BRICS *Newsletter*

Basic Research in Computer Science No 6, April 1997

In this Issue

Welcome

Welcome to the sixth issue of the BRICS newsletter. Its purpose is to inform you of appointments, publications, courses and other activities within BRICS. Further details can be obtained by contacting the addresses on the back page.

We are later than we intended in issuing this BRICS newsletter. Christmas and the early new year were occupied both with LICS'97 (Glynn Winskel is program chair this year) and with producing a report on the first three years of BRICS; the report is to serve both as a progress summary and an application to the Danish National Research Foundation for BRICS to be funded for another five years after '98.

This year marks the start of an important venture, the BRICS International PhD School, under the direction of Mogens Nielsen. The school, like BRICS funded by the Danish National Research Foundation, is an experiment in Danish postgraduate education, the idea being to supplement the twenty or so Danish PhD students of BRICS with a corresponding number from abroad. Yearly there are around five studentships for students from abroad. While the students' core education will be based on the BRICS areas of semantics, logic and algorithmics, the PhD School is also concerned with providing the back-up for students interested in applying their theoretical knowledge to more applied areas.

Coming Events

For details see the BRICS Activities web page:

www.brics.dk / Activities.

Algebraic Theory of Automata, Temporal Logic and Expressiveness

April 29 and 30, 1997, *Denis Thérien*, School of Computer Science, McGill University, Canada, will give two double lectures covering the following topics: algebraic theory of automata, decomposition of finite semigroups, logical description of $*$-free languages and regular languages and boolean circuits.

Information Theoretic Security in Cryptography

May 1–2, 1997, *Ueli Maurer*, Department of Computer Science, ETH (The Swiss Federal Institute of Technology), Zürich, Switzerland, will give a mini-course on Information Theoretic Security in Cryptography.

Cryptography is sometimes perceived as being inherently based on unproven computational assumptions. While this has never been really true, the last few years of research have demonstrated in a rather dramatic way that many types of data security can be realized—also in practice—based on no assumptions whatsoever.

One example of this is Quantum Cryptography, where security is based on fundamental laws of quantum physics. Both the physical realization and the theoretic background has strong links to the related field of Quantum Computing. Another example is noisy communication. While noise is usually dealt with as a problem that should be removed, it can also be turned into an advantage allowing secure communication, assuming only that enemies cannot totally remove noise from the data they receive.

Information theory and related topics, such as universal hashing, are at the heart of such “unconditional” cryptography. The course will introduce the necessary theory and show a number of scenarios where the theory applies.

Combinatorial Optimisation Meeting

May 5–6, 1997, BRICS is organising a meeting on combinatorial optimisation. The aim is to bring together Danish (and non-Danish) combinatorial optimisation people and discuss current state of the art and research directions. Algorithmic aspects are of particular interest.

The featured speaker of the meeting is Martin Grötschel from the Konrad-Zuse-Zentrum für Informationstechnik, Berlin (ZIB). A number of other people have accepted our invitation to talk at the meeting. Further information is available at www.brics.dk / Activities / 97 / CombOpt.

First Order Logic with Dependent Sorts, Simulation, and Higher Dimensional Categories

In the middle of May, *Michael Makkai*, McGill University, Canada, will give a mini-course on first order logic with dependent sorts, simulation, and higher dimensional categories.

Inductive Logic Programming

May 20–23, 1997, *Nada Lavrac*, Department of Intelligent Systems, Jozef Stefan Institute, Ljubljana, Slovenija, and *Peter Flach*, Infolab, Department of Management Information Science, Tilburg University, Holland, will give three double lectures on Inductive Logic Programming.

Inductive logic programming (ILP) is a research area, combining principles of inductive machine learning and logic programming. ILP aims at a formal framework and practical algorithms for inductively learning relational descriptions in the form of logic programs. ILP is of interest to inductive machine learning as it significantly extends the usual attribute-value representation and consequently enlarges the scope of machine learning applications; it is also of interest to logic programming as it extends the basically deductive framework of logic programming towards the use of induction. The course aims at giving an overview of the field, with an emphasis on the foundations and basic techniques.

ILP has already shown its potential for applications. Applications in a machine learning context include: knowledge acquisition, inductive data engineering, scientific discovery and knowledge discovery in databases. Potential applications in logic programming include: knowledge base updating, logic program synthesis, debugging and verification of programs, and deriving integrity constraints from databases.

Combinatorial Pattern Matching ’97

The 8th Annual Symposium on Combinatorial Pattern Matching (CPM ’97) will be held at the Computer Science Department in the University of Aarhus, Århus, Denmark, from Monday, June 30, through Wednesday, July 2, 1997.

Combinatorial Pattern Matching addresses issues of searching and matching strings and more complicated patterns such as trees, regular expressions, graphs, point sets, and arrays. The goal is to derive non-trivial combinatorial properties for such structures and then to exploit these properties in order to achieve improved performance for the corresponding computational problem.

In recent years, a steady flow of high-quality research on this subject has changed a sparse set of isolated results into a full-fledged area of algorithmics with important applications. This area is expected to grow even further due to the increasing demand for speed and efficiency that comes from molecular biology, but also from areas such as information retrieval, pattern recognition, compiling, data compression, program analysis and security. The objective of annual CPM gatherings is to provide an international forum for research in combinatorial pattern matching.

In connection with the symposium on June 28 and June 29, a summer school will be organised at the same location and focus on topics mainly within computational biology.

The event is organised Jotun Hein, Department of Ecology and Genetics, Institute of Biological Sciences, University of Aarhus in cooperation with Christian N. S. Pedersen, BRICS, Aarhus.

Please see www.brics.dk/CPM97 for further information.

Computer Science Logic ’97

The Annual Conference of the European Association for Computer Science Logic (CSL ’97) will be held in Aarhus in the period August 23–29, 1997.

The conference is intended for computer scientists whose research activities involve logic, as well as for logicians working on topics significant for computer science.

Scientific Program

August 23–24: Tutorials on Games and Logic
S. Abramsky (Edinburgh): Game Semantics
E. A. Emerson (Austin): Games, μ-calculus, and program verification
W. Thomas (Kiel), I. Walukiewicz (Warsaw): Determinacy, the Rabin Tree Theorem and its extensions

August 25–29: Invited Lectures and Contributed Papers
Invited speakers: S. Buss (San Diego), H. Comon (Paris), T. Coquand (Gothenburg), M. Hyland (Cambridge), N. Immerman (Amherst), N. Klar-

lund (Murray Hill), Y. Moschovakis (Los Angeles), L. Pacholski (Wroclaw).

Student Grants

A limited number of grants to attend the Conference and the Tutorials are available for students. The grants will cover local expenses and conference fee, but will **not** cover travel expenses.

Further information can be obtained from www.brics.dk/CSL97.

School on Computational and Syntactic Methods

August 11–22, 1997, European Educational Forum (see page 7) organises a *School on Computational and Syntactic Methods* in Mierlo, The Netherlands.

The program of the school covers the following topics: Term Rewriting, Type Systems, Models of Concurrency, Effective Algebras and Process Algebra.

Further information can be obtained by following the corresponding link on the EEF home page: www.tucs.abo.fi/EEF.

School on Natural Computation

August 25–29, 1997, the European Educational Forum organises a *School on Natural Computation*, SNAC, at the Mauno Koivisto Centre, Turku, Finland.

The School on Natural Computation will concentrate on four branches of natural computation, namely artificial neural networks, genetic algorithms, DNA-computing and quantum computing.

The main goal of the school is to establish a deep understanding of the theory and practice of natural computation such that the participants are able to explore the paradigm and to assess its relation to other fields in order to specify better solutions to real world complex problems.

For further information, please consult the associated page: www.abo.fi/˜kaisa/SNAC.html.

Reports on Events

Set Constraints

On August 14 and 15, 1996, *Dexter Kozen*, Joseph Newton Pew, Professor of Computer Science, Cornell University, Ithaca, NY, USA, gave a course of six lectures on Set Constraints. Course description and a list of references can be found on the web.

Competitive Online Algorithms

During August 27–29, 1996, *Susanne Albers*, Max-Planck-Institut für Informatik, Saarbrücken, Germany gave a mini-course on Competitive Online Algorithms.

For the course notes, please see [LS-96-2] in the Lecture Series below. Notes as well as slides are obtainable from www.brics.dk/LS/96/2.

Non-Interleaving Transition Systems

On October 3 and 4, 1996, *Vladimiro Sassone*, Dipartimento di Informatica, Pisa, Italy, gave a mini-course on Non-Interleaving Transition Systems. Course description is on the web.

Explicit Substitution

On October 10 and 11, in connection with the Autumn School on Verification, *Kristoffer Høgsbro*

Rose, BRICS, gave a three lecture mini-course on "Explicit Substitution."

The lecture notes appear as [LS-96-3] in the BRICS Lecture Series below. These notes and selected slides are available from the URL: www.brics.dk/LS/96/3.

Autumn School on Verification

October 28 – November 1, 1996.

We bring here a report written by on of the participants *Robert-C. Riemann*, LRI (Laboratoire de Recherche en Informatique), Université Paris-Sud, Orsay, France, for the EATCS Bulletin (appeared in the February issue).

This years 5 day autumn school of the *Basic Research In Computer Science* (BRICS) institute was held at the University of Aarhus, Denmark. The event centred on verification methods and tools in theorem proving and model checking, and perspectives to combine both methods. 87 participants from 16 countries and 4 continents were invited by BRICS to attend the school. Every participant was supported by a grant which covered tuitions, lodging, and meals.

The lectures and their subjects were as follows. Most of the lectures were ended with a tool demonstration session.

- David Basin (Max-Planck Institut für Informatik, Saarbrücken, Germany): Verification Based on Monadic Logic, (4 hours). The lecture was centred on a new approach to hardware verification based on describing circuits in monadic second-order logic (M2L). David Basin described the theory and practice of how M2L, as embodied in the MONA tool, can be used to automatically verify parameterised circuit designs. The representation of generic designs in this logic and the decision procedure, which reduces formulas to canonical automata, was demonstrated.
- Edmund M. Clarke (Carnegie-Mellon University, Pittsburgh, USA): Symbolic Model Checking, (4 hours). Ed Clarke started with an introduction to CTL, model checking, and binary decision diagrams. The second part of the lecture then introduced us to the specification language SMV (Symbolic Model Checking). Specifications in SMV of a synchronous arbiter, an arbiter cell, and a pipelined ALU served as examples, which were verified in the SMV tool demonstration.
- Thomas A. Henzinger (University of California at Berkeley, USA): Automatic Verification of Real-Time and Hybrid Systems (4 hours). Thomas Henzinger introduced us to the theory of hybrid automaton, a formal model for dynamical systems with discrete and continuous components. He presented several recent results and demonstrated that methods for the verification of finite-state systems can be used to analyse certain systems with uncountable state space.
- Gerard J. Holzmann (Bell Laboratories, Murray Hill, USA): On-the-Fly Model Checking Tutorial, (4 hours). This lecture gave an overview of the design and structure of the SPIN verification tool. Gerard Holzmann reviewed the automata-theoretical foundations, and gave examples of some typical applications of the model checker in practise.
- Kim G. Larsen (BRICS, Aalborg, Denmark): UPPAAL tool demonstration, (1 hour). UPPAAL is a tool for symbolic simulation and automatic verification of real-timed systems modelled as networks of timed automaton. The demonstration showed us how to simulate and verify a small production line.
- Tom F. Melham (University of Glasgow, UK): Some Research Issues in Higher Order Logic Theorem Proving, (4 hours).

Figure 1: The signing of the European Educational Forum contract. The three signatories from left to right: *Ralph-Johan Back* (TUCS), *Mogens Nielsen* (BRICS) and *Jos Baeten* (IPA) supervised by *Grzegorz Rozenberg* (IPA), the chairman of the EEF Steering Committee.

Based on the experience of the HOL research community, Tom Melham talked about requirements of the design of interfaces of verification tools, the embedding of specialised languages into higher order logic supported by a theorem prover, and the integration and implementation of automated proof methods.

- Robert Pollack (University of Edinburgh, UK): What we Learn from Formal Checking, (4 hours). At first Robert Pollack discussed the value of large machine-checked proofs, proposing a view in which belief in machine verifications is based on evidence and asking whether a claimed proof is actually a derivation in the claimed formal system, and whether what it proves is the claimed theorem. He introduced us to type theory, namely the use of inductive definitions to represent data types and relations and the tool LEGO for checking proofs in the Extended Calculus of Constructions.

- Mandayam K. Srivas (Computer Science Laboratory, SRI International, Menlo Park, USA): A Combined Approach to Hardware Verification: Proof-checking, Rewriting with decision procedures and Model-checking, (4 hours). With the goal to illustrate how an integrated verification system that combines proof checking, rewriting, decision procedures, and model checking can be used for effective hardware verification, Mandayam Srivas started with the design philosophy of the Prototype Verification System (PVS). The second part of his lecture was concerned with the combination of rewriting with decision procedures and model checking with proof checking, and the problem of scaling up verification to industrial design.

The series of lectures gave a wide overview of the area of verification based on model checking and theorem proving, for 'newcomers' as well as for participants already familiar with the research area.

The lecture material was published in the BRICS Note Series ([NS-96-2] to [NS-96-12]).

On Wednesday afternoon we were taken to visit the Aarhus museum of science. The school ended with an excellent farewell dinner on Friday evening, held in the university restaurant. According to the tradition at Aarhus University, all participants performed together from time to time during the dinner several songs. The tradition of the last song required us to climb up the dinner tables from verse to verse and so the school was ended with all participants standing on the tables.

The organisation of the school was excellent. In the name of all participants let me thank the organisers: Allan Cheng, Mogens Nielsen, and the secretary: Karen.

European Educational Forum

The European Educational Forum (EEF) is a joint activity of three research schools: BRICS, IPA (Institute for Programming Research and Algorithmics) from the Netherlands, and TUCS (Turku Centre for Computer Science) from Finland.

The goal of EEF is to organise various forms of activities covering a broad spectrum of Basic Research and Applications and aimed at the education of PhD's and young researchers. The form of the activities may vary: it may take form of a workshop, a school, a topic-focused conference or a conference presenting a number of topics that are relevant in Basic Research and Applications at the given moment.

The EEF was entered into contract (figure 1) during the first EEF-event, a School on Embedded Systems held in Veldhoven, The Netherlands, November 25–29, 1996.

For further information please see the EEF home page: www.tucs.abo.fi / EEF.

Kiel Meeting

by Thomas T. Hildebrandt

A cold and wet morning, Tuesday December 3, a number of BRICS people boarded a mini-bus at the parking ground in front of the department, heading towards Kiel. The crowd, a rare mixture of professors, post docs, PhD and grad. students (see picture) had signed up for a one-and-a-half day visit to Kiel.

The scientific content of the trip was "some informal BRICS / Kiel talks" exposing the work of each group for the other. Five of us had volunteered to give a talk on our current line of work, latest results or a theorem one hoped to prove next week.

The drive went as planned (and clearly, the conversation in the bus was a lot more interesting than the landscape). After a short stop at the computer science department in Kiel picking up the Kiel group (see figure 2), we went to a pleasant conference building by the harbour.

Here a good solid lunch awaited us, opening up for an informal chat. Thereafter we delved into the first row of talks, continuing till late in the afternoon, interrupted by coffee breaks and a river-side walk. Later we went to a nice restaurant (after a short walk in the city) and even later to our comfortable hotel where we could rest after a long, but elucidating day.

Wednesday morning we finished the series of talks (10 in all) and headed home in high spirit, only with a short stop by the border buying various kinds of liquid and non-liquid goods.

Figure 2: Participants in the Kiel Meeting. From the left: *Anders Sandholm, P. S. Thiagarajan, Thomas S. Hune, Marcin Jurdziński, Jesper G. Henriksen, Nils Buhrke* (Kiel), *Wolfgang Thomas* (Kiel), *Jens Vöge* (Kiel), *Thomas T. Hildebrandt, Margrit Krause* (Kiel), *Ina Schiering* (Kiel), *Erich Valkema* (Kiel), *Thomas Wilke* (Kiel), *Sebastian Seibert* (Kiel), *Glynn Winskel, Oliver Matz* (Kiel), *Prakash Panangaden, Uffe H. Engberg, Kousha Etessami, Helmut Lescow* (Kiel).

The informal programme for the visit had made it pleasant to listen to the many talks, which were an interesting mixture (some even on work that grew into accepted papers for LICS '97). It gave us a good peep at the field of work done by the group in Kiel, showing where it had an overlap with research at BRICS—and hopefully it worked equally well in the other direction. ≣

Introduction to Linear Logic

Torben Braüner, BRICS, gave December 18, 1996, an introduction to Linear Logic. The course also covered Classical and Intuitionistic Logic. The course material appears as Lecture Series report [LS-96-6] below. ≣

Distributed Logics

January 20–21, 1997, *P. S. Thiagarajan*, SPIC Mathematical Institute, Madras, India, gave a course of two double lectures on Distributed Logics. Course description is on the web. ≣

Newly Appointed Researchers and Guests

Carsten Butz
Carsten Butz joined BRICS in January and is currently giving a course on categorical logic. He obtained his PhD degree from the Mathematical Department of the University of Utrecht, The Netherlands, in 1996. In his thesis he used topological models of infinitary first order theories to construct spaces that compute various well known cohomology theories (which are algebraic invariants). His recent research focuses on logic and category theory.

Arne Andersson
Arne Andersson received his PhD from Lund University in 1990 and he became a reader ("docent") in computer science in 1994. His main research area is fundamental data structures. In particular, his research has been focused on fast algorithms for sorting and searching. He also strives to combine simplicity with efficiency as simple methods are more likely to find their way into teaching and practice. Apart from his research interests, he is a committed teacher; in 1991 he was awarded "best teacher" by the students at Lund University. Life is more than work, among other things he enjoys playing the trumpet, as well as arranging and conducting music. Arne Andersson will spend every fourth week at the BRICS PhD School and the remaining time at the Department of Computer Science, LTH (Lund Institute of Technology), Sweden.

Ulrich Kohlenbach
Ulrich Kohlenbach got his PhD in 1990 from the Dept. of Mathematics of the University Frankfurt where he became Privatdozent in 1995. During the academic year 1996/97 he has been visiting assistant professor at the Department of Mathematics, University of Michigan, Ann Arbor. His main research interests so far have been in the area of proof theory with applications to concrete proofs: extraction of effective data from ineffective proofs, transformations of proofs and their complexity, proofs and functionals of finite type, intuitionistic reasoning, computational mathematics, (weak) fragments of arithmetic and analysis.
Ulrich Kohlenbach will join BRICS in July 1997.

Alessandro Panconesi
Alessandro Panconesi received his PhD in Computer Science from Cornell University in 1993. His thesis entitled "Locality in Distributed Computing" included work from a paper that won the Best Student Paper Award at the 1992 STOC. After returning to Europe, Alessandro held an ERCIM Fellowship, spending 6 months each at Amsterdam, Trondheim and Stockholm. He had a von Humboldt Fellowship at Freie Universität, Berlin and is currently at Nada (Numerical Analysis and Computing Science), KTH (Royal Institute of Technology), Stockholm, Sweden. Alessandro Panconesi will join the BRICS PhD School in September 1997.

David Toman
David Toman will be visiting BRICS in the fall '97 semester. He will join the research at BRICS as well as give introductory lectures on *Databases*. He received his M.S. (Mgr.) degree in computer science from Masaryk University, Brno, Czech Republic, in 1992 and his PhD from Kansas State University in 1996. Currently he is a NATO/NSERC Postdoctoral Fellow at the University of Toronto, Canada. His research interests include temporal, deductive, and constraint databases, logic-based languages for database and information systems, formal methods, logic programming (and other non-imperative paradigms), and programming languages. In his spare time he also likes to hack FreeBSD on his tiny Toshiba Libretto laptop.

Uwe Nestmann
Uwe Nestmann's main interests are the semantics of concurrent and distributed computation and, in particular, calculi for mobile processes. He has received his PhD degree in 1996 from the University of Erlangen-Nürnberg, Germany, for a dissertation on "determinacy and nondeterminacy" in the context of Pict, the concurrent implementation of the asynchronous π-calculus, where he focused on the investigation of choice encodings and type systems for detecting confluent behaviours. The PhD was carried out in tight collaboration with Benjamin Pierce during numerous visits at the Universities of Edinburgh and Cambridge, and finalised during a 6 month stay at INRIA Sophia-Antipolis. He is currently an ERCIM post-doc at INRIA Rocquencourt, where the distributed implementation of the π-calculus is developed, known as the join-calculus, so his current interests include distributed computing.
Uwe Nestmann will join BRICS in Aalborg in October 1997.

Augusto Burgueño Arjona
Augusto Burgueño Arjona is a PhD student at CERT (Centre d'Etudes et de Recherches de Toulouse), France. His research interests include real-time process algebras and verification of real-time systems and recently he is working on an extension of hybrid automata called Slope-Parametric Hybrid Automata (SPHA). He will be visiting BRICS in Aalborg from June to August as a BRICS summer student and work on integrating ideas of SPHA in the Uppaal tool suit.

Vincent Balat
Vincent Balat is a PhD student of Christine Paulin-Mohring from the École Normale Supérieure at Lyon, France. He will visit BRICS in July and August as a BRICS summer student. He will work on the lambda-calculus as a meta-language. Aside from his interest in theoretical computer science, Vincent also has a degree in oboe and cor anglais.

Zhe Yang
Zhe Yang is a PhD student of Robert Paige at Courant Institute of Mathematical Science, New York University, USA. His research focuses on areas such as programmming languages and program transformation techniques. His current interest is in applying various qualitative methods in programming languages to get quantitative improvements in algorithms. Zhe Yang will visit BRICS from mid August to the end of December 1997.

Dissertation Abstracts

Worst Case Efficient Data Structures

by Gerth Stølting Brodal

We study the design of efficient data structures. In particular we focus on the design of data structures where each operation has a worst case efficient implementations. The concrete problems we consider are *partial persistence*, implementation of *priority queues*, and implementation of *dictionaries*.

The first problem we consider is how to make bounded in-degree and out-degree data structures partially persistent, *i.e.*, how to remember old versions of a data structure for later access. A *node copying* technique of Driscoll *et al.* supports update steps in amortized constant time and access steps in worst case constant time. The worst case time for an update step can be linear in the size of the structure. We show how to extend the technique of Driscoll *et al.* such that update steps can be performed in worst case constant time on the pointer machine model.

We present two new comparison based priority queue implementations, with the following properties. The first implementation supports the operations FindMin, Insert and Meld in worst case constant time and Delete and DeleteMin in worst case time $O(\log n)$. The priority queues can be implemented on the pointer machine and require linear space. The second implementation achieves the same worst case performance, but furthermore supports DecreaseKey in worst case constant time. The space requirement is again linear, but the implementation requires auxiliary arrays of size $O(\log n)$. Our bounds match the best known amortized bounds (achieved by respectively binomial queues and Fibonacci heaps). The data structures presented are the first achieving these worst case bounds, in particular supporting Meld in worst case constant time. We show that these time bounds are optimal for all implementations supporting Meld in worst case time $o(n)$. We also present a tradeoff between the update time and the query time of comparison based priority queue implementations. Finally we show that any randomized implementation with expected amortized cost t comparisons per Insert and Delete operation has expected cost at least $n/2^{O(t)}$ comparisons for FindMin.

Next we consider how to implement priority queues on parallel (comparison based) models. We present time and work optimal priority queues for the CREW PRAM, supporting FindMin, Insert, Meld, DeleteMin, Delete and DecreaseKey in constant time with $O(\log n)$ processors. Our implementation is the first supporting all of the listed operations in constant time. To be able to speed up Dijkstra's algorithm for the single-source shortest path problem we present a different parallel priority data structure. With this specialized data structure we give a parallel implementation of Dijkstra's algorithm which runs in $O(n)$ time and performs $O(m \log n)$ work on a CREW PRAM. This represents a logarithmic factor improvement for the running time compared with previous approaches.

We also consider priority queues on a RAM model which is stronger than the comparison model. The specific problem is the maintenance of a set of n integers in the range $0..2^w - 1$ under the operations Insert, Delete, FindMin, FindMax and Pred (predecessor query) on a unit cost RAM with word size w bits. The RAM operations used are addition, left and right bit shifts, and bit-wise boolean operations. For any function $f(n)$ satisfying $\log\log n \leq f(n) \leq \sqrt{\log n}$, we present a data structure supporting FindMin and FindMax in worst case constant time, Insert and Delete in worst case $O(f(n))$ time, and Pred in worst case $O((\log n)/f(n))$ time. This represents the first priority queue implementation for a RAM which supports Insert, Delete and FindMin in worst case time $O(\log\log n)$ — previous bounds were only amortized. The data structure is also the first dictionary implementation for a RAM which supports Pred in worst case $O(\log n/\log\log n)$ time while having worst case $O(\log\log n)$ update time. Previous sublogarithmic dictionary implementations do not provide for updates that are significantly faster than queries. The best solutions known support both updates and queries in worst case time $O(\sqrt{\log n})$.

The last problem consider is the following dictionary problem over binary strings. Given a set of n binary strings of length m each, we want to answer d–queries, *i.e.*, given a binary query string of length m to report if there exists a string in the set within Hamming distance d of the query string. We present a data structure of size $O(nm)$ supporting 1–queries in time $O(m)$ and the reporting of all strings within Hamming distance 1 of the query string in time $O(m)$. The data structure can be constructed in time $O(nm)$. The implementation presented is the first achieving these optimal time bounds for the preprocessing of the dictionary and for 1–queries. The data structure can be extended to support the insertion of new strings in amortized time $O(m)$.

Trust and Dependence Analysis

by Peter Ørbæk

The two pillars of *trust analysis* and *dependence algebra* form the foundation of this thesis. Trust analysis is a static analysis of the run-time trustworthiness of data. Dependence algebra is a rich abstract model of data dependences in programming languages, applicable to several kinds of analyses.

Trust analysis tracks the data flow in a program with the aim of ensuring that appropriate validation checks are made on all data paths leading to "dangerous" operations, such as entering data into a private database, deleting files, etc.

Computers are increasingly being used to handle important transactions across the Internet. Legal documents and money orders are sent across the same network as is used to transfer e-mail and non-business related information. Thus separating these two kinds of information is becoming more and more of an issue. This problem is usually attacked using encryption and digital signatures for the important information flows between computers. But what if there, somewhere inside a program, is a data path between the reading of a message from the network and a dangerous operation depending on that message, such that the signature of the message is not checked on that path?

An example of an application where it is important to perform validity checks is in an HTTP (Hypertext Transfer Protocol [RFC1945,RFC2068]) server. Most such servers allow separate programs (so-called CGI scripts, for Common Gateway Interface) to be started on the server machine in response to requests from clients. The usual convention is that the web browser requests an URL of the form "`http://www.company.com/cgi-bin/`*program*" to start *program*. A CGI script can in principle be any program, so it is important that the server process checks that the requested program is one of the few programs allowed to be executed in this manner. In a complicated server program there are many data paths from the reception of an URL to the command to be executed. It is important that the checks for allowed programs are done on *all* these data paths. Trust analysis has been developed with the goal that a compiler will be able to provide the programmer with a guarantee that checks are present on all data paths between the reception of untrusted input and the execution of commands where only trustworthy (checked, validated) information should be present.

Since we want our analysis to be generally applicable, we do not consider the very specific tests that have to be done on input data in various situations, such as checking digital signatures or verifying pathnames against a known pattern. Devising these tests is still up to the careful program designer. Instead our analysis offers a "`trust`" construct that is meant to be applied to data after they have passed the specific validity checks, the analysis will then propagate this knowledge around the program. At points of the program where something dangerous is about to happen, such as starting another program or starting a transaction against a database, the programmer can write a "`check`" construct to ensure that the arguments can indeed be trusted, which means that they only depend on data that have actually passed the validity checks.

Dependence analysis is a more general framework that can be applied to different concrete analysis tasks. The data flow in a program is modeled as channels with certain properties determining which kinds of data can pass through the channels. Mathematically, a dependence algebra is a semi-lattice ordered semiring with certain additional axioms. Vector spaces over such algebras are studied, and a nontrivial construction of a dependence algebra for a may / must analysis is given.

One of the virtues of dependence algebra-based analysis is that, by emphasising the *connections* between slots manifest in the program, the analysis can be program-point sensitive. In type systems the underlying idea is to associate a type to each slot, such that a given slot is associated with a fixed (possibly polymorphic) type throughout the program. In contrast to this, a dependence-based analysis may allow a slot to hold different types of values at different points in the program without sacrificing detailed knowledge of the type of value in the slot at those points.

Contributions

The thesis introduces the notion of trust analysis, and studies its application to different kinds of languages. First a trust analysis for a simple WHILE language is given, both in the form of an abstract interpretation, and as a constraint generation problem. Then a trust analysis for an extended λ-calculus is given in the form of a type system, together with a type inference algorithm based on constraints (this part of the thesis is joint work with Jens Palsberg). Finally, an implementation of a trust analysis for the C programming language is described. The implementation is based on the dependence algebra technology.

The second contribution of the thesis is the development of the dependence algebra framework for program analyses. The algebraic framework is applied in the trust analyser for C, as well as for a soft type inference algorithm for action semantic equations.

An Axiomatic Approach to Adequacy

by Torben Braüner

This thesis studies adequacy for PCF-like languages in a general categorical framework. PCF (*P*rogramming language for *C*omputable *F*unctions) is a programming language based on a Curry-Howard interpretation of Intuitionistic Logic, namely the well-known typed λ-calculus, augmented with numerals and fixpoint constants. An adequacy result relates the denotational semantics of a program to its operational semantics; it typically states that a program terminates whenever the interpretation is non-bottom. The main concern of the thesis is to generalise to a linear version of PCF, called Linear PCF, the adequacy results known for standard PCF, keeping in mind that there exists a Girard translation from PCF to Linear PCF with respect to which the new constructs should be sound. Linear PCF is obtained by augmenting a Curry-Howard interpretation of Intuitionistic Linear Logic with numerals and fixpoint constants, appropriate for the linear context.

General adequacy results have usually been obtained in order-enriched categories, that is, categories where all hom-sets are typically cpos, maps are assumed to be continuous and fixpoint constants are interpreted using least upper bounds. One of the main contributions of the thesis is to propose a completely different approach to the problem of axiomatising those categories which yield adequate semantics for PCF and Linear PCF. Historically, the axioms of our categorical model were discovered essentially by extracting the properties of the order-structure of an order-enriched categorical model which are actually used to obtain an adequacy result. The order-structure is used to define fixpoints such that for any endomap f the chain of finite approximants $\{\perp; f^n\}_{n\in\omega}$ actually approximates the fixpoint $f^\sharp$ in an appropriate sense. So rather than assume the presence of an order-structure, we axiomatise what is actually needed; namely fixpoints which are approximated by their (formal) finite approximants in an appropriate sense stated in non-order-theoretic terms.

The starting point of our axiomatisation is a cartesian closed category where for each object B we assume the presence of an "undefined" map $\perp_B$: $1 \rightarrow B$ together with a *fixpoint operator*, that is, an operation which to a map $f : B \rightarrow B$ assigns a map $f^\sharp : 1 \rightarrow B$ such that $f^\sharp = f^\sharp; f$. We also assume the presence of an object N together with appropriate maps for dealing with numerals. The categorical axioms consist of a couple of equational (that is, first order) ones together with one non-equational, namely the axiom of rational openness on each fixpoint operator. A fixpoint operator is *rationally open* with respect to an object P iff for all maps $f : B \rightarrow B$ and $g : B \rightarrow P$ it is the case that

$$f^\sharp; g \neq \perp \;\Rightarrow\; \exists n \in \omega.\ \perp; f^n; g \neq \perp .$$

It is shown that this axiom is sufficient, and in a precise sense necessary, for adequacy. Thus, the hidden essential property of an order-theoretic model has been revealed: Definedness of an expression is determined by the (formal) finite approximants to the involved fixpoints. It is furthermore shown that our axioms induce an order-theoretic categorical model in a canonical way.

These ideas are developed in the intuitionistic case (standard PCF) as well as in the linear case (Linear PCF). Two concrete categories satisfying the axioms of the categorical model are given, namely the category of cpos and (strict) continuous functions, and the category of dI-domains and (linear) stable functions. Using instantiations to concrete models of the general adequacy results, various purely syntactic properties of Linear PCF are proved to hold.

Published as BRICS dissertation DS-96-4.

New in the BRICS Report Series, 1997 and 1996

ISSN 0909-0878

9 Jesper G. Henriksen and P. S. Thiagarajan. *A Product Version of Dynamic Linear Time Temporal Logic.* April 1997. 18 pp. To appear in *Concurrency Theory: 7th International Conference*, CONCUR '97 Proceedings, LNCS, 1997.

8 Jesper G. Henriksen and P. S. Thiagarajan. *Dynamic Linear Time Temporal Logic.* April 1997. 33 pp.

7 John Hatcliff and Olivier Danvy. *Thunks and the λ-Calculus (Extended Version).* March 1997. 55 pp. Extended version of article to appear in the *Journal of Functional Programming*.

6 Olivier Danvy and Ulrik P. Schultz. *Lambda-Dropping: Transforming Recursive Equations into Programs with Block Structure.* March 1997. 53 pp. Extended version of an article to appear in the 1997 ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM '97), Amsterdam, The Netherlands, June 1997.

5 Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. *First-Order Logic with Two Variables and Unary Temporal.* March 1997. To appear in *Twelfth Annual IEEE Symposium on Logic in Computer Science*, LICS '97 Proceedings.

4 Richard Blute, Josée Desharnais, Abbas Edalat, and Prakash Panangaden. *Bisimulation for Labelled Markov Processes.* March 1997. 48 pp. To appear in *Twelfth Annual IEEE Symposium on Logic in Computer Science*, LICS '97 Proceedings.

3 A Definability Theorem for First Order Logic. *Butz, Carsten and Moerdijk, Ieke.* March 1997. 10 pp.

2 David A. Schmidt. *Abstract Interpretation in the Operational Semantics Hierarchy.* March 1997. 33 pp.

1 Olivier Danvy and Mayer Goldberg. *Partial Evaluation of the Euclidian Algorithm (Extended Version).* January 1997. 16 pp.

62 P. S. Thiagarajan and Igor Walukiewicz. *An Expressively Complete Linear Time Temporal Logic for Mazurkiewicz Traces.* December 1996. i+13 pp. To appear in *Twelfth Annual IEEE Symposium on Logic in Computer Science*, LICS '97 Proceedings.

61 Sergei Soloviev. *Proof of a Conjecture of S. Mac Lane.* December 1996. 53 pp. Extended abstract appears in Pitt, Rydeheard and Johnstone, editors, *Category Theory and Computer Science: 6th International Conference*, CTCS '95 Proceedings, LNCS 953, 1995, pages 59–80.

60 Johan Bengtsson, Kim G. Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. UPPAAL *in 1995.* December 1996. 5 pp. Appears in Margaria and Steffen, editors, *Tools and Algorithms for The Construction and Analysis of Systems: 2nd International Workshop*, TACAS '96 Proceedings, LNCS 1055, 1996, pages 431–434.

59 Kim G. Larsen, Paul Pettersson, and Wang Yi. *Compositional and Symbolic Model-Checking of Real-Time Systems.* December 1996. 12 pp. Appears in *16th IEEE Real-Time Systems Symposium*, RTSS '95 Proceedings, 1995.

58 Johan Bengtsson, Kim G. Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. UPPAAL *— a Tool Suite for Automatic Verification of Real–Time Systems.* December 1996. 12 pp. Appears in Alur, Henzinger and Sontag, editors, *DIMACS Workshop on Verification and*

Control of Hybrid Systems, HYBRID '96 Proceedings, LNCS 1066, 1996, pages 232–243.

57 Kim G. Larsen, Paul Pettersson, and Wang Yi. *Diagnostic Model-Checking for Real-Time Systems*. December 1996. 12 pp. Appears in Alur, Henzinger and Sontag, editors, *DIMACS Workshop on Verification and Control of Hybrid Systems*, HYBRID '96 Proceedings, LNCS 1066, 1996, pages 575–586.

56 Zine-El-Abidine Benaissa, Pierre Lescanne, and Kristoffer H. Rose. *Modeling Sharing and Recursion for Weak Reduction Strategies using Explicit Substitution*. December 1996. 35 pp. Appears in Kuchen and Swierstra, editors, *8th International Symposium on Programming Languages, Implementations, Logics, and Programs*, PLILP '96 Proceedings, LNCS 1140, 1996, pages 393–407.

55 Kåre J. Kristoffersen, François Laroussinie, Kim G. Larsen, Paul Pettersson, and Wang Yi. *A Compositional Proof of a Real-Time Mutual Exclusion Protocol*. December 1996. 14 pp. To appear in Dauchet and Bidoit, editors, *Theory and Practice of Software Development. 7th International Joint Conference CAAP/FASE*, TAPSOFT '97 Proceedings, LNCS, 1997.

54 Igor Walukiewicz. *Pushdown Processes: Games and Model Checking*. December 1996. 31 pp. Appears in Alur and Henzinger, editors, *8th International Conference on Computer-Aided Verification*, CAV '96 Proceedings, LNCS 1102, 1996, pages 62–74.

53 Peter D. Mosses. *Theory and Practice of Action Semantics*. December 1996. 26 pp. Appears in Penczek and Szalas, editors, *Mathematical Foundations of Computer Science: 21st International Symposium*, MFCS '96 Proceedings, LNCS 1113, 1996, pages 37–61.

52 Claus Hintermeier, Hélène Kirchner, and Peter D. Mosses. *Combining Algebraic and Set-Theoretic Specifications (Extended Version)*. December 1996. 26 pp. Appears in Haveraaen, Owe and Dahl, editors, *Recent Trends in Data Type Specification: 11th Workshop on Specification of Abstract Data Types, joint with 8th COMPASS Workshop*, Selected Papers, LNCS 1130, 1996, pages 255–274.

51 Claus Hintermeier, Hélène Kirchner, and Peter D. Mosses. *R^n- and G^n-Logics*. December 1996. 19 pp. Appears in Gilles, Heering, Meinke and Möller, editors, *Higher-Order Algebra, Logic, and Term-Rewriting: 2nd International Workshop*, HOA '95 Proceedings, LNCS 1074, 1996, pages 90–108.

50 Aleksandar Pekeč. *Hypergraph Optimization Problems: Why is the Objective Function Linear?* December 1996. 10 pp.

49 Dan S. Andersen, Lars H. Pedersen, Hans Hüttel, and Josva Kleist. *Objects, Types and Modal Logics*. December 1996. 20 pp. To be presented at the *4th International Workshop on the Foundations of Object-Oriented*, FOOL4, 1997.

48 Aleksandar Pekeč. *Scalings in Linear Programming: Necessary and Sufficient Conditions for Invariance*. December 1996. 28 pp.

47 Aleksandar Pekeč. *Meaningful and Meaningless Solutions for Cooperative N-person Games*. December 1996. 28 pp.

46 Alexander E. Andreev and Sergei Soloviev. *A Decision Algorithm for Linear Isomorphism of Types with Complexity $Cn(log^2(n))$*. November 1996. 16 pp.

45 Ivan B. Damgård, Torben P. Pedersen, and Birgit Pfitzmann. *Statistical Secrecy and Multi-Bit Commitments*. November 1996. 30 pp. To appear in *IEEE Transactions on Information Theory*.

44 Glynn Winskel. *A Presheaf Semantics of Value-Passing Processes*. November 1996. 23 pp. Extended and revised version of paper appearing in Montanari and Sassone, editors, *Concurrency Theory: 7th International Conference*, CONCUR '96 Proceedings, LNCS 1119, 1996, pages 98–114.

43 Anna Ingólfsdóttir. *Weak Semantics Based on Lighted Button Pressing Experiments: An Alternative Characterization of the Readiness Semantics*. November 1996. 36 pp. Extended abstract presented at the *10th Annual International Conference of the European Association for Computer Science Logic*, CSL '96.

42 Gerth Stølting Brodal and Sven Skyum. *The Complexity of Computing the k-ary Composition of a Binary Associative Operator*. November 1996. 15 pp.

41 Stefan Dziembowski. *The Fixpoint Bounded-Variable Queries are PSPACE-Complete*. November 1996. 16 pp. Presented at the *10th Annual International Conference of the European Association for Computer Science Logic*, CSL '96.

40 Gerth Stølting Brodal, Shiva Chaudhuri, and Jaikumar Radhakrishnan. *The Randomized Complexity of Maintaining the Minimum*. November 1996. 20 pp. To appear in a special issue of *Nordic Journal of Computing* devoted to the proceedings of SWAT '96. Appears in Karlson and Lingas, editors, *Algorithm Theory: 5th Scandinavian Workshop*, SWAT '96 Proceedings, LNCS 1097, 1996, pages 4–15.

39 Hans Hüttel and Sandeep Shukla. *On the Complexity of Deciding Behavioural Equivalences and Preorders – A Survey*. October 1996. 36 pp.

38 Hans Hüttel and Josva Kleist. *Objects as Mobile Processes*. October 1996. 23 pp. Presented at the *12th Annual Workshop on Mathematical Foundations of Programming Languages*, MFPS '96, (Boulder, Colorado, USA, June 3–5, 1996).

37 Gerth Stølting Brodal and Chris Okasaki. *Optimal Purely Functional Priority Queues*. October 1996. 27 pp. To appear in *Journal of Functional Programming*, 6(6), December 1996.

36 Luca Aceto, Willem Jan Fokkink, and Anna Ingólfsdóttir. *On a Question of A. Salomaa: The Equational Theory of Regular Expressions over a Singleton Alphabet is not Finitely Based*. October 1996. 16 pp.

35 Gian Luca Cattani and Glynn Winskel. *Presheaf Models for Concurrency*. October 1996. 16 pp. Presented at the *10th Annual International Conference of the European Association for Computer Science Logic*, CSL '96.

34 John Hatcliff and Olivier Danvy. *A Computational Formalization for Partial Evaluation (Extended Version)*. October 1996. 67 pp. To appear in *Mathematical Structures in Computer Science*.

33 Jonathan F. Buss, Gudmund Skovbjerg Frandsen, and Jeffrey Outlaw Shallit. *The Computational Complexity of Some Problems of Linear Algebra*. September 1996. 39 pp. Revised version to appear in *STACS '97: 14th Annual Symposium on Theoretical Aspects of Computer Science Proceedings*, LNCS, 1997.

32 P. S. Thiagarajan. *Regular Trace Event Structures*. September 1996. 34 pp.

31 Ian Stark. *Names, Equations, Relations: Practical Ways to Reason about 'new'*. September 1996. ii+22 pp. To appear in *Typed Lambda Calculi and Applications: 3rd International Conference*, TLCA '97 Proceedings, LNCS, 1997.

30 Arne Andersson, Peter Bro Miltersen, and Mikkel Thorup. *Fusion Trees can be Implemented with AC^0 Instructions only*. September 1996. 8 pp.

29 Lars Arge. *The I/O-Complexity of Ordered Binary-Decision Diagram Manipulation*. August 1996. 35 pp. An extended abstract version appears in Staples, Eades, Kato, and Moffat, editors, *Algorithms and Computation: 6th International Symposium*, ISAAC '95 Proceedings, LNCS 1004, 1995, pages 82–91.

28 Lars Arge. *The Buffer Tree: A New Technique for Optimal I/O Algorithms*. August 1996. 34 pp. This report is a revised and extended version of the BRICS Report RS-94-16. An extended abstract appears in Akl, Dehne, Sack, and Santoro, editors, *Algorithms and Data Structures: 4th Workshop*, WADS '95 Proceedings, LNCS 955, 1995, pages 334–345.

27 Devdatt P. Dubhashi, Volker Priebe, and Desh Ranjan. *Negative Dependence Through the FKG Inequality*. July 1996. 15 pp.

26 Nils Klarlund and Theis Rauhe. *BDD Algorithms and Cache Misses*. July 1996. 15 pp.

25 Devdatt P. Dubhashi and Desh Ranjan. *Balls and Bins: A Study in Negative Dependence*. July 1996. 27 pp.

New in the BRICS Notes Series, 1996

ISSN 0909-3206

15 CoFI. *CASL – The CoFI Algebraic Specification Language; Tentative Design: Language Summary*. December 1996. viii+34 pp.

14 Peter D. Mosses. *A Tutorial on Action Semantics*. December 1996. 46 pp. Tutorial notes for FME '94 (Formal Methods Europe, Barcelona, 1994) and FME '96 (Formal Methods Europe, Oxford, 1996).

13 Olivier Danvy, editor. *Proceedings of the Second ACM SIGPLAN Workshop on Continuations, CW '97* (ENS, Paris, France, 14 January, 1997), December 1996. 166 pp.

12 Mandayam K. Srivas. *A Combined Approach to Hardware Verification: Proof-Checking, Rewriting with Decision Procedures and Model-Checking; Part II: Articles. BRICS Autumn School on Verification*. October 1996. 56 pp.

11 Mandayam K. Srivas. *A Combined Approach to Hardware Verification: Proof-Checking, Rewriting with Decision Procedures and Model-Checking; Part I: Slides. BRICS Autumn School on Verification*. October 1996. 29 pp.

10 Robert Pollack. *What we Learn from Formal Checking; Part III: Formalization is Not Just Filling In Details. BRICS Autumn School on Verification*. October 1996. iv+42 pp.

9 Robert Pollack. *What we Learn from Formal Checking; Part II: Using Type Theory: An Introduction. BRICS Autumn School on Verification*. October 1996. iv+71 pp.

8 Robert Pollack. *What we Learn from Formal Checking; Part I: How to Believe a Machine-Checked Proof. BRICS Autumn School on Verification*. October 1996. iv+19 pp.

7 Tom F. Melham. *Some Research Issues in Higher Order Logic Theorem Proving. BRICS Autumn School on Verification*. October 1996. 15 pp.

6 Gerard J. Holzmann. *On-the-Fly Model Checking Tutorial. BRICS Autumn School on Verification*. October 1996. 31 pp.

5 Thomas A. Henzinger. *Automatic Verification of Real-Time and Hybrid Systems. BRICS Autumn School on Verification*. October 1996. 28 pp.

4 Edmund M. Clarke, Jr. *Symbolic Model Checking. BRICS Autumn School on Verification*. October 1996. 55 pp.

3 David A. Basin. *Verification Based on Monadic Logic. BRICS Autumn School on Verification*. October 1996. 43 pp.

2 Allan Cheng, Kim G. Larsen, and Mogens Nielsen, editors. *Programme and Abstracts of the BRICS Autumn School on Verification* (Aarhus, Denmark, October 28 – November 1, 1996), August 1996. ii+18pp.

6 Torben Braüner. *Introduction to Linear Logic*. December 1996. iiiv+55 pp.

Abstract

The main concern of this report is to give an introduction to Linear Logic. For pedagogical purposes we shall also have a look at Classical Logic as well as Intuitionistic Logic. Linear Logic was introduced by J.-Y. Girard in 1987 and it has attracted much attention from computer scientists, as it is a logical way of coping with resources and resource control. The focus of this technical report will be on proof-theory and computational interpretation of proofs, that is, we will focus on the question of how to interpret proofs as programs and reduction (cut-elimination) as evaluation. We first introduce Classical Logic. This is the fundamental idea of the proofs-as-programs paradigm. Cut-elimination for Classical Logic is highly non-deterministic; it is shown how this can be remedied either by moving to Intuitionistic Logic or to Linear Logic. In the case on Linear Logic we consider Intuitionistic Linear Logic as well as Classical Linear Logic. Furthermore, we take a look at the Girard Translation translating Intuitionistic Logic into Intuitionistic Linear Logic. Also, we give a brief introduction to some concrete models of Intuitionistic Linear Logic. No proofs will be given except that a proof of cut-elimination for the multiplicative fragment of Classical Linear Logic is included in an appendix.

Contents

5 Devdatt P. Dubhashi. *What Can't You Do With LP?* December 1996. viii+23 pp.

Abstract

These notes from the BRICS course "Pearls of Theory" are an introduction to Linear Programming and its use in solving problems in Combinatorics and in the design and analysis of algorithms for combinatorial problems.

Contents

4 Sven Skyum. *A Non-Linear Lower Bound for Monotone Circuit Size.* December 1996. viii+14 pp.

Abstract

In complexity theory we are faced with the frustrating situation that we are able to prove very few non trivial lower bounds. In the area of combinatorial complexity theory which this note is about, the situation can be described quite precisely in the sense that although almost all functions are very complex (have exponential circuit size), no one has been able to prove any non-linear lower bounds for explicitly given functions.

One alley which is often taken is to restrict the models to make larger lower bounds more likely. Until 1985 no non-linear lower bounds were known for monotone circuit size either (the best lower bound was $4n$). In 1985, Razborov [1] proved a *super polynomial* lower bound for a specific family of Boolean functions.

Razborov proved the following:

> Any family of monotone Boolean circuits accepting graphs containing cliques of size $\lfloor n/2 \rfloor$ has super polynomial size when the graphs are represented by their incidence matrices.

The main purpose of this note is to give a "simple" version of Razborov's proof. This leads to a weaker result but the proof contains all the ingredients of Razborov's proof.

We will prove:

> Any family of monotone Boolean circuits accepting graphs containing cliques of size 3 (*triangles*) has size $\Omega(n^3/\log^4 n)$.

In Section 1 we introduce Boolean functions and the complexity model we are going to use, namely *Boolean circuits*. In Section 2 the appropriate definitions of graphs are given and some combinatorial properties about them are proven. Section 3 contains the proof of the main result.

[1] A. A. Razborov: Lower bounds on the monotone complexity of some Boolean functions, *Dokl. Akad. Nauk SSSR 281(4) (1985) 798 - 801 (In Russian); English translation in: Soviet Math. Dokl. 31 (1985) 354–57.*

Contents

3 Kristoffer H. Rose. *Explicit Substitution – Tutorial & Survey*. September 1996. v+150 pp.

Abstract

These lecture notes are from the BRICS mini-course "Explicit Substitution" taught at University of Aarhus, October 27, 1996.

We give a coherent overview of the area of explicit substitution and some applications. The focus is on the *operational* or *syntactic* side of things, in particular we will not cover the areas of semantics and type systems for explicit substitution calculi. Emphasis is put on providing a universal understanding of the very different techniques used by various authors in the area.

Contents

- 3 Explicit Addresses
    - 3.1 λ-graphs and Explicit Sharing
    - 3.2 Explicit Substitution & Sharing
    - 3.3 Discussion
- 4 Higher-Order Rewriting and Functional Programs
    - 4.1 Combinatory Reduction Systems (CRS)
    - 4.2 Explicit Substitutes
    - 4.3 Explicit Addresses
    - 4.4 CRS and Functional Programs
    - 4.5 Discussion
- 5 Strategies and Abstract Machines
    - 5.1 Strategies for λ-Calculus
    - 5.2 Calculi for Weak Reduction with Sharing
    - 5.3 Reduction Strategies
    - 5.4 Constructors and Recursion
    - 5.5 A Space Leak Free Calculus
    - 5.6 Discussion
- A Appendices: Preliminaries
    - A.1 Reductions
    - A.2 Inductive Notations
    - A.3 λ-Calculus.

2 Susanne Albers. *Competitive Online Algorithms*. September 1996. iix+57 pp.

Abstract

These lecture notes are from the mini-course "Competitive Online Algorithms" taught at Aarhus University, August 27–29, 1996.

The mini-course consisted of three lectures. In the first lecture we gave basic definitions and presented important techniques that are used in the study on online algorithms. The paging problem was always the running example to explain and illustrate the material. We also discussed the k-server problem, which is a very well-studied generalisation of the paging problem.

The second lecture was concerned with self-organising data structures, in particular self-organising linear lists. We presented results on deterministic and randomised online algorithms. Furthermore, we showed that linear lists can be used to build very effective data compression schemes and reported on theoretical as well as experimental results.

In the third lecture we discussed three application areas in which interesting online problems arise. The areas were (1) distributed data management, (2) scheduling and load balancing, and (3) robot navigation and exploration. In each of these fields we gave some important results.

Contents

- 1 Online algorithms and competitive analysis
    - 1.1 Basic definitions
    - 1.2 Results on deterministic paging algorithms
- 2 Randomisation in online algorithms
    - 2.1 General concepts
    - 2.2 Randomised paging algorithms against oblivious adversaries
- 3 Proof techniques
    - 3.1 Potential functions
    - 3.2 Yao's minimax principle
- 4 The k-server problem
- 5 The list update problem
    - 5.1 Deterministic online algorithms
    - 5.2 Randomised online algorithms
    - 5.3 Average case analyses of list update algorithms
- 6 Data compression based on linear lists
    - 6.1 Theoretical results
    - 6.2 Experimental results
    - 6.3 The compression algorithm by Burrows and Wheeler
- 7 Distributed data management
    - 7.1 Formal definition of migration and replication problems

1 Lars Arge. *External-Memory Algorithms with Applications in Geographic Information Systems*. September 1996. iix+53 pp.

Abstract

In the design of algorithms for large-scale applications it is essential to consider the problem of minimising Input/Output (I/O) communication. Geographical information systems (GIS) are good examples of such large-scale applications as they frequently handle huge amounts of spatial data. In this note we survey the recent developments in external-memory algorithms with applications in GIS. First we discuss the Aggarwal-Vitter I/O-model and illustrate why normal internal-memory algorithms for even very simple problems can perform terribly in an I/O-environment. Then we describe the fundamental paradigms for designing I/O-efficient algorithms by using them to design efficient sorting algorithms. We then go on and survey external-memory algorithms for computational geometry problems—with special emphasis on problems with applications in GIS—and techniques for designing such algorithms: Using the orthogonal line segment intersection problem we illustrate the *distribution-sweeping* and the *buffer tree* techniques which can be used to solve a large number of important problems. Using the batched range searching problem we introduce the *external segment tree*. We also discuss an algorithm for the reb/blue line segment intersection problem—an important subproblem in map overlaying. In doing so we introduce the *batched filtering* and the *external fractional cascading* techniques. Finally, we shortly describe TPIE—a Transparent Parallel I/O Environment designed to allow programmers to write I/O-efficient programs.

These lecture notes were made for the CISM Advanced School on Algorithmic Foundations of Geographic Information Systems, September 16–20, 1996, Udine, Italy.

Contents

BRICS Dissertations Series, 1996

ISSN 1396-7002

4 Torben Braüner. *An Axiomatic Approach to Adequacy*. November 1996. Ph.D. thesis. 168 pp.

3 Lars Arge. *Efficient External-Memory Data Structures and Applications*. August 1996. Ph.D. thesis. xii+169 pp.

2 Allan Cheng. *Reasoning About Concurrent Computational Systems*. August 1996. Ph.D. thesis. xiv+229 pp.

1 Urban Engberg. *Reasoning in the Temporal Logic of Actions — The design and implementation of an interactive computer system*. August 1996. Ph.D. thesis. xvi+222 pp.

News

BRICS International PhD School in Computer Science

BRICS now offers an international PhD School in Computer Science at the University of Aarhus. The school provides a PhD programme in computer science, and admits 10-12 students annually. It offers a substantial number of grants for Danish as well as foreign students. The school is funded by the Danish National Research Foundation, and institutionally the school is a part of the Department of Computer Science at University of Aarhus, i.e., it operates within the university regulations, and all PhD candidates get their degree from the University of Aarhus.

The purpose of the PhD School is: to create a truly international graduate school in Denmark offering a programme of courses and project work of high scientific quality, and to recruit Danish as well as foreign PhD students of the highest international standards. It provides an excellent research environment and scientific training facilities, and aims at making its PhD graduates attractive for a wide spectrum of employers—in private and public research and development institutions, both in Denmark and abroad.

It is the hope to attract Computer Science students with four years of full time study and trained researcher interested in joining the PhD School for a shorter or longer period as a teacher and researcher.

The official inauguration of the PhD School took place on February 24, 1997, cf. figure 3. The first students will be admitted on September 1, 1997.

Admission and PhD Student Grants

The admission requirements for PhD studies in computer science are knowledge corresponding to four years of full-time study in computer science at a recognised university. All students will be admitted to the BRICS PhD School studies at some level in the four years of PhD studies, transferring previous relevant educational activities.

There will be a number PhD Student Grants ranging from financial aid to tuition waiver and full studentship.

See the BRICS web pages for further information and instructions on how to apply.

Tools for Automated Verification of Embedded Software – A Collaboration between Basic Research and Industry

by Arne Skou

The danish software company Beologic A/S has recently initiated a research project in collaboration with the section for computer based systems at DTU (Technical University of Denmark) and the concurrency group at BRICS. The collaboration is supported by the danish Centre for In-

Figure 3: Inaugural Talk of *Mogens Nielsen*, director of the new PhD School. In the foreground from the left: *Peder Olesen Larsen* (director of the Danish National Research Foundation), *Henning Lehmann* (rector of the University of Aarhus) and *Ole Vig Jensen* (the Danish Minister of Education).

formation Technology (CIT), and it is a primary goal to support further development of the company's *visualState* tool system. This development will be based on recent results from basic research – in particular within model checking.

Embedded software is part of almost all modern household equipment and also of advanced equipment for control and monitoring in large industrial systems. The complexity of such software is a prerequisite for the equipment functionality, and it is a well recognised fact that the development of verification tools for embedded software is an important competitive parameter for the equipment suppliers. During recent years, there has been a minor revolution of such tools wrt. their functionality and application areas, and it is a necessity that Danish companies are offered the possibility to exploit this development. The project will make a contribution to improve the present market.

A few words about Beologic A/S: The company is a fast growing software company with a major activity on development and marketing of the *visualState* tool for embedded software development. The product supports design and code generation of state machines as well as state space analysis of up to 300 million states. The development of *visualState* is based on more than 10 years of work on Bang & Olufsen products. The company has more than 50 national costumers (including B&O, Grundfos, Danfoss and Terma Electronics), and it is looking forward to offering improved tool performance based on the research project.

Contact persons:

Beologic A/S: Technical Director Niels Bo Theilgaard, phone +45 4453 8888.

Technical University of Denmark: professor Jørgen Staunstrup, phone +45 4525 3740.

BRICS: professor Kim G. Larsen, phone +45 9635 8080.

Calendar of Events

Date	Event
29–30 Apr	Mini-course on Algebraic Theory of Automata, Temporal Logic and Expressiveness
1–2 May	Mini-course on Information Theoretic Security in Cryptography
5–6 May	Combinatorial Optimisation Meeting
Mid May	Mini-course on First Order Logic with Dependent Sorts, Simulation, and Higher Dimensional Categories
20–23 May	Mini-course on Inductive Logic Programming
30 Jun – 2 Jul	Combinatorial Pattern Matching, CPM '97
23–29 Aug	Computer Science Logic, CSL '97

BRICS Address and World Wide Web

Please **note** that the ftp area of the Centre now can be accessed directly from the BRICS ftp address. The old access way, however, will continue to be valid.

For further information, hard copies of information material and reports of the BRICS Series as well as enrolment into the BRICS newsgroup, please contact **BRICS** at

Telephone:	+45 8942 3360
Telefax:	+45 8942 3255
Internet:	<BRICS@brics.dk>

or, in writing, to

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark.

You can get access to information from BRICS through World Wide Web (WWW) and anonymous FTP. To connect to the BRICS WWW entry, open the URL:

```
http://www.brics.dk/
```

The BRICS WWW entry contains updated information about most of the topics covered in the newsletter as well as access to electronic copies of information material and reports of the BRICS Series (look under Publications).

To access the information material and reports of the BRICS Series via anonymous FTP do the following:

```
ftp ftp.brics.dk
get README.
```

BRICS *Newsletter*

ISSN 0909-6043

Editors: Glynn Winskel & Uffe H. Engberg
Lay-out: Uffe H. Engberg
Publisher: BRICS
Print: Departments of Mathematical Sciences
University of Aarhus