BRICS Newsletter

Basic Research in Computer Science No 10, October 1999

Contents

Welcome

Welcome to the tenth 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.

BRICS has been busy, and looking back we can see some new trends. We have held our first meeting on combinatorial optimisation, backed up by courses on randomisation and approximation algorithms and days devoted to coding theory. There have been new initiatives in biological computing, meetings and courses on evolutionary computation, biological concepts for adaptive and distributed algorithms and an introduction to computer vision. Process algebras are being turned to matters of security, evident in all but one of the four mini-courses on concurrency. The central role of the lambda calculus in computer science goes back to the pioneering work of Christopher Strachey and Dana Scott in providing a mathematical semantics of programming languages, and is still a topic of research—we had mini-courses covering optimal graph reduction of the lambda calculus. It will soon be 25 years since the death of Christopher Strachey. In commemoration Olivier Danvy is editing a special issue of "Higher-Order and Symbolic Computation" dedicated to his work (see the short news on page 30). The subject Strachey began is now thriving—our summer school in semantics attracted over 70 students.

You'll find a description of these and other recent and future BRICS activities in this newsletter.

Coming Events

For details and updates, see the BRICS Activities web page:

www.brics.dk/Activities.

Third International Workshop on Action Semantics

The Third International Workshop on Action Semantics is to be held in Recife, Brazil, 15–16 May 2000, as a satellite event of the annual Brazilian Symposium on Programming Languages. See the Action Semantics home page www.brics.dk/Projects/AS/ for further details.

Workshop on Probabilistic Methods in Combinatorial Optimisation

BRICS is planning to host a meeting late August of year 2000 with the title *Probabilistic Methods in Combinatorial Optimisation*.

Following what seems to be a general trend in Algorithms & Complexity, probabilistic techniques are playing an ever increasing role in combinatorial optimisation, notably in the field of approximation algorithms. Besides having as many high quality technical presentations as possible, we would like to "popularise" results and methods to the benefit of communities which study, albeit from a different perspective, the same kind of problems. For this reason, in the hope of fostering both the exchange and the critical appraisal of ideas, we will try to ensure the participation of "probabilists who do not work in combinatorial optimisation" and of "combinatorial optimisation experts not working with probabilities".

The master of ceremonies will be *Michal Karonski*, and *Adam Mickiewicz* of Emory University, and *Alessandro Panconesi*, one of our beloved regular guests, now at the University of Bologna. The local agent will be *Tibor Jordán*, who kindly agreed to take all responsibility in case things go awry.

Michal Karonski is co-editor-in-chief of the high quality journal Random Structures & Algorithms. The journal will devote a special issue to the event, most likely with the following format. The Scientific Advisory Board will ask young researchers to write surveys on specific topics. Besides the organisers, the board so far consists of the following people: *Noga Alon*, *Nati Linial*, *Laszlo Lovasz*, and *David Shmoys*. One or two more people might join later.

Major Events in 2001–02

For the years 2001 and 2002 we just outline the current known major events.

- May, 2001, Aarhus, MFPS, 17th Conference on the Mathematical Foundations of Programming Semantics. The workshop will be held back to back with PADO.
- May, 2001, Aarhus, PADO, 2nd workshop on Programs as Data Objects.
- June/July, 2001, Aarhus, EFF Summer School on Logical Methods.
- August, 2001, Aalborg, CONCUR, 12th International Conference on Concurrency Theory.
- August, 2001, Aarhus, ESA, 9th Annual European Symposium on Algorithms. The workshop will be held back to back with WAE.
- August, 2001, Aarhus, WAE, 5th Workshop on Algorithm Engineering.
- 2002, Aarhus, EEF Summer School on Massive Data Sets.

Figure 1: Participants of the BRICS PhD workshop at the West Coast of Denmark.

Reports on Events

Introduction to Evolutionary Computation

September 14, 16, 21 and 23, 1998, *Zbigniew Michalewicz*, University of North Carolina at Charlotte, USA, gave four double lectures on Introduction to Evolutionary Computation.

BRICS PhD Workshop

On October 22–24, 1998, BRICS had a retreat on the theme of "meta-issues" of research and PhD studies in computer science. The agenda included sessions on issues like:

- Research, Writing, Speaking, and Refereeing Skills
- Responsible Conduct in Research
- Library and Bibliographic data bases
- Working in academia and industry
- Patenting

The retreat held at Fjordgården, Ringkøbing, was primarily aimed at PhD students, but also the more senior participants found the talks and discussions very valuable. Figure 1 shows the participants in the BRICS PhD workshop.

Biological Concepts for Adaptive and Distributed Algorithms

November 9–11, 16 and 17, 1998, *Thiemo Krink*, BRICS, *Per Bak*, Niels Bohr Institute, University of Copenhagen, *Freddy B. Christiansen*, Department of Genetics and Ecology, University of Aarhus, and *Erik Baatrup*, Department of Zoology, University of Aarhus, gave five double lectures on Biological Concepts for Adaptive and Distributed Algorithms.

A Formal Calculus for Distributed Agents

November 12, 18, 20 and 23, 1998, *Matthew Hennessy*, School of Cognitive and Computing Sciences, University of Sussex, United Kingdom, gave four double lectures on A Formal Calculus for Distributed Agents.

Which π-Calculus are we Talking About?

November 30, December 2, 7 and 9, 1998, *Paola Quaglia*, BRICS, gave four double lectures on the π-calculus and related calculi.

Workshop on Partial Evaluation and Semantics-Based Program Manipulation

The 1999 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation, PEPM '99, took place in San Antonio, Texas, USA, on the 22nd and 23rd of January, 1999, following POPL'99 and right next to Fort Alamo.

It consisted of 13 contributed papers, as well as three invited talks by *Alan Bawden*, Brandeis University, USA, *Charles Consel*, Université de Rennes, France, and *Olin Shivers*, MIT, USA. The 13 papers were selected among 24 submissions. These submissions came from all over the world: USA, United Kingdom, France, Germany, Denmark, Spain, Australia, New Zealand, Japan, and Singapore, incidentally all from academia. 85 reviews were generated (3.5 per submission). The PC meeting was electronic and took place in a week. The notifications included both the reviews and, when it was possible, a synthesis of the relevant part of the PC meeting.

The BRICS report [NS-99-1] served as an informal proceedings. In addition, a special issue of Higher-Order and Symbolic Computation has been dedicated to selected papers from PEPM '99. The first part (Vol. 12, No. 4) is already in press. The second part is scheduled to appear in early 2000. At the meeting proper, and courtesy of BRICS, a gadget was issued that turned out to be quite popular: our BRICS pen, labelled "PEPM '99" on the other side.

Control Flow Analysis for the π-Calculus with Applications to Security

February 16 and 18, 1999, *Pierpaolo Degano*, Department of Computer Science, Università di Pisa, Italy, gave two double lectures on Control Flow Analysis for the π-calculus with Applications to Security.

An Overview of Lambda-Calculus Optimal Reductions and of their Implementation

February 24–26, 1999, *Stefano Guerrini*, Department of Computer Science, Queen Mary and Westfield College, London, United Kingdom, gave three double lectures on An Overview of Lambda-Calculus Optimal Reductions and of their Implementation.

Testing of Reactive Systems

March 2 and 3, 1999, BRICS in Aalborg hosted a mini workshop on testing of reactive systems. The main purpose was to exchange ideas about approaches and frameworks for test of reactive, real-time and hybrid systems with the group headed by professor *Jan Peleska*, Faculty of Mathematics and Computer Science, University of Bremen, Germany.

Figure 3: *Brian Nielsen* and *Jan Peleska* at the Testing of Reactive Systems mini workshop.

Figure 2: Participants of the Summer School in Semantics of Computation at the Moesgaard beach.

From Bremen participated Jan Peleska and Cornelia Zahlten. From Aalborg came Kim G. Larsen, Brian Nielsen, Paul Petterson, Anders P. Ravn, Arne Skou with Kåre J. Kristoffersen, Gerd Behrmann, Anna Ingólfsdóttir and students joining for some sessions. From Technical University of Denmark came Jens Christian Godskesen, and Thomas Hune from the Aarhus wing of BRICS joined us for the last day.

The first day featured a presentation of the work in Bremen and its background by Jan. It continued with a presentation of the Uppaal verification tool by Kim and Paul. The afternoon ended with a discussion of a number of recurring topics:

- User interface(s) to the tools
- Practical experiences and success criteria
- Features of Real-time Automata
- Compositionality

The second day had presentations of current work by Brian and Jens Christian. Brian presented his approach to test derivation for the restricted class of Event Recording Automata, while Jens Christian presented work on Interface testing. A discussion session shed some light on the differences and similarities between approaches by applying the Bremen approach to a simple example from Brian's work. In particular, it seems to be a useful idea to limit the test cases through introduction of environment assumptions, and in the real-time case through stability assumptions about the implementation.

Second International Workshop on Action Semantics, AS '99

AS '99, the 2nd International Workshop on Action Semantics, was held as a one-day satellite event of ETAPS '99 in Amsterdam, The Netherlands, and attended by 18 participants. As can be seen from the workshop programme and from the contributed papers collected in the proceedings [NS-99-3], much interesting work was presented and discussed during the workshop, focusing on tool support for Action Semantics, recent action-semantic descriptions, theoretical foundations, and prospects for the future of Action Semantics.

Modelling and Verifying Authentication Protocols

March 2, 4, 9 and 11, 1999, *Sanjiva Prasad*, IIT (Indian Institute of Technology), Delhi, India, and BRICS, University of Aarhus, Denmark, gave four double lectures on Modelling and Verifying Authentication Protocols.

Summer School in Semantics of Computation

On May 3–7, 1999, Glynn Winskel, as part of the European Educational Forum, organised a summer school in Semantics of Computation at Aarhus with BRICS support. There were around 70 participants, mainly PhD students, from all round the world (including Korea, Russia, Argentina, Turkey, and USA). The lecturers and topics were:

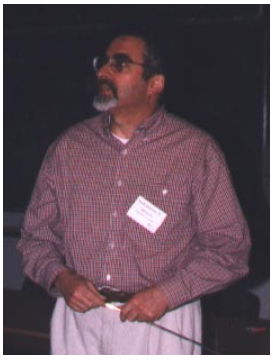


Figure 4: *Bob Tennent* lecturing.

Achim Jung, Birmingham — Domain Theory

Luke Ong, Oxford — Correspondence between Operational and Denotational Semantics

Bob Tennent, Queens, Canada — Denotational Semantics

Jaap van Oosten, Utrecht — Category Theory in Semantics

Andrzej Filinski, Aarhus — Semantics of Types

Each lecturer taught for 5–6 hours using a proportion of the time on exercises. The lecturers took great pains to make their material accessible, which was really appreciated by the students ("they're really teaching" said one student, "not just telling about their recent research"). The summer school was an enjoyable and stimulating experience for students, lecturers and organisers alike.

Figure 5: *Andrzej Filinski* outlining his lecture.

We are pleased to include the following report kindly provided by *Leonor Prensa Nieto*, a PhD student at Technical University of Munich, Germany.

High on a beautiful green campus with a view over the city and sea, the University of Aarhus hosted more than 70 students from about 40 different universities in Europe and America. Over five days we enjoyed what I would describe as the most successful combination of what students could wish for in a summer school. The sun even accompanied us the whole time, something that the locals could hardly believe.

Figure 6: *Luke Ong* devoutly proving that the standard model is not fully abstract for PCF.

The lectures, apart from being scientifically challenging, were an enormous success from a pedagogical point of view. Well aware of presenting at a summer school, the speakers strongly focused their talks towards the students, giving thorough introductions and providing high motivation for the subjects. Despite the difficulty of the topics they managed to give an ordered presentation, stressing the intuition behind and using clarifying examples. Most of the sessions were held in a cozy "chalk and blackboard" style, and the lecturers took care to always follow a solid line of reasoning. Numerous remarks on the relationships with other lectures made the summer school seem quite united.

Figure 7: Chaotic Argentinian tango at the farewell dinner.

During the breaks, lively discussions among students and lecturers were held while drinking coffee with traditional sweet Danish cakes. An excursion to the Museum of Danish Pre-History at Moesgaard distracted us from the semantics of computation and led us into the world of the Vikings, who by the way, were not tall and blonde, but rather small and dark haired. After visiting the museum with its beautiful constructions of Viking houses and churches, we hiked through the woods to an open-air cafe, where more sweet Danish cakes were waiting for us. Always sunny, we ended the excursion walking along a white beach and drinking Danish beer.

Figure 8: *Jaap van Oosten* singing solo at the farewell dinner.

Unforgettable was also the farewell dinner, a delicious typical Danish menu that was followed by the apparently Danish tradition of singing. All participants sang songs in their native language. Harmonic Chinese and Korean, perfectly toned, organised German, sweet Hindi, Italian "bella ciao", chaotic Argentinian tango, Jaap van Oosten's wonderful solo (the interest in category theory increased considerably after his performance) and many more voices concluded with all climbing on the tables to sing to the "Famous Men of Science".

Figure 9: *Achim Jung* and *Glynn Winskel* in deep thoughts.

A sweet souvenir: a broader and richer knowledge of domain and category theory, a better comprehension of denotational semantics as well as its correspondence with operational semantics and a much clearer understanding of the theory of types are, together with valuable friends, what still remains four months after the summer school.

In the name of all participants I would like to thank the organisers, the lecturers and the locals for making the summer school a great success in both the scientific and social aspects. I hope that the next "semantics of computation" event will be held again very soon. ▩

Aalborg Wing Retreat

On May 17–18, 1999, the Aalborg wing of BRICS had a retreat at Rønbjerg Ferie Center. In the new BRICS research plan, the activities in Aalborg were extended to include not only semantics and theory of computation and concurrency but also design and experimental implementation of distributed and real-time systems. The purpose of the retreat was to identify common strategies and future research activities.

The retreat gathered some 22 participants including a small, but important participation from BRICS at Aarhus. Professor *Ole Brun Madsen*, Department of Control Engineering at Aalborg University gave an invited talk on the historical development of the Danish computer network. The sessions covered a variety of research directions including mobility and objects; design, verification and validation; theoretical foundation of computation; and, construction and experimentation in distributed systems. Also, the participants from Aarhus willingly, and on very short notice, shared their current research ideas with us.

Part of the retreat was devoted to discussions on future fund raising and external relationships, as well as educational problems. Finally, the excellent facilities at Rønbjerg Ferie Center gave ample opportunities for all sorts of recreational activities, contributing to the sociable atmosphere at the retreat. ▩

Figure 10: Participants of the Aalborg Wing Retreat.

Randomisation and Approximation Algorithms in Combinatorial Optimisation

May 18 and 20, 1999, *Devdatt Dubhashi*, Indian Institute of Technology, Delhi, India, gave two double lectures on Randomisation and Approximation Algorithms in Combinatorial Optimisation.

Optimal Graph Reduction: Computation, Continuations, Complexity

May 25–28, 1999, *Julia Lawall* and *Harry Mairson*, both Department of Computer Science, Brandeis University, Waltham, Massachusetts, USA, gave four double lectures on Optimal Graph Reduction: Computation, Continuations, Complexity.

Semantics of Objects As Processes, SOAP '99

June 14, 1999, this second informal workshop with selected contributions on *(clean) Semantics for Objects As Processes* was held with the participation of around 30 researchers at the University of Lisbon, Portugal, as a satellite event of ECOOP '99.

The aim of the SOAP workshops is to bring together researchers working mainly in this area, but in related fields as well, where other process models or calculi are used as a basis for the semantics of objects. Similarly, the '99 edition of SOAP was composed of two complementary thematic building blocks.

The first building block addressed the motto "Semantics of Objects As Processes" literally in that objects are represented as a derived concept within a framework of processes; we welcomed Oscar Nierstrasz, Markus Lumpe, and Jean-Guy Schneider as invited speakers to present the work they had been accomplishing together in this area—starting out from a mobile process calculus—and to let us learn about their conclusions. This session was rounded up by a verification approach using a temporal logic as a target setting for, in this case, UML-style objects.

Oscar Nierstrasz — Piccola -A Small Composition Language

Markus Lumpe — The πL-Calculus - A Formal Foundation for Software Composition

Jean-Guy Schneider — Object Models in the πL-Calculus

Günter Graw — Composing object-oriented specifications and verifications with cTLA

The second building block, divided into a session on behavioural subtyping and another one on behavioural typing, is more to be seen as an adaptation of the process-theoretic viewpoint to some object-oriented framework. While the typed λ-calculus is a firm ground to study typing for object-oriented languages, the typing of concurrent objects poses particular problems due to synchronisation constraints. A static notion of typing is not powerful enough to capture dynamic properties of objects' behaviour, like non-uniform service availability. Concurrency theory inspires dynamic notions of typing and subtyping, and the works that constituted this block of SOAP '99 exemplify the research currently being done in the field.

Nabil Hameurlain — Behavioural Types in CoOperative Objects (not presented, but in the proceedings)

Glenn Lewis — A Practical Approach to Behavioural Inheritance in the Context of Coloured Petri Nets

Christof Peter — A Concurrent Object Calculus with Types that Express Sequences

Abdelkrim Nimour — Explicit behavioural typing for object interfaces

The workshop closed with a one-hour informal "round-table" discussion essentially addressing the problems and approaches of behavioural typing and subtyping.

By means of a formal refereeing process, among nine submissions five were selected by the programme committee (Hans Hüttel, Josva Kleist, Uwe Nestmann, and António Ravara) and, except for the one indicated above, also presented at the meeting. We, the local organisers Uwe Nestmann and António Ravara, would like to thank the organisers of ECOOP '99 for helping us logistically to set up the SOAP '99, we thank BRICS, in particular Uffe H. Engberg, for supporting the publication of the proceedings, and we thank Massimo Merro and Silvano Dal-Zilio for their assistance in the refereeing process.

The informal proceedings are available as BRICS Notes [NS-99-2], and also accessible from the workshop site, www.cs.auc.dk/soap99/.

Workshop on Geometric and Topological Methods in Concurrency Theory

June 21–25, 1999, *Lisbeth Fajstrup* and *Martin Raussen* from Department of Mathematical Sciences, Aalborg University held a workshop on Geometric and Topological Methods in Concurrency Theory supported by BRICS. The first two days of the workshop were devoted to tutorials aimed at non-specialists and PhD students. The workshop was directed towards researchers interested in geometric reasoning and modelling dealing with concurrency theory. The talks were one or two hours long. Some 20 participants from Europe, America and India attended the workshop.

Incrementalisation: a Powerful Approach to Efficiency Improvement

June 28 and 29, 1999, *Y. Annie Liu*, Computer Science Department, Indiana University, Bloomington, Indiana, USA, gave two double lectures on Incrementalisation: a Powerful Approach to Efficiency Improvement.

Randomisation and Abstraction: Useful Tools for Optimisation

July 6 and 7, 1999, *Bernd Gärtner*, Institut für Theoretische Informatik, ETH Zürich, Switzerland, gave ten lectures on Randomisation and Abstraction: Useful Tools for Optimisation.

Coding Theory Days

August 18–19, 1999

In recent years applications of coding theory in computer science continue to pop up in very diverse and sometimes quite surprising contexts. This includes for instance secure distributed computing, secret sharing, cryptanalysis, quantum cryptography, probabilistically checkable proofs and derandomisation.

It was therefore natural to organise an event bringing together people from classical coding theory and those interested in the applications, to learn more about new results in coding, and to initiate an interest in the problems arising from the applications.

The coding theory days were organised by Peter Bro Miltersen and Ivan B. Damgård in collaboration with Tom Høholdt from DTU (Technical University of Denmark). We were pleased to find that many good speakers from several institutions wanted to participate.

Olav Geil, Aalborg — Codes and Order Domains

Tom Høholdt, DTU — List Decoding

Peter Bro Miltersen, BRICS — Application of Coding in PCP and Derandomisation

Michael Ben-Or, Hebrew University — Quantum Key Exchange

Agnes Heydtmann, DTU — Decoding Algorithms and Cryptanalysis

Ivan B. Damgård, BRICS — Coding, Secret Sharing and Secure Computation

Johan P. Hansen, Aarhus — Coding Theory and Algebraic Geometry

Louis Salvail, BRICS — Interactive Error Correction

Introduction to Computer Vision

August 19 and 20, 1999, *John Hallam*, Artificial Intelligence, Division of Informatics, The University of Edinburgh, Scotland, gave three double lectures on Introduction to Computer Vision.

Combinatorial Optimisation Meeting

An informal workshop, called Combinatorial Optimisation Meeting, was hosted by BRICS on August 27–31, 1999. It was the first major five-day meeting in this area in Aarhus. The topic, Combinatorial Optimisation, or more broadly Discrete Mathematics, includes graph theory, matroid theory, combinatorial (approximation) algorithms, polyhedral combinatorics, combinatorial geometry, integer programming, and so on. During the past 20–30 years this area has become a major branch of mathematics with an increasing number of applications and with plenty of connections to theoretical computer science.

Eleven researchers accepted our invitation from France, Hungary, Sweden, UK, and the USA as well as from other universities in Denmark. The list of lectures included talks given by these guests and also talks by six speakers, who were visiting or employed by BRICS at that time. Moreover, several PhD students from BRICS (and one from Odense) registered, too, totalling to about 20–25 participants.

Figure 12: *G. Simonyi, Z. Szigeti, R. Ravi* and *T. Jordán* discussing a matching problem.

Figure 11: A group of relieved participants after the very last talk.

The size of the meeting was small enough to have a relaxed and informal atmosphere while the style and quality of talks (most of them up-to-date 45- or 90-minute survey talks in hot topics) made the participants work and collaborate and resulted in a very active week and a successful meeting.

Three topics were in focus: combinatorial or LP-based approximation algorithms, structural results on graphs, and applications of integer programming in solving large real-world problems.

R. Ravi, Carnegie Mellon University, Pittsburgh, USA, summarised the basics of bicriteria approximation algorithms. He was responsible for the official meeting photos, too, see the enclosed pictures for the result. *Alberto Caprara*, BRICS and Bologna, Italy, *Gregory Gutin*, Brunel University, London, United Kingdom, *Alessandro Panconesi*, BRICS and Bologna, Italy, and *Zoltán Szigeti*, Université Paris VI, France, showed how to design approximation algorithms for different graph problems, such as edge-colouring, TSP, minimum-size biconnected subgraphs, and others. *Fabian Chudak*, IBM T. J. Watson Research Center, USA, and *Maxim Sviridenko*, BRICS, talked about close-to-optimal solutions for certain facility location and scheduling problems, respectively. *Jens Lagergren*, Stockholm, Sweden, gave best possible approximation guarantees for the matrix-to-line problem, an important problem in computational biology.

A second group of talks included two survey talks on graph entropy and on the complexity of LPs for combinatorial problems by *Gábor Simonyi*, Hungarian Academy of Sciences, Budapest, and *Devdatt Dubhashi*, BRICS and Göteborg, Sweden, respectively. Coloring and connectivity problems of graphs as well as structural results on directed graphs were discussed in 45-minute talks by *Jørgen Bang-Jensen*, *Bjarne Toft*, both SDU, Odense, and from BRICS *Romeo Rizzi* and *Anders Yeo*. Bjarne mingled new results and historical remarks in his colourful talk. We learnt exciting details about the birth of a celebrated theorem of Petersen, the Danish graph theorist.

We also had three survey talks about real-life applications, based on integer programming methods. *Knut Reinert*, Celera Genomics, Washington, USA, showed how to solve sequence alignment problems efficiently, which is a fundamental problem in computational biology. *Pawel Winter*, DIKU, Copenhagen, presented a method for solving large instances of geometric Steiner tree problems. *Dag Wedelin*, Chalmers University, Göteborg, described the details of an efficient

optimiser package, applied in crew scheduling problems.

The social side of the meeting included two excellent dinners, served in the nearby "meeting hotel", and three lunches at the university. Special thanks to Ann Eg Mølhave and Karen Kjær Møller for the assistance. The Aarhus Festival Week, another major event held parallel to our meeting, took care of the evening attractions. The meeting was organised by Tibor Jordán.

A Taster of Descriptive Set Theory

September 9, 14 and 16, 1999, *Julian Bradfield*, BRICS and LFCS, Edinburgh, gave three double lectures on A Taster of Descriptive Set Theory.

Denotational Semantics of Types

October 14, 22 and 27, November 3, 1999, *John C. Reynolds*, Computer Science Department, Carnegie Mellon University, USA, gave four double lectures on Denotational Semantics of Types.

Logic of Proofs

October 21, 26 and 28, 1999, *Tatiana Yavorskaja (Sidon)*, Moscow State University, Russia, gave three double lectures on Logic of Proofs: Uniform Provability Semantics for the Modality and Lambda-Terms.

Newly Appointed Researchers, Guests and PhDs

Mary Cryan
Mary Cryan has just finished her PhD thesis on *Learning and Approximation Algorithms for problems motivated by Evolutionary Trees* at the University of Warwick. Her broad research interests lie in the design and analysis of algorithms and in complexity theory. She is especially interested in probabilistic analysis, learning theory, and algorithms for computational biology. Mary joined BRICS in September 1999.

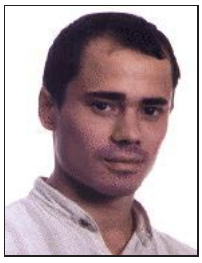
Rolf Fagerberg
Rolf Fagerberg received his PhD from University of Odense, Denmark, in 1997, supervised by Joan Boyar. His research interests are centered around data structures, in particular search trees and priority queues, and I/O-efficient algorithms, but cover algorithmics in general. He held a post doc position at the University of Odense before joining BRICS in February 1999.

Tibor Jordán
Tibor Jordán received his PhD from Eötvös University, Budapest, Hungary in 1995. Since then he has been guest researcher at CWI, Amsterdam, assistant professor at the Technical University of Budapest, and post-doc researcher at the University of Odense. He joined BRICS in February 1999. His research area includes combinatorial optimisation, graph theory, and approximation algorithms.

Ernst-Rüdiger Olderog
Professor Ernst-Rüdiger Olderog from the Theoretical Department of the Faculty of Informatics at the Carl von Ossietzky University of Oldenburg, Germany, is visiting BRICS, Aalborg, from mid October 1999 and the rest of the year. Ernst-Rüdigers interests include Duration Calculus and Hybrid Systems.

Paul Pettersson
Paul Pettersson obtained his PhD from Uppsala University in 1999. His thesis, *Modelling and Verification of Real-Time System Using Timed Automata: Theory and Practice*, describes on-the-fly, symbolic and compositional model-checking techniques for timed system. It also demonstrates the potential applications of the techniques by presenting three industrial-size case studies where the techniques, implemented in the tool UPPAAL, have been applied. Paul joined BRICS in Aalborg in February 1999. He is currently involved in the VHS project, Verification of Hybrid Systems.

Vinodchandran N. Variyam
Vinodchandran N. Variyam received his PhD degree from the Institute of Mathematical Sciences, Chennai, India in February 1999. His thesis titled *Counting Complexity and Computational Group Theory* studies the structural complexity of many important computational problems that arise from group theory. The thesis also introduces a new model for learning in order for a tighter analysis of the complexity of learning some algebraic concept classes. Vinodchandran joined BRICS as a postdoctoral fellow in March 1999 and his main interests include complexity theory and computational learning theory.



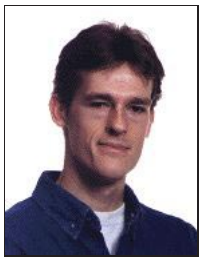
Maxim Sviridenko
Maxim Sviridenko received his PhD from Sobolev Institute of Mathematics, Novosibirsk, Russia in January 1999. His thesis, *Approximation Algorithms for Discrete Facility Location Problems* was devoted to the worst-case analysis of approximation algorithms for some classical location problems. His current research interests include approximation algorithms, hardness of approximation, scheduling theory. Maxim joined BRICS in August 1999.

Tommy Thorn
Tommy Thorn received his PhD from IRISA/INRIA in Rennes, France under the supervision of Daniel Le Métayer. The topic of the thesis is the automatic methods for formal verification of security policies in the context of programming language based security. From February 1999 to September 1999 Tommy worked on aspects of the correctness of the Java Card language using the automatic proof assistant Coq. His interests include the application of formal methods to practical problems, notably computer security. He joined BRICS in October 1999.

David Toman
David Toman is visiting BRICS in the autumn '99 semester and gives an introductory course on *Databases*. He received his MS (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 Assistant Professor at the Department of Computer Science, Faculty of Mathematics, University of Waterloo, 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 programming paradigms), and programming languages.

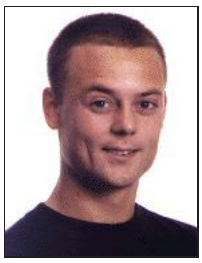
Igor Walukiewicz
Igor Walukiewicz received his PhD from University of Warsaw, Poland in 1994. His thesis *A Complete Deductive System for the Mu-Calculus* presented a finitary complete axiomatisation of the Mu-Calculus. From 1994 to 1996 he held a post-doc position at

BRICS. Since 1996 he is a lecturer at Warsaw University. His research interests concentrate around logics with applications to computer science. In particular he is interested in monadic second order logic, program logics, automata theory and finite model theory. The motivation for his research comes mostly from problems in verification and concurrency. He is visiting BRICS from September 1999 to January 2000.



Anders Yeo
Anders Yeo is an Australian citizen, who, apart from short periods, has lived in Denmark since the age of five. His main hobby is squash, where he currently represents Denmark on the national team. He received his PhD from the University of Odense in 1998, supervised by Jørgen Bang-Jensen and Gregory Gutin. His thesis, *Semicomplete Multipartite Digraphs*, mainly considers cycles in Semicomplete Multipartite Digraphs, but also contains results from a wide area of graph theory and algorithmics. From March 1998 to August 1998 he was employed as a research assistant at Odense University. From September 1998 until August 1999, he held a post-doc position at the University of Victoria, Canada, which was financed by the Danish research council. He joined BRICS in September 1999.

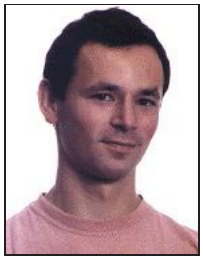
BRICS is also happy to welcome the following newly admitted PhD students.



Mikkel Nygaard Hansen
Mikkel Nygaard is a newly accepted PhD student under the supervision of Glynn Winskel. In his four years as student in Aarhus, he has come to take an interest in theoretical computer science, in particular semantics of programming languages. Mikkel is now concerned with applications of category theory in concurrency.

Dan Hernest
Dan Hernest graduated in July 1998 from the Faculty of Mathematics of the University of Bucharest, Rumania with a thesis on Probabilistic Model Theory, written under the guidance of Prof. George Georgescu. His former area of interest was many valued logics. Dan hasn't settled on a specific main area of research yet but he is mainly concerned with logic in Computer Science. His preliminary supervisor is Olivier Danvy.

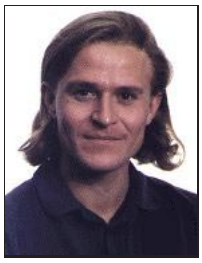
Mads Johan Jurik
Mads Johan Jurik is a newly accepted PhD student after 4 years of study at the Department of Computer Science at the University of Aarhus. His main area of interest is cryptology, with a special interest in Electronic payment systems, under the supervision of Ivan B. Damgård. Other areas of interest include algorithms and compilers.



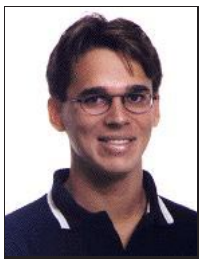
Maciej Koprowski
Maciej Koprowski graduated from the Nicholas Copernicus University in Torun, Poland. His MSc thesis concerned algorithms for elliptic curves over finite fields. His main areas of interest are cryptology and algorithmics. His preliminary supervisor is Ivan B. Damgård.

Giuseppe Milicia
Giuseppe Milicia got his MSc (Laurea) in Computer Science in July 1998 from the University of Pisa, Italy. He started working on his MSc thesis under the supervision of Dr. Vladimiro Sassone and Dr. Gianluigi Ferrari while staying at the Queen Mary and Westfield College, London, United King-

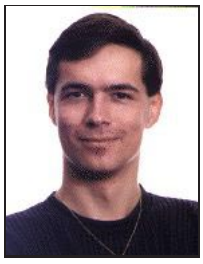
dom. He is a PhD student at BRICS since January 1999. His interests include process calculi, mobile agents, object orientation and type systems. His preliminary supervisor is Mogens Nielsen.



Jesper Buus Nielsen
Jesper Buus Nielsen became a PhD student at BRICS in August 1999 after five year of study in Aarhus. His primary fields of interest are public-key cryptography and cryptologic protocol theory, but include complexity theory, algebra, and AI. Initially he will be concentrating his work on deniable-encryption protocols and other types of encryption which loosely could be characterised using the term non-committing encryption. These are objects which are interesting in themselves, but especially as components in cryptologic protocols.



Paulo Oliva
Paulo Oliva graduated in February 1999 from the Faculty of Computer Science at the Federal University of Pernambuco, Brazil. During his undergraduate course he worked as a research assistant in the fields of approximative pattern matching, information retrieval and automatic placement of integrated circuits. His preliminary supervisor is Peter Bro Miltersen.



Jiří Srba
Jiří Srba graduated in July 1998 from the Faculty of Informatics at the Masaryk University in Brno, Czech Republic. He started the first year of his PhD study in Brno under the guidance of Ivana Černá and Mojmír Křetínský. Since August 1999 he is a new PhD student at BRICS. His research interests are within concurrency theory, in particular process algebra, focusing on decidability/complexity questions. His preliminary supervisor is Mogens Nielsen.

Daniele Varacca
Daniele Varacca graduated in Mathematics in July 1998 from the University of Parma, Italy. His MSc thesis, written under the direction of Marco Forti, University of Pisa, Italy, concerns the mathematical foundations of category theory. Daniele has not settled on a specific field of research yet, but he is mainly concerned with logic, game theory, category theory, semantics. His preliminary supervisor is Glynn Winskel.

Dissertation Abstracts

Categorical Models for Concurrency

Independence, fairness and dataflow

by Thomas Troels Hildebrandt

A formal semantics gives meaning to computational systems by describing their behaviour in a *mathematical model*. This thesis is concerned with formal semantics and models for *concurrent* computational systems, i.e. systems such as e.g. telephone networks or traffic control systems, consisting of a number of parallel computing sequential systems possibly distributed between different locations. For concurrent systems the interesting aspect of their computation is often how they *interact* with the environment during a computation and not in which state they terminate, indeed they may not be intended to terminate at all. For this reason they are often referred to as *reactive systems*, to distinguish them from traditional *calculational* systems, as e.g. a program calculating your income tax, for which the interesting behaviour is the answer it gives when

(or if) it terminates, in other words the (possibly partial) function it computes between input and output. *Church's thesis* tells us that regardless of whether we choose the *lambda calculus*, *Turing machines*, or almost any modern programming language such as *C* or *Java* to describe calculational systems, we are able to describe exactly the same class of functions. However, there is no agreement on observable behaviour for concurrent reactive systems, and consequently there is no correspondent to Church's thesis. A consequence of this fact is that an overwhelming number of different and often competing notions of observable behaviours, primitive operations, languages and mathematical models for describing their semantics, have been proposed in the literature on concurrency.

The work presented in this thesis focuses on three main topics in concurrency theory: *independence*, *fairness* and *non-deterministic dataflow*. These are treated within a categorical approach to semantics for concurrency which has been developed through the last 15 years, aiming at a more coherent theory. The initial stage of this approach is reported on in the chapter on models for concurrency by Winskel and Nielsen in the Handbook of Logic in Computer Science, and focuses on relating the already existing models and techniques for reasoning about them. This work was followed by a uniform characterisation of behavioural equivalences from the notion of *bisimulation from open maps* proposed by Joyal, Winskel and Nielsen, which was applicable to any of the categorical models and shown to capture a large number of existing variations on bisimulation. At the same time, a class of abstract models for concurrency was proposed, the *presheaf models for concurrency*, which have been developed over the last 5 years, culminating in the recent thesis of Cattani.

Our work on independence, concerned with explicitly representing that actions result from independent parts of a system, falls in two parts. The first contributes to the initial work on describing formal relationships between existing models. The second contributes to the study of concrete instances of the abstract notion of bisimulation from open maps. In more detail, the first part gives a complete formal characterisation of the relationship between two models, *transition systems with independence* and *labelled asynchronous transition systems* respectively, which somehow escaped previous treatments. The second part studies the borderline between two well known notions of bisimulation for independence models such as 1-safe Petri nets and the two models mentioned above. The first is known as the *history-preserving bisimulation* (HPB), the second is the bisimulation obtained from the open maps approach, originally introduced under the name *hereditary history-preserving bisimulation* (HHPB) as a strengthening of HPB obtained by adding *backtracking*. Both bisimulations have the property of being a congruence with respect to *action refinement*, however where the first in addition has been proven to be *decidable* for finite state systems, the latter has recently been shown to be *undecidable* for finite state systems. We consider history-preserving bisimulation with *bounded backtracking*, and show that it gives rise to an infinite, *strict* hierarchy of *decidable* history-preserving bisimulations separating HPB and HHPB.

The work on fairness and non-deterministic dataflow takes its starting point in the work on presheaf models for concurrency in which these two aspects have not been treated previously.

Fairness is concerned with ruling out some completed, possibly infinite, computations as *unfair*. Our approach is to give a presheaf model for the observation of *infinite* as well as finite computations. This includes a novel use of a *Grothendieck topology* to capture unique completions of infinite computations. The presheaf model is given a concrete representation as a category of *generalised synchronisation trees*, which we formally relate to the general transition systems proposed by Hennessy and Stirling for the study of fairness. In particular the bisimulation obtained from open maps is shown to coincide with their notion of *extended bisimulation*. Benefitting from the general results on presheaf models we give the first denotational semantics of Milner's SCCS with finite delay that coincides with his opera-

tional semantics up to extended bisimulation.

The work on non-deterministic dataflow, i.e. systems communicating asynchronously via buffered channels, recasts dataflow in a modern categorical light using *profunctors* as a generalisation of relations. The well known causal anomalies associated with relational semantics of indeterminate dataflow are avoided, but still we preserve much of the intuitions of a relational model. The model extends the traditional presheaf models usually applied to semantics for *synchronous* communicating processes described by CCS-like calculi, and thus opens up for the possibility of combining these two paradigms. We give a concrete representation of our model as a category of (unfolded) *monotone port automata*, previously studied by Panangaden and Stark. We show that the notion of bisimulation obtained from open maps is a congruence with respect to the operations of dataflow. Finally, we use the categorical formulation of feedback using *traced monoidal categories*. A pay off is that we get a semantics of higher-order networks almost for free, using the *geometry of interaction* construction.

Abstraction-Based Verification of Distributed Systems

by Henrik Ejersbo Jensen

This thesis presents abstraction-based proof methods and practical abstraction strategies to support the integration of theorem proving and model checking methods in verification of distributed systems. The thesis is in two parts. In the first part we present abstraction frameworks for untimed systems described as I/O automata and for real-time systems described as timed automata. The frameworks provide general conditions for preservation of properties from concrete systems to abstract ones. For the I/O automaton model we present preservation conditions for safety and liveness properties stated over actions as well as over states. The preservation conditions are based on simulation relations. The abstraction theory is formalised using the Larch theorem prover and a scheme for translating I/O automata in to the SPIN model checker is examined. For the timed automaton model we provide preservation conditions based on requirements stated as automaton specifications with a satisfaction relation in the form of a timed ready simulation relation. Our preservation conditions are based on an action parameterised variant of this simulation relation. The timed abstraction framework is stated in the input language of the UPPAAL model checker for real-time systems providing a close link to automatic verification. In the second part of this thesis we provide abstraction-based proofs for three nontrivial distributed algorithms all parameterised in the number of processes: Burns' Mutual Exclusion algorithm, The Bounded Concurrent Timestamp System (BCTSS) algorithm, and Fischer's Real-Time Mutual Exclusion algorithm. The proof of Burns' algorithm utilises an abstraction strategy based on skolemisation and the proof is carried out by support from the Larch Prover and the SPIN model checker. The proof of the BCTSS algorithm is the most advanced in this thesis. The BCTSS algorithm is one of the most complicated algorithms in the distributed systems literature and existing proofs are all long and hard to understand. Our abstraction proof exploits a combination of induction and abstraction strategies to delegate major proof tasks to automatic verification in the SPIN model checker. The proof of Fischer's algorithm utilises a combination of compositionality and abstraction strategies based on network invariants. The UPPAAL model checker is used to verify the constructed abstraction.

Compositional Verification of Concurrent Systems

A possible cure for the state explosion problem

by Kåre Jelling Kristoffersen

In the development of embedded software and hardware, verification of intermediate designs has established itself as a serious supplement to the activity of testing the final product. During

the last decade several research efforts have been made in developing automatic tools for carrying out this verification. In particular, tools with a firm theoretical foundation and efficient algorithmic realization have been sought.

In general, for a concurrent system, the size of the state space grows exponentially with the number of components in the system. Thus, naive verification algorithms which simply traverse the complete state space using an explicit state-representation will scale very poorly. This phenomenon is commonly referred to as the State Explosion Problem, and is a consequence of the provable theoretical intractability of verification. In order to overcome this problem (without solving P=PSPACE) tool developers are challenged with the problem of finding heuristics allowing for efficient representation and manipulation of state spaces in practical situations.

This thesis presents a collection of algorithmic techniques and tools that significantly improve the current status of verification of designs in development of embedded software and hardware for discrete state systems and real time systems. In particular, the State Explosion Problem is dealt with using a blend of compositional techniques, exploiting the structure and dependencies in concurrent systems, together with space efficient symbolic representations for sets of states.

Relational Reasoning about Functions and Nondeterminism

by Søren B. Lassen

This dissertation explores a uniform, relational proof style for operational arguments about program equivalences. It improves and facilitates many previously given proofs, and it is used to establish new proof rules for reasoning about term contexts, recursion, and nondeterminism in higher-order programming languages.

Part I of the thesis develops an algebra of relations on terms and exploits these relations in operational arguments about contextual equivalence for a typed deterministic functional language. Novel proofs of the basic laws, sequentiality and continuity properties, induction rules, and the CIU Theorem are presented together with new relational proof rules akin to Sangiorgi's "bisimulation up to context" for process calculi.

Part II extends the results from the first part to nondeterministic functional programs. May and must operational semantics and contextual equivalences are defined and their properties are explored by means of relational techniques. For must contextual equivalence, the failure of ordinary syntactic ω-continuity in the presence of countable nondeterminism is addressed by a novel transfinite syntactic continuity principle. The relational techniques are also applied to the study of lower and upper applicative simulation relations, yielding new results about their properties in the presence of countable and fair nondeterminism, and about their relationship with the contextual equivalences.

See [DS-98-2].

Computational Biology

by Rune Bang Lyngsø

All living organisms are based on genetic material, genetic material that is specific for the organism. This material – that inexplicably can be regarded as sequences – can thus be a valuable source of information, both concerning the basic processes of life and the relationships among different species. During the last twenty years the ability to 'read' this genetic material has increased tremendously. This development has led to an explosion in available data.

But all this data is of little use if methods are not available for deducing information from it. Simply by the sheer amount of data, it is of vital importance that these methods are automated to a very large degree. This demand has spawned the field of *computational biology*, a field that from a computer science point of view offers novel problems as well as variants over known prob-

lems.

In this dissertation we focus on problems directly related to the biological sequences. That is, problems concerned with

- detecting patterns in one sequence,
- comparing two or more sequences,
- inferring structural information about the biomolecule represented by a given sequence.

We discuss the modelling aspects involved when solving real world problems, and look at several models from various areas of *computational biology*. This includes examining the complexity of and developing efficient algorithms for some selected problems in these models.

The dissertation includes five articles, four of which have been previously published, on

- finding repetitions in a sequence with restrictions on the distance between the repetitions in the sequence;
- comparing two coding DNA sequences, that is, a comparison of DNA sequences where the encoded protein is taken into account;
- comparing two hidden Markov models. Hidden Markov models are often used as representatives of families of evolutionary or functionally related sequences;
- inferring the secondary structure of an RNA sequence, that is, the set of base pairs in the three dimensional structure, based on a widely accepted energy model;
- an approximation algorithm for predicting protein structure in a simple lattice model.

These articles contain the technical details of the algorithms discussed in the first part of the dissertation.

Complexity of Data Structures

by Theis Rauhe

This thesis consists of a number of results providing various complexity results for a number of dynamic data structures and problems. A large part of the result is devoted techniques for proving lower bounds in Yao's cell probe model. In addition we also provide upper bound results for a number of data structure problems.

First we study the signed prefix sum problem: given a string of length n of 0s and signed 1s compute the sum of its ith prefix during updates. We show a lower bound of $\Omega(\log n/\log\log n)$ time per operation, even if the prefix sums are bounded by $\log n/\log\log n$ during all updates. We show how these results allow us to prove lower bounds for a variety of dynamic problems. We give a lower bound for the dynamic planar point location in monotone subdivisions of $\Omega(\log n/\log\log n)$ per operation. We give a lower bound for dynamic transitive closure on upward planar graphs with one source and one sink of $\Omega(\log n/(\log\log n)^2)$ per operation. We give a lower bound of $\Omega(\sqrt{(\log n/\log\log n)})$ for the dynamic membership problem of any Dyck language with two or more letters.

Next we introduce new models for dynamic computation based on the cell probe model. We give these models access to nondeterministic queries or the right answer ± 1 as an oracle. We prove that for the dynamic partial sum problem, these new powers do not help, the problem retains its lower bound of $\Omega(\log n/\log\log n)$. From these results we obtain stronger lower bounds of order $\Omega(\log n/\log\log n)$ for the conventional Random Access Machine and cell probe model of the above problems for upward planar graphs and dynamic membership for Dyck languages. In addition we also characterise the complexity of the dynamic problem of maintaining a symmetric function for prefixes of a string of bits. Common to these lower bounds are their use of the chronogram method introduced in a seminal paper of Fredman and Saks.

Next we introduce a new lower bound technique that differs from the mentioned chronogram method. This method enable us to provide lower bounds for another fundamental dynamic problem we call the marked ancestor problem; consider a rooted tree whose nodes can be in two states: marked or unmarked. The marked ancestor problem is to maintain a data structure with the following operations: $mark(v)$ marks node v; $unmark(v)$ removes any marks from node v; $firstmarked(v)$ returns the first marked node on the path from v to the root. We show tight upper and lower bounds for the marked ancestor problem. The upper bounds hold on the unit-cost Random Access Machine, and the lower bounds in cell probe model. As corollaries to the lower bound we prove (often optimal) lower bounds on a number of problems. These include planar range searching, including the existential or emptiness problem, priority search trees, static tree union-find, and several problems from dynamic computational geometry, including segment intersection, interval maintenance, and ray shooting in the plane. Our upper bounds improve algorithms from various fields, including dynamic dictionary matching, coloured ancestor problems, and maintenance of balanced parentheses.

We study the fundamental problem of sorting in a sequential model of computation and in particular consider the time-space trade-off (product of time and space) for this problem. Beame has shown a lower bound of $\Omega(n^2)$ for this product leaving a gap of a logarithmic factor up to the previously best known upper bound of $O(n^2 \log n)$ due to Frederickson. We present a comparison based sorting algorithm which closes this gap. The time-space product $O(n^2)$ upper bound holds for the full range of space bounds between $\log n$ and $n/\log n$.

The last problem we consider is dynamic pattern matching. Pattern matching is the problem of finding all occurrences of a pattern in a text. In a dynamic setting the problem is to support pattern matching in a text which can be manipulated on-line, *i.e.*, the usual situation in text editing. Previous solutions support moving a block from one place to another in the text in time linear to the block size where efficient pattern matching is supported too. We present a data structure that supports insertions and deletions of characters and movements of arbitrary large blocks within a text in $O(\log^2 n \log\log n \log^* n)$ time per operation. Furthermore a search for a pattern P in the text is supported in time $O(\log n \log\log n + occ + |P|)$, where occ is the number of occurrences to be reported. An ingredient in our solution to the above main result is a data structure for the *dynamic string equality* problem due to Mehlhorn, Sundar and Uhrig. As a secondary result we give almost quadratic better time bounds for this problem which in addition to keeping polylogarithmic factors low for our main result, also improves the complexity for several other problems. ▪

Algorithms in Computational Biology

by Christian Nørgaard Storm Pedersen

In the thesis we are concerned with constructing algorithms that address problems with biological relevance. This activity is part of a broader interdisciplinary area called computational biology, or bioinformatics, that focus on utilising the capacities of computers to gain knowledge from biological data. Most problems in computational biology are related to molecular or evolutionary biology and focus on analysing and comparing the genetic material of organisms. The genetic material of an organism is the blueprint of the molecules the organism needs for the complex task of living. A deciding factor in shaping the area of computational biology is the fact that the biomolecules DNA, RNA and protein which are responsible for storing and utilising the genetic material of an organism can be described as sequences of finitely many different residues, that is, can be described as strings over finite alphabets. For example, a DNA sequence can be described as a string over the alphabet $\{\mathrm{A}, \mathrm{G}, \mathrm{C}, \mathrm{T}\}$, where each character represents one of the four possible nucleotides that bond together to form the DNA molecule. Since the genetic material of

an organism is encoded in DNA sequences and reflected in RNA and protein sequences, this representation of biomolecules allows a wide range of algorithmic techniques concerned with strings to be applied for analysing and comparing biological data.

The work of constructing algorithms that address problems with biological relevance, that is, the work of constructing algorithms in computational biology, consists of two interacting steps. The first step is to pose a biologically interesting question and to construct a *model* of the biological reality that makes it possible to formulate the question as a *computational problem*. The second step is to construct an *algorithm* that solves the formulated computational problem. The second step is algorithmic work where the quality of the constructed algorithm is measured according to standard algorithmic methodology in terms of the resources, most prominently time and space, which are required by the algorithm to solve the problem. However, since the problem solved by the constructed algorithm originates from a question with biological relevance, its quality can also be judged by the biological relevance of the answers it produces. These two aspects of quality implies that the work to construct a good algorithm is an interdisciplinary activity that often involves interchanging between modelling the biological reality and reconstructing the algorithm, until a reasonable balance between the resource assumption of the constructed algorithm and the biological relevance of the answers it produces is achieved. Our contributions in the thesis to the field of computational biology are algorithms that address problems within biological sequence analysis and structure prediction.

The genetic material of organisms evolve by discrete mutations, most prominently substitutions, insertions and deletions of nucleotides. Since the genetic material is stored in DNA sequences and reflected in RNA and protein sequences, it makes sense to compare two or more biological sequences in order to look for similarities and differences that can be used to infer knowledge about the relatedness of the sequences and to reconstruct part of their common evolutionary history. It is widely believed that the evolution of genetic material follows the path of least resistance. This is called the parsimony principle and makes it natural to formulate the computational problem of comparing two biological sequences as the problem of computing a minimum cost sequence of events that transforms the one sequence into the other sequence. It is a question of modelling to decide which events to allow and how to assign costs to the allowed events. When the allowed events are limited to substitution, insertion and deletion of residues, the sequence comparison problem is often formulated as an alignment problem and solved efficiently by dynamic programming. In the thesis we consider the problem of comparing two coding DNA sequences when taking the relationship between DNA and proteins into account. This is done by using a cost model that penalises an event on the DNA according to the change it induces on the encoded protein. We analyse the model in details and improve on an earlier algorithm for the alignment problem in the model by reducing its running time by a quadratic factor. This results in an alignment algorithm with an asymptotic running time that is similar to the asymptotic running time of alignment algorithms in much simpler models.

If a family of related biological sequences is available it is natural to derive a compact characterisation of the sequence family. Such a characterisation can for example be used to search for unknown members of the sequence family. A hidden Markov model is a generative model that describes a probability distribution over a set of strings. A widely used way to describe the characteristics of a sequence family is to construct a hidden Markov model that generates members of the sequence family with high probability and non-members with low probability. In the thesis we consider the general problem of comparing hidden Markov models. We define novel measures between hidden Markov models and show how to compute them efficiently using dynamic programming. Since hidden Markov models are widely used to characterise biological sequence families, the measures and methods we present

for comparing hidden Markov models have applications to comparison of biological sequence families.

Besides comparing sequences and sequence families we also consider problems of finding regularities in a single sequence. Finding regularities in a biological sequence can be used to reconstruct part of the evolutionary history of the sequence or to identify the sequence among other sequences. The last application plays a role in genetic fingerprinting. In the thesis we focus on general string problems that can be motivated by biological applications because biological sequences are strings. We present an algorithm which finds all maximal pairs of equal substrings in a string, where each pair of equal substrings adheres to restrictions on the number of characters between the occurrences of the two substrings in the string. This is a generalisation of finding tandem repeats and the running time of the algorithm is comparable to the running time of existing algorithms for finding tandem repeats. The algorithm is based on a general technique that combines a traversal of a suffix tree with efficient merging of search trees. We use the same general technique as the basis for an algorithm that finds all maximal quasiperiodic substrings in a string. A quasiperiodic substring is a substring which can be described as concatenations and superpositions of a shorter substring. The algorithm we present for finding maximal quasiperiodic substrings has a running time that is a logarithmic factor better than the running time of the existing best algorithm for the problem.

Analysing and comparing biomolecules as strings can reveal a lot of information and is the foundation of most algorithms in computational biology. It is however widely believed that the functionality of a biomolecule is primarily determined by its three-dimensional structure. Knowledge about the three-dimensional structure of biomolecules thus often supplies important information that, for example, can be used to design new biomolecules with a specific functionality, or to improve on the comparison of biomolecules by revealing similarities that are not visible in the string representation of the biomolecules. Since it is a difficult and time consuming task to determine the three-dimensional structure of a biomolecule experimentally, methods for computational prediction of the structure of biomolecules are in demand. Constructing such methods is a difficult problem that involves a lot of modelling and often results in the formulation of intractable computational problems. In the thesis we present an algorithm that improves on the widely used mfold algorithm for RNA secondary structure prediction by allowing a less restrictive model of structure formation without an increase in the asymptotic running time. We also present an analysis of the protein folding problem in a simple two-dimensional lattice model called the hydrophobic-hydrophilic model. This analysis shows that several complicated folding algorithms do not produce better foldings in the worst case, in terms of free energy, than an existing simple folding algorithm.

Programming Languages: Design, Analysis, and Semantics

by Anders Bækgaard Sandholm

The thesis consists of three parts. The first part presents contributions in the fields of domain-specific language design, runtime system design, and static program analysis, the second part presents contributions in the field of control synthesis, and finally the third part presents contributions in the field of denotational semantics.

Domain-specific language design is an emerging trend in software engineering. The idea is to express the analysis of a particular problem domain through the design of a programming language tailored to solving problems within that domain. There are many established examples of domain-specific languages, such as LaTeX, flex, and yacc. The novelty is to use such techniques for solving traditional software engineering tasks, where one would previously construct some collection of libraries within a general-purpose language. In Part I, we start

by presenting the design of a domain-specific language for programming Web services. The various domain-specific aspects of Web service programming such as Web server runtime system design, document handling, and concurrency control, are treated.

Having treated many of the Web related topics in some detail, we turn to the particular ideas and central design issues involved in the development of our server-side runtime system. We build a runtime system that not only shares the advantage of traditional CGI-based services, namely that of portability, we also overcome many of the known disadvantages, such as intricate and tedious programming and poor performance.

We then turn to the use of dynamic Web documents, which in our language rely heavily on the use of static program analysis. Static program analysis allows one to offer static compile-time techniques for predicting safe and computable approximations to the set of values or behaviour arising dynamically at runtime when executing a program on a computer. Some of the traditional applications of program analysis are avoidance of redundant and superfluous computations. Another aspect is that of program validation, where one provides guarantees that the analysed code can safely be executed. We present an example of such an analysis as explained in the following. Many interactive Web services use the CGI interface for communication with clients. They will dynamically create HTML documents that are presented to the client who then resumes the interaction by submitting data through incorporated form fields. This protocol is difficult to statically type-check if the dynamic document is created by arbitrary script code using “printf” statements. Previous proposals have suggested using static document templates which trade flexibility for safety. We propose a notion of typed, higher-order templates that simultaneously achieve flexibility and safety. Our type system is based on a flow analysis of which we prove soundness. We also present an efficient runtime implementation that respects the semantics of only well-typed programs.

Having dealt with dynamic Web documents, we turn to the topic of control synthesis. In the design of safety critical systems it is of great importance that certain guarantees can be made about the system. We describe in Part II of this thesis a method for synthesising controllers from logical specifications. First, we consider the development of the BDD-based tool Mona for translating logical formulae into automata and give examples of its use in small applications. Then we consider the use of Mona for control synthesis, that is, we turn to the use of Mona as a controller generator. First, in the context of our domain specific language for generating Web services, where concurrency control aspects are treated using our control synthesis technique. Second, in the context of LEGO robots, where we show how controllers for autonomous LEGO robots are generated using our technique and actually implemented on the physical LEGO brick.

Finally, in Part III we consider the problem of sequentiality and full abstraction in denotational semantics. The problem of finding an abstract description of sequential functional computation has been one of the most enduring problems of semantics. The problem dates from a seminal paper of Plotkin, who pointed out that certain elements in Scott models are not definable. In Plotkin’s example, the ”parallel or” function cannot be programmed in PCF, a purely functional, sequential, call-by-name language. Moreover, the function causes two terms to be distinct denotationally even though the terms cannot be distinguished by any program. The problem of modelling sequentiality is enduring because it is robust. For instance, changing the reduction strategy from call-by-name to call-by-value makes no difference. When examples like parallel or do not exist, the denotational model is said to be fully abstract. More precisely, full abstraction requires an operational definition of equivalence (interchangeability in all programs) to match operational equivalence. It has been proved that there is exactly one fully abstract model of PCF. Until recently, all descriptions of this fully abstract model have used operational

semantics. New constructions using logical relations have yielded a more abstract understanding of Milner's model.

We consider in Part III how to adapt and extend the logical relations model for PCF to a call-by-value setting. More precisely, we construct a fully abstract model for the call-by-value, purely functional, sequential language FPC.

New in the BRICS Report Series, 1998–99

ISSN 0909-0878

34 Flemming Friche Rodler. *Wavelet Based 3D Compression for Very Large Volume Data Supporting Fast Random Access*. October 1999. 36 pp.

33 Luca Aceto, Zoltán Ésik, and Anna Ingólfsdóttir. *The Max-Plus Algebra of the Natural Numbers has no Finite Equational Basis*. October 1999. 25 pp. To appear in *Theoretical Computer Science*.

32 Luca Aceto and François Laroussinie. *Is your Model Checker on Time? — On the Complexity of Model Checking for Timed Modal Logics*. October 1999. 11 pp. Appears in Kutyłowski, Pacholski and Wierzbicki, editors, *Mathematical Foundations of Computer Science: 24th International Symposium*, MFCS '99 Proceedings, LNCS 1672, 1999, pages 125–136.

31 Ulrich Kohlenbach. *Foundational and Mathematical Uses of Higher Types*. September 1999. 34 pp.

30 Luca Aceto, Willem Jan Fokkink, and Chris Verhoef. *Structural Operational Semantics*. September 1999. 128 pp. To appear in Bergstra, Ponse and Smolka, editors, *Handbook of Process Algebra*, 1999.

29 Søren Riis. *A Complexity Gap for Tree-Resolution*. September 1999. 33 pp.

28 Thomas Troels Hildebrandt. *A Fully Abstract Presheaf Semantics of SCCS with Finite Delay*. September 1999. 37 pp. To appear in *Category Theory and Computer Science: 8th International Conference*, CTCS '99 Proceedings, ENTCS, 1999.

27 Olivier Danvy and Ulrik P. Schultz. *Lambda-Dropping: Transforming Recursive Equations into Programs with Block Structure*. September 1999. 57 pp. To appear in the November 2000 issue of *Theoretical Computer Science*. This revised report supersedes the earlier BRICS report RS-98-54.

26 Jesper G. Henriksen. *An Expressive Extension of TLC*. September 1999. 20 pp. To appear in Thiagarajan and Yap, editors, *Fifth Asian Computing Science Conference*, ASIAN '99 Proceedings, LNCS, 1999.

25 Gerth Stølting Brodal and Christian N. S. Pedersen. *Finding Maximal Quasiperiodicities in Strings*. September 1999. 20 pp.

24 Luca Aceto, Willem Jan Fokkink, and Chris Verhoef. *Conservative Extension in Structural Operational Semantics*. September 1999. 23 pp. To appear in the *Bulletin of the EATCS*.

23 Olivier Danvy, Belmina Dzafic, and Frank Pfenning. *On proving syntactic properties of CPS programs*. August 1999. 14 pp. To appear in Gordon and Pitts, editors, *3rd Workshop on Higher Order Operational Techniques in Semantics*, HOOTS '99 Proceedings, ENTCS, 1999.

22 Luca Aceto, Zoltán Ésik, and Anna Ingólfsdóttir. *On the Two-Variable Fragment of the Equational Theory of the Max-Sum Algebra of the Natural Numbers*. August 1999. 22 pp.

21 Olivier Danvy. *An Extensional Characterization of Lambda-Lifting and Lambda-Dropping*. August 1999. 13 pp. Extended version of an article to appear in *Fourth Fuji International Symposium on Functional and Logic Programming*, FLOPS '99 Proceedings (Tsukuba, Japan, November 11–13, 1999). This report supersedes the earlier BRICS report RS-98-2.

20 Ulrich Kohlenbach. *A Note on Spector's Quantifier-Free Rule of Extensionality*. August 1999. 5 pp. To appear in *Archive for Mathematical Logic*.

19 Marcin Jurdziński and Mogens Nielsen. *Hereditary History Preserving Bisimilarity is Undecidable*. June 1999. 18 pp.

18 M. Oliver Möller and Harald Rueß. *Solving Bit-Vector Equations of Fixed and Non-Fixed Size*. June 1999. 18 pp. Revised version of an article appearing under the title *Solving Bit-Vector Equations* in Gopalakrishnan and Windley, editors, *Formal Methods in Computer-Aided Design: Second International Conference*, FMCAD '98 Proceedings, LNCS 1522, 1998, pages 36–48.

17 Andrzej Filinski. *A Semantic Account of Type-Directed Partial Evaluation*. June 1999. To appear in Nadathur, editor, *International Conference on Principles and Practice of Declarative Programming*, PPDP99 '99 Proceedings, LNCS, 1999.

16 Rune B. Lyngsø and Christian N. S. Pedersen. *Protein Folding in the 2D HP Model*. June 1999. 15 pp.

15 Rune B. Lyngsø, Michael Zuker, and Christian N. S. Pedersen. *An Improved Algorithm for RNA Secondary Structure Prediction*. May 1999. 24 pp. An alloy of two articles appearing in Istrail, Pevzner and Waterman, editors, *Third Annual International Conference on Computational Molecular Biology*, RECOMB 99 Proceedings, 1999, pages 260–267, and *Bioinformatics*, 15, 1999.

14 Marcelo P. Fiore, Gian Luca Cattani, and Glynn Winskel. *Weak Bisimulation and Open Maps*. May 1999. To appear in Longo, editor, *Fourteenth Annual IEEE Symposium on Logic in Computer Science*, LICS '99 Proceedings, 1999.

13 Rasmus Pagh. *Hash and Displace: Efficient Evaluation of Minimal Perfect Hash Functions*. May 1999. 11 pp. A short version to appear in *Algorithms and Data Structures: 6th International Workshop*, WADS '99 Proceedings, LNCS, 1999.

12 Gerth Stølting Brodal, Rune B. Lyngsø, Christian N. S. Pedersen, and Jens Stoye. *Finding Maximal Pairs with Bounded Gap*. April 1999. 31 pp. To appear in *Combinatorial Pattern Matching: 10th Annual Symposium*, CPM '99 Proceedings, LNCS, 1999.

11 Ulrich Kohlenbach. *On the Uniform Weak König's Lemma*. March 1999. 13 pp.

10 Jon G. Riecke and Anders B. Sandholm. *A Relational Account of Call-by-Value Sequentiality*. March 1999. 51 pp. To appear in *Information and Computation*, LICS '97 Special Issue. Extended version of an article appearing in *Twelfth Annual IEEE Symposium on Logic in Computer Science*, LICS '97 Proceedings, 1997, pages 258–267. This report supersedes the earlier BRICS report RS-97-41.

9 Claus Brabrand, Anders Møller, Anders B. Sandholm, and Michael I. Schwartzbach. *A Runtime System for Interactive Web Services*. March 1999. 21 pp. Appears in Mendelzon, editor, *Eighth International World Wide Web Conference*, WWW8 Proceedings, 1999, pages 313–323 and *Computer Networks*, 31:1391–1401, 1999.

8 Klaus Havelund, Kim G. Larsen, and Arne Skou. *Formal Verification of a Power Controller Using the Real-Time Model Checker* UPPAAL. March 1999. 23 pp. To appear in Katoen, editor, *5th International AMAST Workshop on Real-Time and Probabilistic Systems*, ARTS '99 Proceedings, LNCS, 1999.

7 Glynn Winskel. *Event Structures as Presheaves—Two Representation Theorems*. March 1999. 16 pp.

6 Rune B. Lyngsø, Christian N. S. Pedersen, and Henrik Nielsen. *Measures on Hidden Markov Models*. February 1999. 27 pp. To appear in *Seventh International Conference*

on Intelligent Systems for Molecular Biology, ISMB '99 Proceedings, 1999.

5 Julian C. Bradfield and Perdita Stevens. *Observational Mu-Calculus*. February 1999. 18 pp.

4 Sibylle B. Fröschle and Thomas Troels Hildebrandt. *On Plain and Hereditary History-Preserving Bisimulation*. February 1999. 21 pp.

3 Peter Bro Miltersen. *Two Notes on the Computational Complexity of One-Dimensional Sandpiles*. February 1999. 8 pp.

2 Ivan B. Damgård. *An Error in the Mixed Adversary Protocol by Fitzi, Hirt and Maurer*. February 1999. 4 pp.

1 Marcin Jurdziński and Mogens Nielsen. *Hereditary History Preserving Simulation is Undecidable*. January 1999. 15 pp.

55 Andrew D. Gordon, Paul D. Hankin, and Søren B. Lassen. *Compilation and Equivalence of Imperative Objects (Revised Report)*. December 1998. iv+75 pp. This is a revision of Technical Report 429, University of Cambridge Computer Laboratory, June 1997, and the earlier BRICS report RS-97-19, July 1997. Appears in Ramesh and Sivakumar, editors, *Foundations of Software Technology and Theoretical Computer Science: 17th Conference*, FST&TCS '97 Proceedings, LNCS 1346, 1997, pages 74–87.

54 Olivier Danvy and Ulrik P. Schultz. *Lambda-Dropping: Transforming Recursive Equations into Programs with Block Structure*. December 1998. 55 pp. This report is superseded by the later report BRICS RS-99-27.

53 Julian C. Bradfield. *Fixpoint Alternation: Arithmetic, Transition Systems, and the Binary Tree*. December 1998. 20 pp.

52 Josva Kleist and Davide Sangiorgi. *Imperative Objects and Mobile Processes*. December 1998. 22 pp. Appears in Gries and de Roever, editors, *IFIP Working Conference on Programming Concepts and Methods*, PROCOMET '98 Proceedings, 1998, pages 285–303.

51 Peter Krogsgaard Jensen. *Automated Modeling of Real-Time Implementation*. December 1998. 9 pp. Appears in *The 13th IEEE Conference on Automated Software Engineering*, ASE '98 Doctoral Symposium Proceedings, 1998, pages 17–20.

50 Luca Aceto and Anna Ingólfsdóttir. *Testing Hennessy-Milner Logic with Recursion*. December 1998. 15 pp. Appears in Thomas, editor, *Foundations of Software Science and Computation Structures: Second International Conference*, FoSSaCS '99 Proceedings, LNCS 1578, 1999, pages 41–55.

49 Luca Aceto, Willem Jan Fokkink, and Anna Ingólfsdóttir. *A Cook's Tour of Equational Axiomatizations for Prefix Iteration*. December 1998. 14 pp. Appears in Nivat, editor, *Foundations of Software Science and Computation Structures: First International Conference*, FoSSaCS '98 Proceedings, LNCS 1378, 1998, pages 20–34.

48 Luca Aceto, Patricia Bouyer, Augusto Burgueño, and Kim G. Larsen. *The Power of Reachability Testing for Timed Automata*. December 1998. 12 pp. Appears in Arvind and Ramanujam, editors, *Foundations of Software Technology and Theoretical Computer Science: 18th Conference*, FST&TCS '98 Proceedings, LNCS 1530, 1998, pages 245–256.

47 Gerd Behrmann, Kim G. Larsen, Justin Pearson, Carsten Weise, and Wang Yi. *Efficient Timed Reachability Analysis using Clock Difference Diagrams*. December 1998. 13 pp. To appear in *Computer-Aided Verification: 11th International Conference*, CAV '99 Proceedings, LNCS, 1999.

46 Kim G. Larsen, Carsten Weise, Wang Yi, and Justin Pearson. *Clock Difference Diagrams*. December 1998. 18 pp. Presented at *10th Nordic Workshop on Programming Theory*, NWPT '10 Abstracts, 1998. To appear in *Nordic Journal of Computing*.

45 Morten Vadskær Jensen and Brian Nielsen. *Real-Time Layered Video Compression using SIMD Computation*. December 1998. 37 pp. Appears in Zinterhof, Vajtersic and Uhl, editors, *Parallel Computing: Fourth International ACPC Conference*, ACPC '99 Proceedings, LNCS 1557, 1999, pages 377–387.

44 Brian Nielsen and Gul Agha. *Towards Reusable Real-Time Objects*. December 1998. 36 pp. To appear in *The Annals of Software Engineering*, IEEE, 7, 1999.

43 Peter D. Mosses. *CASL: A Guided Tour of its Design*. December 1998. 31 pp. To appear in Fiadeiro, editor, *Recent Trends in Algebraic Development Techniques: 13th Workshop*, WADT '98 Selected Papers, LNCS 1589, 1999.

42 Peter D. Mosses. *Semantics, Modularity, and Rewriting Logic*. December 1998. 20 pp. Appears in Kirchner and Kirchner, editors, *International Workshop on Rewriting Logic and its Applications*, WRLA '98 Proceedings, ENTCS 15, 1998.

41 Ulrich Kohlenbach. *The Computational Strength of Extensions of Weak König's Lemma*. December 1998. 23 pp.

40 Henrik Reif Andersen, Colin Stirling, and Glynn Winskel. *A Compositional Proof System for the Modal μ-Calculus*. December 1998. 29 pp.

39 Daniel Fridlender. *An Interpretation of the Fan Theorem in Type Theory*. December 1998. 15 pp. To appear in *International Workshop on Types for Proofs and Programs 1998*, TYPES '98 Selected Papers, LNCS, 1999.

38 Daniel Fridlender and Mia Indrika. *An n-ary* `zipWith` *in Haskell*. December 1998. 12 pp.

37 Ivan B. Damgård, Joe Kilian, and Louis Salvail. *On the (Im)possibility of Basing Oblivious Transfer and Bit Commitment on Weakened Security Assumptions*. December 1998. 22 pp. To appear in *Advances in Cryptology: International Conference on the Theory and Application of Cryptographic Techniques*, EUROCRYPT '99 Proceedings, LNCS, 1999.

36 Ronald Cramer, Ivan B. Damgård, Stefan Dziembowski, Martin Hirt, and Tal Rabin. *Efficient Multiparty Computations with Dishonest Minority*. December 1998. 19 pp. To appear in *Advances in Cryptology: International Conference on the Theory and Application of Cryptographic Techniques*, EUROCRYPT '99 Proceedings, LNCS, 1999.

35 Olivier Danvy and Zhe Yang. *An Operational Investigation of the CPS Hierarchy*. December 1998. Extended version of a paper appearing in Swierstra, editor, *Programming Languages and Systems: Eighth European Symposium on Programming*, ESOP '99 Proceedings, LNCS 1576, 1999, pages 224–242.

34 Peter G. Binderup, Gudmund Skovbjerg Frandsen, Peter Bro Miltersen, and Sven Skyum. *The Complexity of Identifying Large Equivalence Classes*. December 1998. Appears in *Fundamenta Informaticae*, 38:17–29.

33 Hans Hüttel, Josva Kleist, Uwe Nestmann, and Massimo Merro. *Migration = Cloning ; Aliasing (Preliminary Version)*. December 1998. 40 pp. Appears inCardelli, editor, *Foundations of Object-Oriented: 6th International Conference*, FOOL '99 Informal Proceedings, 1999.

32 Jan Camenisch and Ivan B. Damgård. *Verifiable Encryption and Applications to Group Signatures and Signature Sharing*. December 1998. 18 pp.

31 Glynn Winskel. *A Linear Metalanguage for Concurrency*. November 1998. 21 pp.

30 Carsten Butz. *Finitely Presented Heyting Algebras*. November 1998. 30 pp.

29 Jan Camenisch and Markus Michels. *Proving in Zero-Knowledge that a Number is the Product of Two Safe Primes*. November 1998. 19 pp, to appear in *Advances in Cryptology: International Conference on the Theory and Application of Cryptographic Techniques*, EUROCRYPT '99 Proceedings, LNCS, 1999.

28 Rasmus Pagh. *Low Redundancy in Dictionaries with $O(1)$ Worst Case Lookup Time*. November 1998. 15 pp. To appear in *26th International Colloquium on Automata, Languages, and Programming*, ICALP '99 Proceedings, LNCS 1644, 1999.

27 Jan Camenisch and Markus Michels. *A Group Signature Scheme Based on an RSA-Variant*. November 1998. 18 pp. Preliminary version appeared in Ohta and Pei, editors, *Advances in Cryptology: Fourth ASIACRYPT Conference on the Theory and Applications of Cryptologic Techniques*, ASIACRYPT '98 Proceedings, LNCS 1514, 1998, pages 160–174.

26 Paola Quaglia and David Walker. *On Encoding $p\pi$ in $m\pi$*. October 1998. 27 pp. Full version of paper appearing in Arvind and Ramanujam, editors, *Foundations of Software Technology and Theoretical Computer Science: 18th Conference*, FST&TCS '98 Proceedings, LNCS 1530, 1998, pages 42–53.

25 Devdatt P. Dubhashi. *Talagrand's Inequality in Hereditary Settings*. October 1998. 22 pp.

24 Devdatt P. Dubhashi. *Talagrand's Inequality and Locality in Distributed Computing*. October 1998. 14 pp Appears in Luby, Rolim and Serna, editors, *Randomization and Approximation Techniques in Computer Science: Second International Workshop: Second International Workshop*, RANDOM '98 Proceedings, LNCS 1518, 1998, pages 60–70.

23 Devdatt P. Dubhashi. *Martingales and Locality in Distributed Computing*. October 1998. 19 pp. Appears in Arvind and Ramanujam, editors, *Foundations of Software Technology and Theoretical Computer Science: 18th Conference*, FST&TCS '98 Proceedings, LNCS 1530, 1998, pages 174–185.

New in the BRICS Notes Series, 1998–99

ISSN 0909-3206

3 Peter D. Mosses and David A. Watt, editors. *Proceedings of the Second International Workshop on Action Semantics, AS '99*, (Amsterdam, The Netherlands, March 21, 1999), May 1999. iv+172 pp.

2 Hans Hüttel, Josva Kleist, Uwe Nestmann, and António Ravara, editors. *Proceedings of the Workshop on Semantics of Objects As Processes, SOAP '99*, (Lisbon, Portugal, June 15, 1999), May 1999. iv+64 pp.

1 Olivier Danvy, editor. *ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation, PEPM '99*, (San Antonio, Texas, USA, January 22–23, 1999), January 1999.

8 Olivier Danvy and Peter Dybjer, editors. *Proceedings of the 1998 APPSEM Workshop on Normalization by Evaluation, NBE '98 Proceedings*, (Gothenburg, Sweden, May 8–9, 1998), December 1998.

BRICS Lecture Series, 1998

ISSN 1395-2048

4 Paola Quaglia. *The π-Calculus: Notes on Labelled Semantics*. December 1998. viii+16 pp.

Abstract

The π-calculus is a name-passing calculus that allows the description of distributed systems with a dynamically changing interconnection topology. Name communication, together with the possibility of declaring and exporting local names, gives the calculus a great expressive power. For instance, it was remarkably shown that process-passing calculi, that express mobility at higher order, can be naturally encoded in π-calculus.

Since its very first definition, the π-calculus proliferated in a family of calculi slightly de-

parting the another either in the communication paradigm (polyadic vs monadic, asynchronous vs synchronous) or in the bisimulation semantics (labelled vs unlabelled, late vs early vs open vs barbed vs ...).

These short notes present a collection of the labelled strong semantics of the (synchronous monadic) π-calculus. The notes could not possibly substitute any of the standard references listed in the Bibliography. They rather represent an attempt to group together, using a uniform notation and the terminology that got assessed over the last years, a few definitions and concepts otherwise scattered throughout the π-calculus literature.

Contents

3 Olivier Danvy. *Type-Directed Partial Evaluation*. December 1998. Extended version of lecture notes to appear in Hatcliff, Mogensen and Thiemann, editors, *Partial Evaluation: Practice and Theory*, PEPT '98 Lecture Notes, LNCS, 1998.

Abstract

Type-directed partial evaluation uses a normalization function to achieve partial evaluation. These lecture notes review its background, foundations, practice, and applications. Of specific interest is the modular technique of offline and online type-directed partial evaluation in Standard ML of New Jersey.

Contents

2 Carsten Butz. *Regular Categories and Regular Logic*. October 1998.

Abstract

Notes handed out to students attending the course on Category Theory at the Department of Computer Science in Aarhus, Spring 1998. These notes were supposed to give more detailed information about the relationship between regular categories and regular logic than is contained in Jaap van Oosten's script on category theory (BRICS Lectures Series LS-95-1). Regular logic is there called coherent logic.

Contents

BRICS Dissertations Series, 1998

ISSN 1396-7002

3 Kim Sunesen. *Reasoning about Reactive Systems*. December 1998. PhD thesis.

2 Søren B. Lassen. *Relational Reasoning about Functions and Nondeterminism*. December 1998. PhD thesis. x+126 pp.

1 Ole I. Hougaard. *The CLP(OIH) Language*. February 1998. PhD thesis. xii+187 pp.

News and Technical Contributions

Short News

- The Esprit-LTR project VHS, Verification of Hybrid Systems, was successfully reviewed and extended in June 1999 at Bad Malente, Germany.

- The European Educational Forum, EEF, has recently been extended with an Italian and a German chapter.

  The goal of the EEF is to organise training activities directed at PhD students and young researchers from all over Europe.

  In the Italian chapter, IP (Italian inter universitary Partnership), participates the universities of Bologna, Genova, Milano, Pisa, Roma, Salerno and Torino.

  The German chapter includes the three universities RWTH (Rheinisch-Westfälische Technische Hochschule) Aachen, Paderborn and TUM (Technical University of Munich).

- *Eric Jul* from DIKU, University of Copenhagen, has been appointed Honourable Professor at the Department of Computer Science, Aalborg, and is associated the Aalborg Wing of BRICS where he will build up and strength the experimental environment.

- June 1999 professor *Kim Guldstrand Larsen* received a Honorary Doctorate at Uppsala University, Sweden, for his work with UPPAAL—an integrated tool environment for modelling, validation and verification of real-time systems. See also the section on UPPAAL2K on page 35.

- Since summer 1998, *Olivier Danvy* has been serving as co-editor-in-chief of the journal "Higher-Order and Symbolic Computation" (formerly Lisp and Symbolic Computation, `www.wkap.nl/journals/lasc`), together with *Carolyn L. Talcott*, from Stanford University. This journal is generally interested in all aspects of programming languages and programming, including foundational and theoretical ones.

  In particular, volume 13, number 1, is dedicated to Christopher Strachey, at the occasion of the 25th anniversary of his passing away. The issue includes Strachey's 1968 lecture notes on fundamental concepts in programming languages (with a foreword by Peter D. Mosses), Strachey and Wadsworth's original monograph on continuations (with a foreword by Christopher Wadsworth), and a collection of tributes by Rod Burstall, R. Kent Dybvig, Dan Ghica, Mike Gordon, David Hartley, Tony

Hoare, Michael Jackson (the computer scientist), Peter Landin, Robert Milne, Roger Penrose, Martin Richards, Dave Schmidt, Dana Scott, Joe Stoy, and Bob Tennent.

- We heard with sadness of Bob Paige's passing away on Tuesday 5th of October, 1999, in New York, after a long disease. Bob visited Denmark in the mid 90's, giving a BRICS mini-course on *Analysis and Transformation of Set-Theoretic Languages* in the summer of 1995.

Document Structure Description 1.0—an XML Schema Language

by Nils Klarlund, Anders Møller, and Michael I. Schwartzbach

XML (Extensible Markup Language) is rapidly growing as the standard unified notation for mark-up languages, with sponsorship from the W3C and most of industry. The intended benefit is that all (semi)structured documents on the Web can be processed with the same set of tools, thus encouraging inter-corporation and the dynamic exchange of data.

XML is supported by several established tools with many more proposals on the way. Eventually, there will be standard tools for XML transformations, XML layout, XML queries, and much more. Some areas have been firmly developed; for example, XSL (Extensible Stylesheet Language) is close to being adopted as a standard for transformations. In other areas, many different proposals are still being considered. One important such area is that of XML schema languages. These are notations for specifying the syntax of particular applications of XML and should replace the simple DTDs that are native to the XML notation. In many ways, they can also be viewed as generalisations of database schemas.

Document Structure Description 1.0 is a complete specification of a new XML notation for describing classes of XML documents. The notation is designed to be a simple tool based on familiar concepts. DSDs provide more flexible and precise structural descriptions than are possible with DTDs or the current proposal from the W3C XML schema working group. DSD allows context-dependent descriptions of content and attributes, generates a CSS-like (Cascading Style Sheets) default mechanism independent of formatting models, and allows an extension mechanism so that descriptions may be updated with new structural concepts. We have fully implemented a processor that, for a given DSD, can check XML documents in linear time.

The DSD home page is www.brics.dk/DSD.

Déjà vu in Barcelona[1]

Devdatt P. Dubhashi[2]

I don't mean only the feeling on visiting wonderful Barcelona again – Las Ramblas, Gaudi, Sagrada Familia and cafe con leche. But also a *deja vu* of the intellectual kind listening to a delightful invited talk by Emo Welzl at RANDOM '98.

Faithful readers of this newsletter might remember a piece I wrote together with Desh Ranjan entitled "Great(er) Expectations" (BRICS Newsletter 5, July 1996). That's what triggered the feeling in Emo's talk and inspired me to write this report.

Emo's talk dealt with the technique of *random sampling*, a hugely successful paradigm in algorithms in general, and in computational geometry in particular (which was the focus of his talk). Abstractly, we have a set S of objects and we wish to find the "configuration" they form. Random sampling proceeds by picking a smaller sample $R \subseteq S$ at random and determine the

[1] Work done while visiting **BRICS**.

[2] Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, Hauz Khas, New Delhi 110016, INDIA, email: dubhashi@cse.iitd.ernet.in.

"configuration" formed by R first. I'll give several concrete examples from his talk below; in all of them, S consists of points in Euclidean space.

Keeping to the abstract setting, suppose there is a function φ defined on subsets of S. Given $T \subseteq S$ and $x \in S \setminus T$, we say that x *violates* T if $\varphi(T \cup \{x\}) \neq \varphi(T)$. Dually, we call $x \in T$ an *extreme* point of T if $\varphi(T) \neq \varphi(T \setminus \{x\})$. This was the terminology used by Emo. In the terminology of Mulmuley [2, 3], we would talk of the *configuration* $\sigma = \sigma(T)$ determined by T and the set

$$D(\sigma(T)) := \{x \in T \mid \varphi(T) \neq \varphi(T \setminus x)\},$$

of extreme points of T is called the *defining* set of the configuration σ. Similarly the set

$$C(\sigma(T)) := \{x \in S \setminus T \mid \varphi(T) \neq \varphi(T \cup x)\},$$

of violating points is called the *conflict set* of σ. We shall use both terminologies interchangeably and write $D(S)$ or $D(\sigma)$ etc [3]. For the analysis of various random structures and algorithms, one is interested in the expected size of $C(R)$ for a random subset R of size r.

It's high time we gave some concrete examples!

Example 1 (Smallest Number) Let S be a set of real numbers and for $T \subseteq S$, let $\varphi(T) := \min\{x \mid x \in T\}$. Then an extreme point in T is a minimum element and a conflicting point in $S \setminus T$ is an element less than all elements in T.

In this case, the expected value $a_r := \mathtt{E}[C(R) \mid |R| = r]$ of the conflict set of a random sample R of size r is at most $\frac{n-r}{r+1}$. It is instructive to see how a standard approach to showing this goes. Suppose the elements of S are all distinct so that we may as well identify S with $[n]$. For $i \in [n]$, introduce the indicator $X_i := 1[i \in C(R)]$, and noticing that $|C(R)| = \sum_i X_i$ sets up things nicely for linearity of expectation:

$$\begin{aligned} a_r &= \sum_i \mathtt{E}[X_i] \\ &= \sum_i \mathtt{Pr}[X_i = 1] \\ &= \sum_i |\{T \subseteq [n] \mid i < T, |T| = r\}| \binom{n}{r}^{-1} \\ &= \binom{n}{r}^{-1} \sum_{1 \leq i < n} \binom{n-i}{r} \end{aligned}$$

At this point, we are forced to consult Knuth [1, p. 174], where the "upper summation" identity tells us that the binomial coefficient sum is $\binom{n}{r+1}$ and we get the desired result. But already we needed a non–trivial binomial coefficient identity – only the ominous tip of the iceberg!

As you'd imagine, the same thing goes for the kth–smallest number as well. In this case, the result is $a_r \leq k\frac{n-r}{r+1}$.

Example 2 (Convex Hull and Smallest Enclosing Disk) Here S is s set of points in some higher dimensional Euclidean space, and $\varphi(T)$ is the convex hull or the smallest enclosing ball of the set of points in T. For the smallest enclosing ball, $b_r \leq 3$ for all $r \geq 0$ – draw a picture, but don't fall into the trap (as I did) of thinking that this has anything to do with the fact that any three non–collinear points lie on a circle!

So let's return to the general setting: suppose we pick R to be a random sample of size $r \leq n =: |S|$. That is, R is chosen uniformly from amongst all subsets of size r. The question is: what is the expected sizes a_r and b_r of the conflict and defining sets respectively? In symbols, we're interested in:

$$a_r := \mathtt{E}[|C(R)| \mid |R| = r], \quad b_r := \mathtt{E}[|D(R)| \mid |R| = r.$$

They certainly depend on the particular situation at hand, but whatever they are, amazingly, there is always a delightfully simple relation between the two:

$$(r+1)a_r = (n-r)b_{r+1}. \tag{1}$$

Deja vu! Look at how exactly this relation took centre–stage in our newsletter contribution cited

[3] As pointed out to me by Emo subsequently, there are subtle differences between the two sets of concepts in degenerate situations, but I'll ignore that for this report.

above! As Emo remarked, this simple relation lies behind a plethora of results in Computational Geometry based on random sampling. Except that this is completely hidden in the literature where all manner of binomial coefficient identities are invoked for the analysis – this is what I was referring to by the "ominous tip of the iceberg" in Example 1 above. By contrast, the only identity actually needed is the humble

$$\frac{\binom{n}{r+1}}{\binom{n}{r}} = \frac{n-r}{r+1}. \tag{2}$$

Here in fact, is the delightfully simple proof of (1) using only (2). The key point is, as alert readers would have noticed already, there is actually redundancy in the two dual sets of notions we've introduced: a point $x \in T$ is extreme iff it violates $T \setminus x$, i.e.

$$x \in D(T) \quad \leftrightarrow \quad x \in C(T \setminus x).$$

Let us therefore construct a bipartite graph G, between the sets of size r and $r+1$, connecting a set T of size r with a set T' of size $r+1$ iff $T' = T \cup x$ where, equivalently, x is extreme for T' or violates T. *Deja vu*: This is the same bipartite graph we had in "Great(er) Expectations"! Now,

$$\begin{aligned} b_{r+1} &:= \mathtt{E}[D(T') \mid |T'| = r+1] \\ &= \sum_{|T'|=r+1} |D(T')| \binom{n}{r+1}^{-1} \\ &= \sum_{T'} \sum_{x \in D(T')} \binom{n}{r+1}^{-1} \\ &= \sum_{(T,T') \in G} \binom{n}{r+1}^{-1} \end{aligned}$$

Similarly,

$$a_r := \mathtt{E}[C(R) \mid |R| = r] = \sum_{(T,T') \in G} \binom{n}{r}^{-1}.$$

So that finally,

$$\begin{aligned} \frac{a_r}{b_{r+1}} &= \frac{\sum_{(T,T') \in G} \binom{n}{r}^{-1}}{\sum_{(T,T') \in G} \binom{n}{r+1}^{-1}} \\ &= \frac{r+1}{n-r}. \end{aligned}$$

There are several variations possible on the theme. First, we notice that the proof would be the same for any probability distribution that is *symmetric*: we only used the fact that any r element subset has the same probability, $\binom{n}{r}$ of being picked. Thus the result holds for the pseudo–random distributions discussed in [3]. This is of great significance from an algorithmic point of view since it opens the way for standard derandomisation techniques, such as those described in Mulmuley's book.

Even for probability distributions that are not symmetric, one can get similar results. A natural and widely occurring distribution on subsets is the *product measure*:

$$\mu(T) := \prod_{x \in T} p_x \prod_{x \notin T} q_x,$$

where $p_x, q_x, x \in S$ are arbitrary positive reals. *Deja vu*! This is the measure we were concerned with in "Great(er) Expectations". It is also the standard measure for the random graph and it crops up in many other situations like LP–based methods of *randomised rounding*. Reexamining the proof above shows that for such a product measure μ,

$$\frac{a_r}{b_{r+1}} = \frac{\sum_T \mu(T)}{\sum_{T'} \mu(T')},$$

and so,

$$\min_x \frac{q_x}{p_x} \leq \frac{a_r}{b_{r+1}} \leq \max_x \frac{q_x}{p_x}.$$

Another fruitful line is to extend the relations "violate" and "extreme" that we had between a point and a subset to relations between subsets. Say that a disjoint subset U violates T if $U \cap C(T) \neq \emptyset$ i.e. U contains a point that violates T. Dually, say that $U \subseteq T$ is extreme in T if $U \cap D(T) \neq \emptyset$ i.e. U contains an extreme point of T. How are these two related? Unfortunately, it is *not true* that U violates T iff U is extreme for $T \bigcup U$.

Example 3 (Closest pair) Let $\varphi(T)$ denote the distance between a closest pair in T. Let us take a pair of points p, q very close to each other (in particular closer than any pair in T) but far away from all points in T. Now, $\{p, q\}$ is extreme in $T \bigcup \{p, q\}$ but neither p nor q by itself violates T and so neither does the set $\{p, q\}$.

Emo defined a property to satisfy *locality* if in fact this is true: U violates T iff U is extreme for $T \bigcup U$. This is true of all the other examples listed above except the example of closest pairs.

Let us rephrase locality equivalently as follows: U does *not* violate T iff U is *not* extreme for $T \bigcup U$. This prompts us to construct a bipartite graph between sets of size r and sets of size $r + k$ connecting a set T of size r with a set T' of size $r + k$ if $T' = T \bigcup U$ and, equivalently, U does not violate T or U is not extreme for T'. What does the same argument for (1) give us?

$$\binom{n}{r} a_r^k = \binom{n}{r+k} b_{r+k}^k, \tag{3}$$

where

$$a_r^k := \mathtt{E}\left[\binom{n-r-|C(R)|}{k} \mid |R| = r\right], \quad r+k \leq n,$$

and

$$b_r^k := \mathtt{E}\left[\binom{r-|D(R)|}{k} \mid |R| = r\right] \quad k \leq r.$$

Finally, let us see what this implies in *regular* structures which Emo defined as follows: φ is regular for S of *dimension* δ if $|D(T)| = \delta$ for all $T \subseteq S$ with $|T| \geq \delta$. In the smallest number example, we have a local property which is regular of dimension 1 while in the smallest enclosing ball example, we have a local property of dimension at most 3 (nothing to do with three non–colinear points being co–cyclic!). Then since $b_r^k = \binom{r-\delta}{k}$ for all $r \geq \delta$, from (3), we get a *higher moment* formula:

$$\mathtt{E}\left[\binom{n-r-|C(R)|}{k} \mid |R| = r\right] = \frac{\binom{n}{r+k}\binom{r+k}{k}}{\binom{n}{r}}.$$

Notice that this depends only on the locality and regularity properties – these two alone completely determine the distribution of $|C(R)|$, at least up to the applicable moments! Hence, one can apply a tail estimate obtained from such an applicable kth moment for one concrete simple problem, say, the smallest number, and apply it without change to any other problem sharing the same locality and regularity properties! Notice also that the analysis is valid without any independence assumptions which will open the avenues for derandomisation.

There are lots of other variations on the theme that I'm sure occur to you as you read this – for instance, a weaker version of locality that would be satisfied by the closet pair example. Emo and his co–author Bernd Gärtner have promised that they will soon write up a fuller account.

References

[1] R. Graham, D. Knuth and O. Patshnik, *Concrete Mathematics*, Addison Wesley, 1989.

[2] K. Mulmuley, "Randomised Geometric Algorithms and Pseudorandom Generators", *Algorithmica*, 16 pp. 450–463, 1996.

[3] K. Mulmuley, *Computational geometry: An Introduction through Randomised Algorithms*, Prentice Hall.

UPPAAL2k

by Paul Pettersson and Kim Guldstrand Larsen

Background

UPPAAL [1] is a tool for modelling, simulation and verification of real-time systems, developed jointly by BRICS at Aalborg University and the Department of Computer Systems at Uppsala University. The tool is appropriate for systems that can be modelled as a collection of non-deterministic processes with finite control structure and real-valued clocks, communicating through channels or shared variables. Typical application areas include real-time controllers and communication protocols in particular, those where timing aspects are critical.

UPPAAL consists of three main parts: a description language, a simulator and a model checker. The description language is a non-deterministic guarded command language with real-valued clock variables and simple data types. It serves as a modelling or design language to describe system behaviour as networks of automata extended with clock and data variables. The simulator is a validation tool which enables examination of possible dynamic executions of a system during early design (or modelling) stages and thus provides an inexpensive mean of fault detection prior to verification by the model checker which covers the exhaustive dynamic behaviour of the system. The model checker is to check invariant and bounded-liveness properties by exploring the symbolic state-space of a system, i.e., reachability analysis in terms of symbolic states represented by constraints.

Since the first release of UPPAAL in 1995, the tool has been further developed by the teams in Aalborg and Uppsala. Figure 13 illustrates how this has affected the performance of the tool in terms of three examples from the literature.

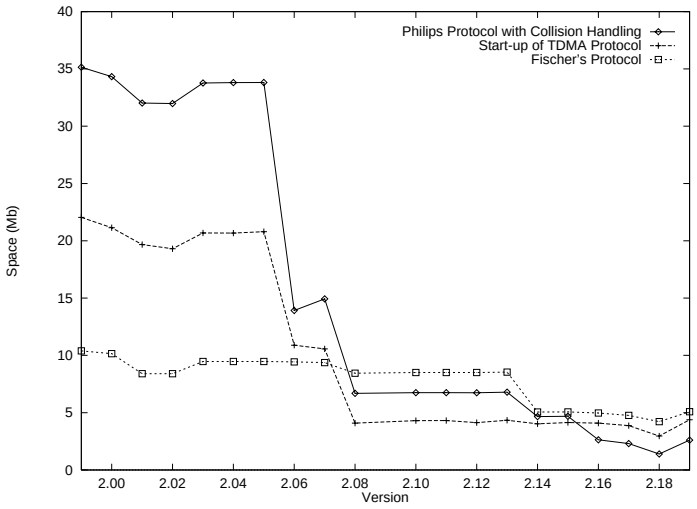


Figure 13: Time (in seconds) benchmarks for UPPAAL version 1.99 to 2.19. Version 1.99 and 2.19 are dated December 1996 and September 1998 respectively. All tool versions were compiled with gcc 2.7.2.3 and executed on the same Pentium II 375 MHz machine.

The diagram shows how the time requirements of UPPAAL improved in the period December 1996 to September 1998 when compiled with the same compiler and executed on the same machine. The space reduction is similar [2].

The New GUI

In July 1999 a new version of UPPAAL, called UPPAAL2k, was released. The new version, which required almost two years of development, is designed to improve the graphical interface of the tool, to allow for easier maintenance, and to be portable to the most common operating systems while still preserving UPPAAL's ease-of-use and efficiency. To meet these requirements the new version is designed as a client/server application with a verification server providing efficient C++ services to a Java client over a socket based protocol. This design also makes it possible to execute the server and the GUI on two different machines.

The new GUI, shown in Figure 14, has new interfaces for the three main tool components of UPPAAL, i.e., the system editor, the simulator and the verifier. Being integrated in one common

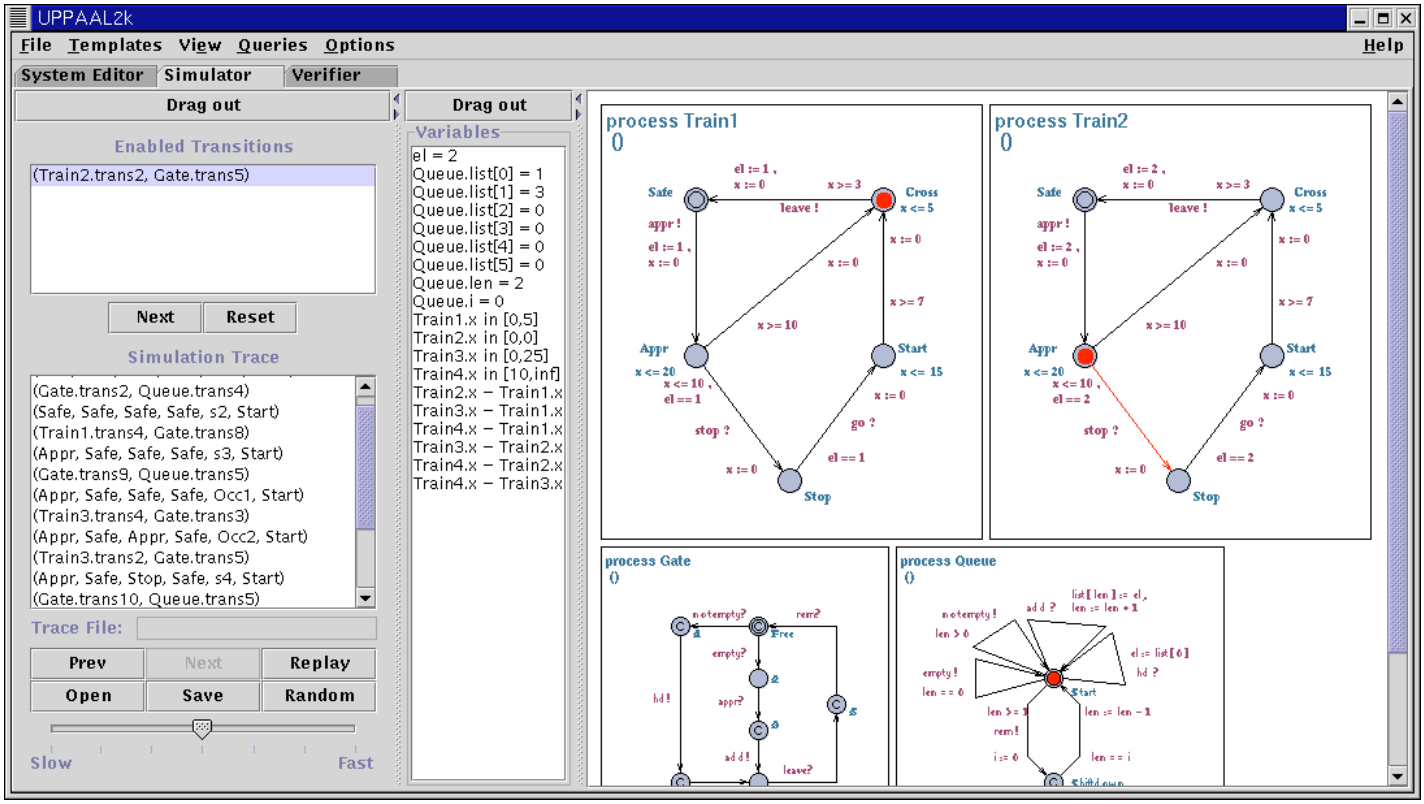


Figure 14: UPPAAL2k on screen.

interface, the three tools now have more uniform interfaces compared to previous UPPAAL versions. The three tools operate on the same internal system model which makes exchange of information between the tools easier, e.g., loading a diagnostic trace generated by the verifier into the simulator for further inspection. In addition, several new functionalities have been implemented in the tool. For example, the new system editor has been tailored and extended for the new system description language of UPPAAL2k (see below), the simulator has been modified to allow the user to configure the level of details to be displayed of the simulated system, and the verification interface has been enriched with a requirement specification editor which stores the previous verification results of a logical property until the property or the system description is modified.

The new UPPAAL version also has a richer modelling language than its predecessors. The new language supports process templates and more complex (bounded) data structures, such as data variables, constants, arrays etc. A process template in the new language is a timed automaton extended with a list of formal parameters and a set of locally declared clocks, variables and constants. Typically, a system description will consist of a set of instances of timed automata declared from the process templates, and of some global data, such as global clocks, variables, synchronisation channels etc. In addition, automata instances may also be defined from templates reused from existing system descriptions. Thus, the adopted notion of process templates (particularly when used in combination with the possibility to declare local process data) allows for convenient re-use of existing models.

UPPAAL2k is currently available for Linux, SunOS and MS Windows platforms. It can be downloaded from the UPPAAL home page www.uppaal.com. Since July 1999, the tool has been downloaded by 149 different users in 60 countries.

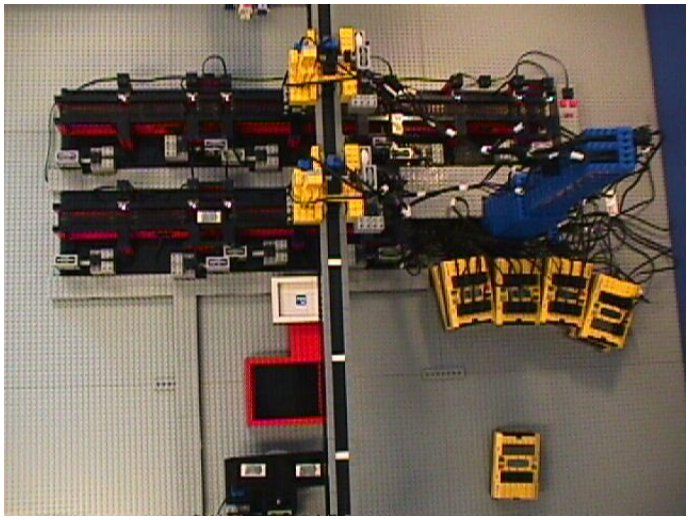


Figure 15: An overview of the LEGO® plant.

The New Verifier

A main focus of the UPPAAL project is to develop efficient algorithms and data structures for the verification of timed systems. The new verification server of UPPAAL2k contains some recent developments in this area (though some of the implementations are not yet available in the public version).

In two recent papers [3, 4], Behrmann et al presents a new data structure called *Clock Difference Diagrams*, CDDs. The new structure is BDD-like (it allows for sharing of isomorphic sub trees) but intended for representing and efficiently manipulating the non convex subsets of the Euclidean space encountered during verification of timed automata. The CDDs have been implemented in UPPAAL to perform the symbolic state-space exploration instead of the normally used data structure, called DBMs. In an experiment where the tool was applied to eight industrial examples, the space savings using CDDs were between 46% and 99% with moderate increase in run time.

Another recent paper [5] describes a new verification technique called *Compositional Backwards Reachability*, CBR. The technique uses *compositionality* and *dependency analysis* to improve the efficiency of symbolic model checking of state/event models. In an untimed setting, the technique has made possible automatic verification of very large industrial design. For example a system with 1421 concurrent machines was checked in less than 20 minutes on a standard PC. An implementation of this technique for timed systems is currently under development and has already proved its applicability on some benchmark examples.

The UPPAAL2k verification server has also been extended with some verification techniques described elsewhere in the literature. The current version supports the *bit-state hashing* under-approximation technique which has been successfully used in the model-checking tool SPIN for several years. A technique for generating an over-approximation of a system's reachable state-space based on a *convex-hull* representations of constraints is also supported. Finally, an abstraction technique based on *(in-)active clock reductions* will be available in the next release (in December 1999).

The New Case Studies

UPPAAL2k has already been applied in case studies. In this section we briefly describe some recent case studies performed at BRICS.

In [6], Hune et al. addresses the problem of synthesising production schedules and control programs for the batch production plant model built in LEGO® MINDSTORMS™ RCX™ shown in Figure 15, 16 and 17. A timed automata model of the plant which faithfully reflects the level of abstraction needed to synthesise control programs is described. This makes the model very detailed and complicated for automatic analysis. To solve this problem a general way of adding guidance to a model by augmenting it with additional guidance variables and transition guards is presented. Applying the technique makes synthesis of control problems feasible for a plant producing as many as 60 batches. In comparison, only two batches could be scheduled without guides. The synthesised control programs have been executed in the plant. Doing this revealed some model errors.

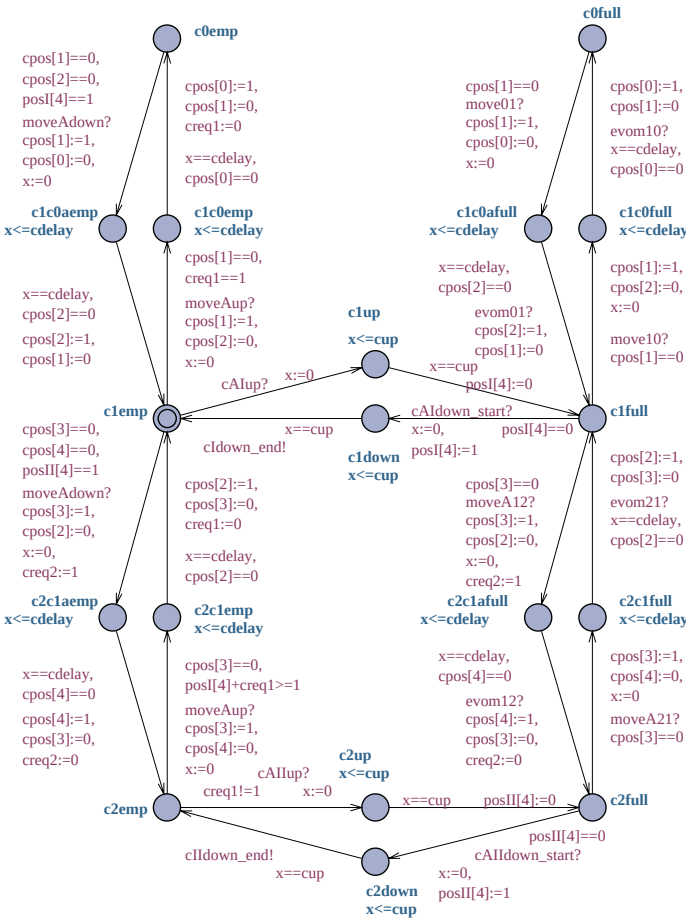


Figure 16: A timed automaton modelling a crane.

```
''''Delay 15
PB.Wait 2, 1500

''''cAIup();
''''Crane A - Pick UP
PB.PlaySystemSound 1
PB.SendPBMessage 2, 97 ' Pick up, on C1
PB.SetVar 1, 15, 0 'Wait for ack
PB.While 0, 1, 3, 2, 97
  PB.Wait 2, 20
  PB.SetVar 1, 15, 0 'Read the message
  PB.ClearPBMessage
  PB.SumVar 2, 2, 1
  PB.If 0, 2, 2, 2, 20
    'If looped 20 times
    PB.PlaySystemSound 1
    PB.SendPBMessage 2, 97 'Then Send
      'message, again same as sendig 0
    PB.SetVar 2, 2, 0
    PB.EndIf
PB.EndWhile

''''Delay 10
PB.Wait 2, 1000
```

Figure 17: A (partial) LEGO® RCX™ program.

The paper [7] and more recently [8, 9] also consider systems controlled by LEGO® RCX™ bricks. Here the studied problem is that of checking properties of the actual programs, rather than abstract models of programs. It is shown how UPPAAL models can be automatically synthesised from RCX™ programs, written in the programming language *Not Quite C*, NQC. Moreover, a protocol to facilitate the distribution of NQC programs over several RCX™ bricks is developed and proved to be correct. The developed translation and protocol are applied to a distributed LEGO® system with two RCX™ bricks pushing boxes between two conveyer belts moving in opposite directions. The system is modelled and some verification results with UPPAAL2k are reported.

In [10], Kristoffersen et al present an analysis of an experimental batch plant using UPPAAL2k. The plant is modelled as a network of timed automata where automata are used for modelling the physical components of the plant, such as the valves, pumps, tanks etc. To model the actual levels of liquid in the tanks, integer variables are used in combination with real-valued clocks which control the change between the (discrete) levels at instances of time which may be predicted from a more accurate hybrid automata model. An crucial assumption of this discretisation is that the interaction between the tanks and the rest of the plant must be such that any plant event affecting the tanks only occurs at these time instances. If this assumption can be guaranteed (which is one of the verification efforts in this framework), the verification results are exact and not only conservative with respect to a more accurate model, where the continuous change of the levels may have been given by some suitable differential equation.

Further Information

More information about UPPAAL can be found on the website www.uppaal.com/. The page contains files and instructions for installing UPPAAL2k, as well as postscript and pdf versions of the papers referred to in this document.

References

[1] UPPAAL *in a Nutshell*, Kim G. Larsen, Paul Pettersson, and Wang Yi. In International Journal on Software Tools for Technology Transfer 1(1–2), pages 134–152. Springer-Verlag. 1998.

[2] *Modelling and Analysis of Real-Time Systems Using Timed Automata: Theory and Practice*, Paul Pettersson. PhD Thesis, Uppsala University. 1999.

[3] *Efficient Timed Reachability Analysis Using Clock Difference Diagrams*, Gerd Behrmann, Kim G. Larsen, Justin Pearson, Carsten Weise, and Wang Yi. In Proceedings of International Conference on Computer Aided Verification. LNCS, Springer-Verlag 1999.

[4] *Clock Difference Diagrams*, Kim G. Larsen, Justin Pearson, Carsten Weise, and Wang Yi. In Nordic Journal of Computing, 6(3), pages 271–298, 1999.

[5] *Verification of Large State/Event Systems using Compositionally and Dependency Analysis*, Jørn Lind-Nielsen, Henrik Reif Andersen, Gerd Behrmann, Henrik Hulgaard, Kåre J. Kristoffersen, and Kim G. Larsen. In Proceedings of International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pages 201–206. LNCS 1384, Springer-Verlag, 1999.

[6] *Guided Synthesis of Control Programs Using* UPPAAL, Thomas Hune, Kim G. Larsen, and Paul Pettersson. Submitted for publication. 1999.

[7] *Modelling a real-time language*, Thomas Hune. In Proceedings of workshop on Formal Methods for Industrial Critical Systems. 1999.

[8] *Verifying Distributed LEGO RCX Programs Using* UPPAAL, Morten Laursen, Rune G. Madsen and Steffen K. Mortensen. 8*th* Semester Project Report. Aalborg University. 1999.

[9] *Automatic Verification of LEGO RCX Systems Using* UPPAAL, Torsten K. Iversen and Cris. B. Thomasen. 8*th* Semester Project Report. Aalborg University. 1999.

[10] *Experimental Batch Plant - VHS Case Study 1 Using Timed Automata and* UPPAAL, Kåre J. Kristoffersen, Kim G. Larsen, Paul Pettersson, and Carsten Weise. Deliverable of EPRIT-LTR Project 26270 VHS (Verification of Hybrid Systems).

Calendar of Events

Date	Event
Late Aug '00	Workshop on Probabilistic Methods in Combinatorial Optimisation, Aarhus
May '01	MFPS, 17th Conference on the Mathematical Foundations of Programming Semantics, Aarhus
May '01	PADO, 2nd workshop on Programs as Data Objects, Aarhus
Jun / Jul' 01	EFF Summer School on Logical Methods, Aarhus
Aug '01	CONCUR, 12th International Conference on Concurrency Theory, Aalborg
Aug '01	ESA, 9th Annual European Symposium on Algorithms, Aarhus
Aug '01	WAE, 5th Workshop on Algorithm Engineering, Aarhus
2002	EFF Summer School on Massive Data Sets, Aarhus

BRICS Address and World Wide Web

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:

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: Faculty of Science
University of Aarhus