to control the behavior of programs to the point where adversary arguments can be applied. This means that the lower bounds only hold when the initial inputs — say, n integers, for a problem to be solved by n processors — are drawn from a domain whose size is an extremely rapidly growing function of n. Recent work has shown that these lower bounds may not hold for moderate-sized domains: Berkman, JáJá, Krishnamurthy, Thurimella, and Vishkin have demonstrated algorithms that work on domain $\{1, 2, \ldots, s\}$ and can merge two sorted sequences of length n in $O(\log\log\log s)$ steps on the CREW PRAM and can find the prefix maxima of n numbers in the same time bound on the PRIORITY CRCW PRAM.

In this talk, I showed that for $s(n) = 2^{n^{\Theta(\log\log n)}}$, any PRIORITY algorithm that computes the maximum of n numbers from the domain $\{1, 2, \ldots, s\}$ requires time $\Omega(\log\log\log s)$ time. The proof replaces powerful Ramsey theorems with simple ones such as the sunflower theorem of Erdős and Rado. The lower bound cannot be shown for all functions $s(n)$ because, for instance, the maximum of numbers in the range $\{1, 2, \ldots, n^{O(1)}\}$ can be computed in $O(1)$ time, even on the COMMON CRCW PRAM. More work is needed to determine the exact tradeoff between domain size and time for this and other problems parameterized in this fashion.

Fast Optimal Comparison-Based Sorting, with Applications

by RAJEEV RAMAN (joint work with Torben Hagerup)

We consider the problem of sorting on the CRCW PRAM model. If we require the sorted output to be in consecutive locations of shared memory, sorting n elements must take $\Omega(\log n/\log\log n)$ time with any polynomial number of processors (Beame and Håstad, (1989)).

This lower bound does not apply if the algorithm is permitted to place extra copies of some of the input elements in the output. We therefore consider the problem of *padded sorting*, which allows the n input elements to be placed in an array of size $(1 + o(1))n$ **in** sorted order, with blanks filled in appropriately. A padded output is quite often just as good as the usual compact output, for example, searching is just as easy.

We give a randomized comparison-based algorithm that padded sorts n elements in

Precise Time Bounds for Computing Boolean Functions on CREW PRAMs

by Peter Rossmanith

This work deals with the precise number of steps that are necessary and sufficient to compute a given boolean function on a CREW PRAM. A new technique to derive lower bounds for this model is presented. The technique is based on a recursive definition of sets of polynomials that represent boolean functions. The t-th set contains only polynomials that represent boolean functions that can be computed in t steps or less. A slightly more complicated definition even leads to sets that exactly contain those functions.

The new method is used to show some counter-intuitive results. While a CREW PRAM takes exactly $\phi(n)$ steps to compute the OR of n variables, it takes $\phi(n+1)$ steps to computes the NOR. ($\phi(n) = \min\{\, t \mid F_{2t+1} \leq n \,\}$, where F_i is the i-th Fibonacci number.) That means that for infinitely many n it takes one step more to compute the NOR than to compute the OR. Similarly, it takes $\phi(n)$ steps to compute the OR of the second to the n-th input variable, i.e., it does not help to ignore the first variable in this case. The reason for this behavior is that the first memory cell contains the first input variable at the beginning of the computation, as well as the result afterwards.

Computing Intersections and Arrangements for Red-Blue Curve Segments

by Christine Rüb

Let A and B be two sets of "well-behaved" (i.e., continuous and x-monotone) curve segments in the plane, where no two segments in A (similarly, B) intersect. In this talk, we show how to report all points of intersection between segments in A and segments in B, and how to construct the arrangement defined by the segments in $A \cup B$ in parallel using the concurrent-read-exclusive-write (CREW) PRAM model. The algorithms perform a work of $O(n \log n + k)$ using $p \leq n + k/\log n$ ($p \leq n/\log n + k/\log^2 n$, resp.,) processors if we assume that the handling of segments is "cheap", e.g., if two segments intersect at most a constant number of times, where n is the total number of segments and k is the number of points of intersection. If we only assume that a single processor can compute an arbitrary point of intersection between two segments in constant time, the performed work increases to $O(n \log n + m(k+p))$, where m is the maximal number of points of intersection between two segments. We also show how to count the number of points of intersection between segments in A and segments in B in time $O(\log n)$ using n processors on a CREW PRAM if two curve segments intersect at most twice.

Free-Space Optical Network for Massive Parallel Processing

by Larry Rudolph

We describe an interconnection network based on lasers, detectors, and hologram, which is capable of interconnecting more than 100,000 processors. The processors themselves are traditional but they communicate via free space optics. Associated with each processor is an electronically controlled switch that can direct the optical laser beam towards any one of n different target facets. Once the beam hits a facet it is routed to the destination. We first describe the fundamental limits on the number of processors according to geometry, diffraction, and power considerations. Next, we describe the architecture of the network. Routing issues are discussed.

Possible abstract models of computations, derived from such an architecture, are discussed.

Reducing Number of Processors in the Parallel Recognition of Two Subclasses of Cfl's

by WOJCIECH RYTTER

It is shown that the hardest part in the recognition of cfl's (context-free languages) is a kind of reachability problem for a huge graph. Generally this is solved via transitive closure, then it takes n^6 processors. For unambiguous and deterministic languages the corresponding graph has a special property of the path uniqueness. Instead of transitive closure the tree contraction can be applied. This economizes the number of processors in $(\log n)^2$ time computation to n^3 (for unambiguous) and n^2 (for deterministic cfl's). This result was done together with M. Chytil, M. Crochemore and B. Monien. Parallel $\log n$ time computations are much harder. The number of processors for the recognition of deterministic cfl's is reduced by almost a linear factor. This result was done together with B. Monien and H. Schaepers. The related (maybe even more important) result is the simplification of the Reif algorithm for the parallel simulation of deterministic pda's.

Remarks on Analyzing Probabilistic Algorithms

by ALAN SIEGEL

We illustrate the opportunities offered by resampling, Chernoff bounds and other probabilistic inequalities to simplify the analysis of probabilistic algorithms. Probability estimates for complicated configurations in combinatorial selection without replacement are shown to be readily attained from Chernoff bounds for Bernoulli trials, and a new Chernoff bound for Bernoulli trials is presented, which is tight as a function of the mean and variance. In addition, a variety of probabilistic inequalities — some longstanding but seemingly unknown to the Computer Science Community — are introduced and shown to have potential for further use.

A Theory of Wormhole Routing in Parallel Computers

by ELI UPFAL (joint work with Prabhakar Raghavan)

Virtually all theoretical work on message routing in parallel computers has dwelt on *packet routing*: messages are conveyed as packets, an entire packet can reside at a node of the network, an a packet is sent from the queue of one node to the queue of another node until it reaches its destination. The current trend in multicomputer architecture, however, is to use *wormhole routing*. In wormhole routing, a message is transmitted as a continuous stream of bits, physically occupying a sequence of nodes/edges in the network. Thus, a message resembles a worm burrowing through the network. We give theoretical analyses of simple wormhole routing algorithms, showing them to be nearly optimal for butterfly and mesh connected networks.

Periodic Sorting on Two-Dimensional Meshes

by Rolf Wanka (joint work with Mirosław Kutyłowski)

We consider the following periodic sorting procedure on two-dimensional meshes of processors: Initially, each node contains one number. We proceed in rounds, where each round consists of sorting the columns of the grid, and, during the second phase, of sorting the rows according to the snake-like ordering. We characterize exactly the number of rounds necessary to sort on an $l \times m$-grid in the worst case, where l is the number of the rows and m is the number of the columns. An upper bound of $\lceil \log l \rceil + 1$ was known. This bound is tight for the case that m is not a power of 2. Surprisingly, it turns out that much less rounds are necessary, if m is a power of 2 (and $m \ll l$): In this case, exactly $\min\{\log m + 1, \lceil \log l \rceil + 1\}$ rounds are needed in the worst case.

Optimal Routing of Parentheses and Dynamic Expression Evaluation on the Hypercube

by Ralph Werchner (joint work with Ernst Mayr)

We consider a new class of routing requests or partial permutations for which we give optimal on-line routing algorithms on the hypercube and shuffle-exchange network. For well-formed words of parentheses, our algorithm establishes communication between all matching pairs in logarithmic time. This routing problem can be reduced to a somewhat simpler problem, to route all matching pairs into common meeting points. The latter problem is solved using a divide(-and-conquer) approach. Our routing algorithm can be applied to the membership problem for Dyck languages and a number of syntactic problems for algebraic expressions. Furthermore, essentially the same approach leads to an algorithm evaluating expressions of linear size in logarithmic time on a hypercube.

E-Mail Addresses

Richard Anderson:	anderson@cs.washington.edu
Richard Cole:	cole@cs.nyu.edu
Martin Dietzfelbinger:	martind@uni-paderborn.de
Faith Fich:	fich@cs.toronto.edu
Torben Hagerup:	torben@mpi-sb.mpg.de
Marek Karpiński	karpinski@cs.uni-bonn.de
Ralf Klasing:	atze@uni-paderborn.de
Manfred Kunde:	kunde@informatik.tu-muenchen.de
Mirosław Kutyłowski:	mirekk@plwruw11.bitnet
Klaus-Jörn Lange:	lange@informatik.tu-muenchen.de
Fabrizio Luccio:	luccio@di.unipi.it
Reinhard Lüling:	rl@uni-paderborn.de
Yossi Matias:	matias@umiacs.umd.edu
Ernst Mayr:	mayr@thi.informatik.uni-frankfurt.de
Kurt Mehlhorn:	mehlhorn@mpii01004.cs.uni-sb.de
Friedhelm Meyer auf der Heide:	fmadh@uni-paderborn.de
Burkhard Monien:	bm@uni-paderborn.de
Ian Parberry:	ian@ponder.csci.unt.edu
Prabhakar Ragde:	plragade@maytag.waterloo.edu
Rajeev Raman:	raman@mpi-sb.mpg.de
Rüdiger Reischuk:	reischuk@iti.informatik.th-darmstadt.de
Peter Rossmanith:	rossmani@informatik.tu-muenchen.de
Christine Rüb:	rueb@mpi-sb.mpg.de
Larry Rudolph:	rudolph@cs.huji.ac.il
Wojciech Rytter:	rytter@plearn.bitnet
Alan Siegel:	siegel@cs.nyu.edu
Eli Upfal:	ely@almaden.ibm.com
Rolf Wanka:	wanka@uni-paderborn.de
Ingo Wegener:	wegener@cantor.informatik.uni-dortmund.de
Ralph Werchner:	werchner@thi.informatik.uni-frankfurt.de

Addresses

Richard Anderson
University of Washington
Department of Computer Science
and Engeneering / FR-35
Seattle WA 98195
U.S.A.
☎ +1-206-543-4305

Martin Dietzfelbinger
Universität-GH Paderborn
FB 17 – Mathematik/Informatik
Heinz-Nixdorf-Institut
Warburger Str. 100
D-W-4790 Paderborn
Germany
☎ +49-5251-60-3308

Torben Hagerup
Max-Planck-Institut für Informatik
Im Stadtwald 15
D-W-6600 Saarbrücken 11
Germany
☎ +49-681-302-5358

Ralf Klasing
Universität-GH Paderborn
FB 17 – Mathematik/Informatik
Warburger Str. 100
D-W-4790 Paderborn
Germany
☎ +49-5251-60-2660

Mirosław Kutyłowski
Universytet Wrocławski
Instytut Informatyki
Przesmyckiego 20
PL-51-151 Wrocław
Poland

Richard Cole
New York University
Courant Institute of
Mathematical Sciences
251 Mercer Street
New York NY 10012
U.S.A.
☎ +1-212-998-3119

Faith Fich
University of Toronto
Dept. of Computer Science
10 King's College of Road
Toronto Ontario M5S 1A4
Canada
☎ +1-416-978-6183

Marek Karpiński
Universität Bonn
Inst. für Informatik
Römerstr. 164
D-W-5300 Bonn
Germany
☎ +49-228-550-224

Manfred Kunde
TU München
Institut für Informatik
Arcisstraße 21
D-W-8000 München 2
Germany

Klaus-Jörn Lange
TU München
Institut für Informatik
Arcisstraße 21
D-W-8000 München 2
Germany
☎ +49-89-2105-2403

Fabrizio Luccio
Università di Pisa
Dipartimento di Informatica
Corso Italia 40
I-56125 Pisa
Italy
☎ +39-50-510-220 / 235

Reinhard Lüling
Universität-GH Paderborn
FB 17 – Mathematik/Informatik
Warburger Str. 100
D-W-4790 Paderborn
Germany
☎ +49-5251-60-3331

Yossi Matias
Univ. of Maryland at College Park
UMIACS
College Park MD 20742
U.S.A.
☎ +1-301-405-6715

Ernst W. Mayr
Universität Frankfurt
Fachbereich Informatik (20)
Robert-Mayer-Str. 11-15
D-W-6000 Frankfurt 11
Germany
☎ +49-69-798-8326

Kurt Mehlhorn
Max-Planck-Institut für Informatik
Im Stadtwald 15
D-W-6600 Saarbrücken 11
Germany
☎ +49-681-302-5351

Friedhelm Meyer auf der Heide
Universität-GH Paderborn
FB 17 – Mathematik/Informatik
Heinz-Nixdorf-Institut
Warburger Str. 100
D-W-4790 Paderborn
Germany
☎ +49-5251-60-3310 / -3306

Burkhard Monien
Universität-GH Paderborn
FB 17 – Mathematik/Informatik
Warburger Str. 100
D-W-4790 Paderborn
Germany
☎ +49-5251-60-26-50

Ian Parberry
University of North Texas
Dept. of Computer Science
P.O. Box 13886
Denton TX 76203-3886
U.S.A.
☎ +1-817-565-2845

Prabhakar Ragde
University of Waterloo
Department of Computer Science
Waterloo Ontario N2L 3GI
Canada
☎ +1-519-888-4660

Rajeev Raman
Max-Planck-Institut für Informatik
Im Stadtwald 15
D-W-6600 Saarbrücken 11
Germany
☎ +49-681-302 5421

Rüdiger Reischuk
TH Darmstadt
Institut für Theoretische Informatik
Alexanderstr. 10
D-W-6100 Darmstadt
Germany
☎ +49-6151-16-3416

Christine Rüb
Max-Planck-Institut für Informatik
Im Stadtwald 15
D-W-6600 Saarbrücken 11
Germany
☎ +49-681-302 5422

Peter Rossmanith
TU München
Institut für Informatik
Arcisstraße 21
D-W-8000 München 2
Germany
☎ +49-89-2105-2397

Larry Rudolph
The Hebrew University of Jerusalem
Department of Computer Science
91904 Jerusalem
Israel
☎ +972-2-585261

Wojciech Rytter
University of Warsaw
Dept. of Mathematics & Informatics
Ul. Banacha 2
00-913 Warsaw 59
Poland

Alan Siegel
New York University
Courant Institute of
Mathematical Sciences
251 Mercer Street
New York NY 10012
U.S.A.
☎ +1-212-998-3122

Eli Upfal
IBM Almaden Research Center
K 53/802
650 Harry Road
San Jose CA 95120-6099
U.S.A.
☎ +1-408-927-1788

Rolf Wanka
Universität-GH Paderborn
FB 17 – Mathematik/Informatik
Heinz-Nixdorf-Institut
Warburger Str. 100
D-W-4790 Paderborn
Germany
☎ +49-5251-60-3309

Ingo Wegener
Universität Dortmund
Fachbereich Informatik LS II
Postfach 500 500
D-W-4600 Dortmund 50
Germany
☎ +49-231-755-2776

Ralph Werchner
Universität Frankfurt
Fachbereich Informatik (20)
Robert-Mayer-Str. 11-15
D-W-6000 Frankfurt 11
Germany
☎ +49-69-798-8324

Zuletzt erschienene und geplante Titel:

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.-12.9.91 (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. Dress, M. Karpinski, M. Singer(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)

K. Compton, J.E. Pin , W. Thomas (editors):
Automata Theory: Infinite Computations, Dagstuhl-Seminar-Report; 28, 6.-10.1.92 (9202)

H. Langmaack, E. Neuhold, M. Paul (editors):
Software Construction - Foundation and Application, Dagstuhl-Seminar-Report; 29, 13..-17.1.92 (9203)

K. Ambos-Spies, S. Homer, U. Schöning (editors):
Structure and Complexity Theory, Dagstuhl-Seminar-Report; 30, 3.-7.02.92 (9206)

B. Booß, W. Coy, J.-M. Pflüger (editors):
Limits of Modelling with Programmed Machines, Dagstuhl-Seminar-Report; 31, 10.-14.2.92 (9207)

K. Compton, J.E. Pin , W. Thomas (editors):
Automata Theory: Infinite Computations, Dagstuhl-Seminar-Report; 28, 6.-10.1.92 (9202)

H. Langmaack, E. Neuhold, M. Paul (editors):
Software Construction - Foundation and Application, Dagstuhl-Seminar-Report; 29, 13..-17.1.92 (9203)

K. Ambos-Spies, S. Homer, U. Schöning (editors):
Structure and Complexity Theory, Dagstuhl-Seminar-Report; 30, 3.-7.02.92 (9206)

B. Booß, W. Coy, J.-M. Pflüger (editors):
Limits of Modelling with Programmed Machines, Dagstuhl-Seminar-Report; 31, 10.-14.2.92 (9207)

N. Habermann, W.F. Tichy (editors):
Future Directions in Software Engineering, Dagstuhl-Seminar-Report; 32; 17.02.-21.02.92 (9208)

R. Cole, E.W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms; Dagstuhl-Seminar-Report; 33; 02.03.-06.03.92 (9210)

P. Klint, T. Reps (Madison, Wisconsin), G. Snelting (editors):
Programming Environments; Dagstuhl-Seminar-Report; 34; 09.03.-13.03.92 (9211)

H.-D. Ehrich, J.A. Goguen, A. Sernadas (editors):
Foundations of Information Systems Specification and Design; Dagstuhl-Seminar-Report; 35; 16.03.-19.03.9 (9212)

W. Damm, Ch. Hankin, J. Hughes (editors):
Functional Languages:
Compiler Technology and Parallelism; Dagstuhl-Seminar-Report; 36; 23.03.-27.03.92 (9213)

Th. Beth, W. Diffie, G.J. Simmons (editors):
System Security; Dagstuhl-Seminar-Report; 37; 30.03.-03.04.92 (9214)

C.A. Ellis, M. Jarke (editors):
Distributed Cooperation in Integrated Information Systems; Dagstuhl-Seminar-Report; 38; 06.04.-08.04.92 (9215)