BRICS Research Series, Abstracts, 1996

April 29, 2003

This document is also available as *PostScript* and *DVI*.

References

[RS-96-62] *PostScript*, *PDF*, *DVI*.
P. S. Thiagarajan and Igor Walukiewicz.
An Expressively Complete Linear Time Temporal Logic for Mazurkiewicz Traces.
December 1996.
19 pp. Appears in *Twelfth Annual IEEE Symposium on Logic in Computer Science*, LICS '97 Proceedings, 1997, pages 183–194.

Abstract: A basic result concerning LTL, the propositional temporal logic of linear time, is that it is expressively complete; it is equal in expressive power to the first order theory of sequences. We present here a smooth extension of this result to the class of partial orders known as Mazurkiewicz traces. These partial orders arise in a variety of contexts in concurrency theory and they provide the conceptual basis for many of the partial order reduction methods that have been developed in connection with LTL-specifications.

We show that $LTrL$, our linear time temporal logic, is equal in expressive power to the first order theory of traces when interpreted over (finite and) *infinite* traces. This result fills a prominent gap in the existing logical theory of infinite traces. $LTrL$ also provides a syntactic characterisation of the so called trace consistent (robust) LTL-specifications. These are specifications expressed as LTL formulas that do not distiguish between different linearizations of the same trace and hence are amenable to partial order reduction methods.

[RS-96-61] *PostScript*, *PDF*, *DVI*.
Sergei Soloviev.
Proof of a Conjecture of S. Mac Lane.
December 1996.
53 pp. Extended abstract appears in Pitt, Rydeheard and Johnstone, editors, *Category Theory and Computer Science: 6th International Conference*, CTCS '95 Proceedings, LNCS 953, 1995, pages 59–80 Journal version appears in *Annals of Pure and Applied Logic*, 90(1–3):101–162, 1997.

Abstract: Some sufficient conditions on a Symmetric Monoidal Closed category **K** are obtained such that a diagram in a free SMC category generated by the set **A** of atoms commutes if and only if all its interpretations in **K** are commutative. In particular, the category of vector spaces on any field satisfies these conditions (only this case was considered in the original Mac Lane conjecture). Instead of diagrams, pairs of derivations in Intuitionistic Multiplicative Linear logic can be considered (together with categorical equivalence). Two derivations of the same sequent are equivalent if and only if all their interpretations in **K** are equal. In fact, the assignment of values (objects of **K**) to atoms is defined constructively for each pair of derivations. Taking into account a mistake in R. Voreadou's proof of the "abstract coherence theorem" found by the author, it was necessary to modify her description of the class of non-commutative diagrams in SMC categories; our proof of S. Mac Lane conjecture proves also the correctness of the modified description.

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

Abstract: UPPAAL is a tool suite for automatic verification of safety and bounded liveness properties of real-time systems modeled as networks of timed automata , developed during the past two years. In this paper, we summarize the main features of UPPAAL in particular its various extensions developed in 1995 as well as applications to various case-studies, review and provide pointers to the theoretical foundation..

The current version of UPPAAL is available on the World Wide Web via the UPPAAL home page `http://www.docs.uu.se/-docs/rtmv/uppaal`.

[RS-96-59] *PostScript*, *PDF*.
Kim G. Larsen, Paul Pettersson, and Wang Yi.
Compositional and Symbolic Model-Checking of Real-Time Systems.
December 1996.
12 pp. Appears in *The 16th IEEE Real-Time Systems Symposium*, RTSS '95 Proceedings, 1995, pages 76–89.

Abstract: Efficient automatic model-checking algorithms for real-time systems have been obtained in recent years based on the state-region graph technique of Alur, Courcoubetis and Dill. However, these algorithms are faced with two potential types of explosion arising from parallel composition: explosion in the space of control nodes, and explosion in the region space over clock-variables.

In this paper we attack these explosion problems by developing and combining *compositional* and *symbolic* model-checking techniques. The presented techniques provide the foundation for a new automatic verification tool UPPAAL. Experimental results indicate that UPPAAL performs time- and space-wise favorably compared with other real-time verification tools.

[RS-96-58] *PostScript*, *PDF.*
Johan Bengtsson, Kim G. Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi.
UPPAAL — *a Tool Suite for Automatic Verification of Real–Time Systems.*
December 1996.
12 pp. Appears in Alur, Henzinger and Sontag, editors, *Hybrid Systems III: Verification and Control*, LNCS 1066, 1995, pages 232–243.
Abstract: UPPAAL is a tool suite for automatic verification of safety and bounded liveness properties of real-time systems modeled as networks of timed automata. It includes: a *graphical interface* that supports graphical and textual representations of networks of timed automata, and automatic transformation from graphical representations to textual format, a *compiler* that transforms a certain class of linear hybrid systems to networks of timed automata, and a *model–checker* which is implemented based on constraint–solving techniques. UPPAAL also supports diagnostic model-checking providing diagnostic information in case verification of a particular real-time systems fails.

The current version of UPPAAL is available on the World Wide Web via the UPPAAL home page `http://www.docs.uu.se/-docs/rtmv/uppaal`.

[RS-96-57] *PostScript*, *PDF.*
Kim G. Larsen, Paul Pettersson, and Wang Yi.
Diagnostic Model-Checking for Real-Time Systems.
December 1996.
12 pp. Appears in Alur, Henzinger and Sontag, editors, *Hybrid Systems III: Verification and Control*, LNCS 1066, 1995, pages 575–586.
Abstract: UPPAAL is a new tool suit for automatic verification of networks of timed automata. In this paper we describe the diagnostic model-checking feature of UPPAAL and illustrates its usefulness through the debugging of (a version of) the Philips Audio-Control Protocol. Together with a graphical interface of UPPAAL this diagnostic feature allows for a number of errors to be more easily detected and corrected.

[RS-96-56] *PostScript*, *PDF.*
Zine-El-Abidine Benaissa, Pierre Lescanne, and Kristoffer H. Rose.
Modeling Sharing and Recursion for Weak Reduction Strategies using Explicit Substitution.
December 1996.

35 pp. Appears in Kuchen and Swierstra, editors, *Programming Languages, Implementations, Logics, and Programs: 8th International Symposium*, PLILP '96 Proceedings, LNCS 1140, 1996, pages 393–407.

Abstract: We *present* the $\lambda\sigma_w^a$-calculus, a formal synthesis of the concepts of sharing and explicit substitution for weak reduction. We show how $\lambda\sigma_w^a$ can be used as a foundation of implementations of functional programming languages by modeling the essential ingredients of such implementations, namely *weak reduction strategies*, *recursion*, *space leaks*, *recursive data structures*, and *parallel evaluation*, in a uniform way.

First, we give a precise account of the major reduction strategies used in functional programming and the consequences of choosing λ-graph-reduction vs. environment-based evaluation. Second, we show how to add *constructors and explicit recursion* to give a precise account of recursive functions and data structures even with respect to space complexity. Third, we formalize the notion of *space leaks* in $\lambda\sigma_w^a$ and use this to define a space leak free calculus; this suggests optimisations for call-by-need reduction that prevent space leaking and enables us to prove that the "trimming" performed by the STG machine does not leak space.

In summary we give a formal account of several implementation techniques used by state of the art implementations of functional programming languages.

[RS-96-55] *PostScript.*

Kre J. Kristoffersen, Franois Laroussinie, Kim G. Larsen, Paul Pettersson, and Wang Yi.

A Compositional Proof of a Real-Time Mutual Exclusion Protocol.

December 1996.

14 pp. Appears with the title *A Compositional Proof of a Mutual Exclusion Protocol* in Bidoit and Dauchet, editors, *Theory and Practice of Software Development: 7th International Joint Conference CAAP/FASE*, TAPSOFT '97 Proceedings, LNCS 1214, 1997, pages 565–579.

Abstract: In this paper, we apply a compositional proof technique to an automatic verification of the correctness of Fischer's mutual exclusion protocol. It is demonstrated that the technique may avoid the state–explosion problem. Our compositional technique has recently been implemented in a tool CMC[1], which gives experimental evidence that the size of the verification effort required of the technique only grows polynomially in the size of the number of processes in the protocol. In particular, CMC verifies the protocol for 50 processes within 172.3 seconds and using only 32MB main memory. In contrast all existing verification tools for timed systems will suffer from the state–explosion problem, and no tool has to our knowledge succeeded in verifying the protocol for more than 11 processes.

[1]CMC: Compositional Model Checking

[RS-96-54] *PostScript*, *DVI.*
Igor Walukiewicz.
Pushdown Processes: Games and Model Checking.
December 1996.
31 pp. Appears in Alur and Henzinger, editors, *Computer-Aided Verification: 8th International Conference*, CAV '96 Proceedings, LNCS 1102, 1996, pages 62–74. Journal version appears in *Information and Computation*, 164:234–263, 2001.

Abstract: Games given by transition graphs of pushdown processes are considered. It is shown that if there is a winning strategy in such a game then there is a winning strategy that is realized by a pushdown process. This fact turns out to be connected with the model checking problem for the pushdown automata and the propositional μ-calculus. It is show that this model checking problem is DEXPTIME-complete.

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

Abstract: Action Semantics is a framework for the formal description of programming languages. Its main advantage over other frameworks is pragmatic: action-semantic descriptions (ASDs) scale up smoothly to realistic programming languages. This is due to the inherent extensibility and modifiability of ASDs, ensuring that extensions and changes to the described language require only proportionate changes in its description. (In denotational or operational semantics, adding an unforeseen construct to a language may require a reformulation of the entire description.)

After sketching the background for the development of action semantics, we summarize the main ideas of the framework, and provide a simple illustrative example of an ASD. We identify which features of ASDs are crucial for good pragmatics. Then we explain the foundations of action semantics, and survey recent advances in its theory and practical applications. Finally, we assess the prospects for further development and use of action semantics.

The action semantics framework was initially developed at the University of Aarhus by the present author, in collaboration with David Watt (University of Glasgow). Groups and individuals scattered around five continents have since contributed to its theory and practice.

[RS-96-52] *PostScript*, *DVI.*
Claus Hintermeier, Hlne Kirchner, and Peter D. Mosses.
Combining Algebraic and Set-Theoretic Specifications (Extended Version).

December 1996.

26 pp. Appears in Haveraaen, Owe and Dahl, editors, *Recent Trends in Data Type Specification: 11th Workshop on Specification of Abstract Data Types*, RTDTS '95 Selected Papers, LNCS 1130, 1996, pages 255–274.

Abstract: Specification frameworks such as B and Z provide power sets and cartesian products as built-in type constructors, and employ a rich notation for defining (among other things) abstract data types using formulae of predicate logic and lambda-notation. In contrast, the so-called algebraic specification frameworks often limit the type structure to sort constants and first-order functionalities, and restrict formulae to (conditional) equations. Here, we propose an intermediate framework where algebraic specifications are enriched with a set-theoretic type structure, but formulae remain in the logic of equational Horn clauses. This combines an expressive yet modest specification notation with simple semantics and tractable proof theory.

[RS-96-51] *PostScript.*

Claus Hintermeier, Hlne Kirchner, and Peter D. Mosses.

R^n- and G^n-Logics.

December 1996.

19 pp. Appears in Dowek, Heering, Meinke and Mller, editors, *Higher-Order Algebra, Logic, and Term-Rewriting: 2nd International Workshop*, HOA '96 Proceedings, LNCS 1074, 1996, pages 90–108.

Abstract: Specification frameworks such as B and Z provide power sets and cartesian products as built-in type constructors, and employ a rich notation for defining (among other things) abstract data types using formulae of predicate logic and lambda-notation. In contrast, the so-called algebraic specification frameworks often limit the type structure to sort constants and first-order functionalities, and restrict formulae to (conditional) equations. Here, we propose an intermediate framework where algebraic specifications are enriched with a set-theoretic type structure, but formulae remain in the logic of equational Horn clauses. This combines an expressive yet modest specification notation with simple semantics and tractable proof theory.

[RS-96-50] *PostScript, DVI.*

Aleksandar Pekeč.

Hypergraph Optimization Problems: Why is the Objective Function Linear?

December 1996.

10 pp.

Abstract: Choosing an objective function for an optimization problem is a modeling issue and there is no a-priori reason that the objective function must be linear. Still, it seems that linear 0-1 programming formulations are overwhelmingly used as models for optimization problems over discrete structures. We show that this is not an accident. Under some reasonable conditions (from

the modeling point of view), the linear objective function is the only possible one.

[RS-96-49] *PostScript*, *DVI.*

Dan S. Andersen, Lars H. Pedersen, Hans Httel, and Josva Kleist.

Objects, Types and Modal Logics.

December 1996.

20 pp. Appears in Pierce, editor, *4th International Workshop on the Foundations of Object-Oriented*, FOOL4 Proceedings, 1997.

Abstract: In this paper we present a modal logic for describing properties of terms in the object calculus of Abadi and Cardelli. The logic is essentially the modal mu-calculus. The fragment allows us to express the temporal modalities of the logic CTL. We investigate the connection between the type system $\mathbf{Ob}_{1<:\mu}$ and the mu-calculus, providing a translation of types into modal formulae and an ordering on formulae that is sound w.r.t. to the subtype ordering of $\mathbf{Ob}_{1<:\mu}$.

[RS-96-48] *PostScript*, *DVI.*

Aleksandar Pekeč.

Scalings in Linear Programming: Necessary and Sufficient Conditions for Invariance.

December 1996.

28 pp.

Abstract: We analyze invariance of the conclusion of optimality for the linear programming problem under scalings (linear, affine,...) of various problem parameters such as: the coefficients of the objective function, the coefficients of the constraint vector, the coefficients of one or more rows (columns) of the constraint matrix. Measurement theory concepts play a central role in our presentation and we explain why such approach is a natural one.

[RS-96-47] *PostScript*, *DVI.*

Aleksandar Pekeč.

Meaningful and Meaningless Solutions for Cooperative N-person Games.

December 1996.

28 pp.

Abstract: Game values often represent data that can be measured in more than one acceptable way (e.g., monetary amounts). We point out that in such cases a statement about cooperative n-person game model might be “meaningless” in the sense that its truth or falsity depends on the choice of an acceptable way to measure game values. In particular we analyze statements about solution concepts such as the core, stable sets, the nucleolus, the Shapley value (and its generalizations).

[RS-96-46] *PostScript*, *DVI.*

Alexander E. Andreev and Sergei Soloviev.

A Decision Algorithm for Linear Isomorphism of Types with Complexity $Cn(log^2(n))$.

November 1996.

16 pp.

Abstract: It is known that ordinary isomorphisms (associativity and commutativity of "times", isomorphisms for "times" unit and currying) provide a complete axiomatisation for linear isomorphism of types. One of the reasons to consider linear isomorphism of types instead of ordinary isomorphism was that better complexity could be expected. Meanwhile, no upper bounds reasonably close to linear were obtained. We describe an algorithm deciding if two types are linearly isomorphic with complexity $Cn(log^2(n))$.

[RS-96-45] *PostScript.*
Ivan B. Damgrd, Torben P. Pedersen, and Birgit Pfitzmann.
Statistical Secrecy and Multi-Bit Commitments.
November 1996.
30 pp. Appears in *IEEE Transactions on Information Theory*, 44(3):1143–1151, (1998).

Abstract: We present and compare definitions of the notion of "statistically hiding" protocols, and we propose a novel statistically hiding commitment scheme. Informally, a protocol statistically hides a secret if a computationally unlimited adversary who conducts the protocol with the owner of the secret learns almost nothing about it. One definition is based on the L_1-norm distance between probability distributions, the other on information theory. We prove that the two definitions are essentially equivalent. For completeness, we also show that statistical counterparts of definitions of computational secrecy are essentially equivalent to our main definitions.

Commitment schemes are an important cryptologic primitive. Their purpose is to commit one party to a certain value, while hiding this value from the other party until some later time. We present a statistically hiding commitment scheme allowing commitment to many bits. The commitment and reveal protocols of this scheme are constant round, and the size of a commitment is independent of the number of bits committed to. This also holds for the total communication complexity, except of course for the bits needed to send the secret when it is revealed. The proof of the hiding property exploits the equivalence of the two definitions.

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

Abstract: This paper investigates presheaf models for process calculi with value passing. Denotational semantics in presheaf models are shown to correspond to operational semantics in that bisimulation obtained from open maps is proved to coincide with

bisimulation as defined traditionally from the operational semantics. Both "early" and "late" semantics are considered, though the more interesting "late" semantics is emphasised. A presheaf model and denotational semantics is proposed for a language allowing process passing, though there remains the problem of relating the notion of bisimulation obtained from open maps to a more traditional definition from the operational semantics. A tentative beginning is made of a "domain theory" supporting presheaf models.

[RS-96-43] *PostScript*, *DVI*.
Anna Inglfsdttir.
Weak Semantics Based on Lighted Button Pressing Experiments: An Alternative Characterization of the Readiness Semantics.
November 1996.
36 pp. Appears in van Dalen and Bezem, editors, *European Association for Computer Science Logic: 10th Workshop*, CSL '96 Selected Papers, LNCS 1258, 1997, pages 226–243.
Abstract: Imposing certain restrictions on the transition system that defines the behaviour of a process allows us to characterize the readiness semantics of Olderog and Hoare by means of black–box testing experiments, or more precisely by the lighted button testing experiments of Bloom and Meyer. As divergence is considered we give the semantics as a preorder, the readiness preorder, which kernel coincides with the readiness equivalence of Olderog and Hoare. This leads to a bisimulation like characterization and a modal characterization of the semantics. A concrete language, recursive free CCS without τ, is introduced, a proof system defined and it is shown to be sound and complete with respect to the readiness preorder. In the completeness proof the modal characterization plays an important role as it allows us to prove algebraicity of the preorder purely operationally.

[RS-96-42] *PostScript*, *DVI*.
Gerth Stlting Brodal and Sven Skyum.
The Complexity of Computing the k-ary Composition of a Binary Associative Operator.
November 1996.
15 pp.
Abstract: We show that the problem of computing all contiguous k-ary compositions of a sequence of n values under an associative and commutative operator requires $3((k-1)/(k+1))n-\mathrm{O}(k)$ operations.

For the operator max we show in contrast that in the decision tree model the complexity is $\left(1+\Theta(1/\sqrt{k})\right)n-\mathrm{O}(k)$. Finally we show that the complexity of the corresponding on-line problem for the operator max is $(2-1/(k-1))n-\mathrm{O}(k)$.

[RS-96-41] *PostScript*, *DVI*.
Stefan Dziembowski.
The Fixpoint Bounded-Variable Queries are PSPACE-Complete.

November 1996.

16 pp. Appears in van Dalen and Bezem, editors, *European Association for Computer Science Logic: 10th Workshop*, CSL '96 Selected Papers, LNCS 1258, 1997, pages 89–105.

Abstract: We study complexity of the evaluation of fixpoint bounded- variable queries in relational databases. We exhibit a finite database such that the problem whether a closed fixpoint formula using only 2 individual variables is satisfied in this database is PSPACE-complete. This clarifies the issues raised by Moshe Vardi ((1995)) *Proceedings of the 14th ACM Symposium on Principles of Database Systems*, pages 266–276). We study also the complexity of query evaluation for a number of restrictions of fixpoint logic. In particular we exhibit a sublogic for which the upper bound postulated by Vardi holds.

[RS-96-40] *PostScript*, *DVI.*

Gerth Stlting Brodal, Shiva Chaudhuri, and Jaikumar Radhakrishnan.

The Randomized Complexity of Maintaining the Minimum.

November 1996.

20 pp. Appears in *Nordic Journal of Computing*, 3(4):337–351, 1996 (special issue devoted to Selected Papers of the 5th Scandinavian Workshop on Algorithm Theory (SWAT'96)). Appears in Karlson and Lingas, editors, *5th Scandinavian Workshop on Algorithm Theory*, SWAT '96 Proceedings, LNCS 1097, 1996, pages 4–15.

Abstract: The complexity of maintaining a set under the operations Insert, Delete and FindMin is considered. In the comparison model it is shown that any randomized algorithm with expected amortized cost t comparisons per Insert and Delete has expected cost at least $n/(e2^{2t}) - 1$ comparisons for FindMin. If FindMin is replaced by a weaker operation, FindAny, then it is shown that a randomized algorithm with constant expected cost per operation exists; in contrast, it is shown that no deterministic algorithm can have constant cost per operation. Finally, a deterministic algorithm with constant amortized cost per operation for an offline version of the problem is given.

[RS-96-39] *PostScript*, *DVI.*

Hans Httel and Sandeep Shukla.

On the Complexity of Deciding Behavioural Equivalences and Preorders – A Survey.

October 1996.

36 pp.

Abstract: This paper gives an overview of the computational complexity of all the equivalences in the linear/branching time hierarchy and the preorders in the corresponding hierarchy of preorders. We consider finite state or *regular* processes as well as infinite-state BPA processes.

A distinction, which turns out to be important in the finite-state processes, is that of *simulation-like* equivalences/preorders

vs. *trace-like* equivalences and preorders. Here we survey various known complexity results for these relations. For regular processes, all simulation-like equivalences and preorders are decidable in polynomial time whereas all trace-like equivalences and preorders are PSPACE-Complete. We also consider interesting special classes of regular processes such as *deterministic*, *determinate*, *unary*, *locally unary*, and *tree-like* processes and survey the known complexity results in these special cases.

For infinite-state processes the results are quite different. For the class of *context-free processes* or *BPA processes* any preorder or equivalence beyond bisimulation is undecidable but bisimulation equivalence is polynomial time decidable for *normed* BPA processes and is known to be elementarily decidable in the general case. For the class of *BPP* processes, all preorders and equivalences apart from bisimilarity are undecidable. However, bisimilarity is decidable in this case and is known to be decidable in polynomial time for normed BPP processes.

[RS-96-38] *PostScript*, *DVI*.
Hans Httel and Josva Kleist.
Objects as Mobile Processes.
October 1996.
23 pp. Presented at *12th Annual Conference on the Mathematical Foundations of Programming Semantics, MFPS '96 Proceedings (Boulder, Colorado, USA, June 3–5*, 1996)), TCS, 1996.

Abstract: The object calculus of Abadi and Cardelli is intended as model of central aspects of object-oriented programming languages. In this paper we encode the object calculus in the asynchronous π-calculus without matching and investigate the properties of our encoding.

[RS-96-37] *PostScript*, *DVI*.
Gerth Stlting Brodal and Chris Okasaki.
Optimal Purely Functional Priority Queues.
October 1996.
27 pp. Appears in *Journal of Functional Programming*, 6(6):839–858, November 1996.

Abstract: Brodal recently introduced the first implementation of imperative priority queues to support *findMin*, *insert*, and *meld* in $O(1)$ worst-case time, and *deleteMin* in $O(\log n)$ worst-case time. These bounds are asymptotically optimal among all comparison-based priority queues. In this paper, we adapt Brodal's data structure to a purely functional setting. In doing so, we both simplify the data structure and clarify its relationship to the binomial queues of Vuillemin, which support all four operations in $O(\log n)$ time. Specifically, we derive our implementation from binomial queues in three steps: first, we reduce the running time of *insert* to $O(1)$ by eliminating the possibility of cascading links; second, we reduce the running time of *findMin* to $O(1)$ by adding a global root to hold the minimum element; and finally, we reduce the running time of *meld* to $O(1)$ by allowing priority queues to contain other priority queues. Each of these steps is expressed using ML-style functors. The last transformation, known as data-structural bootstrapping, is an interesting application of higher-order functors and recursive structures.

[RS-96-36] *PostScript*, *DVI*.
Luca Aceto, Willem Jan Fokkink, and Anna Inglfsdttir.
On a Question of A. Salomaa: The Equational Theory of Regular Expressions over a Singleton Alphabet is not Finitely Based.
October 1996.
16 pp. Appears in *Theoretical Computer Science*, 209(1–2):141–162, December 1998.
Abstract: Salomaa ((1969) *Theory of Automata*, page 143) asked whether the equational theory of regular expressions over a singleton alphabet has a finite equational base. In this paper, we provide a negative answer to this long standing question. The proof of our main result rests upon a model-theoretic argument. For every finite collection of equations, that are sound in the algebra of regular expressions over a singleton alphabet, we build a model in which some valid regular equation fails. The construction of the model mimics the one used by Conway ((1971) *Regular Algebra and Finite Machines*, page 105) in his proof of a result, originally due to Redko, to the effect that infinitely many equations are needed to axiomatize equality of regular expressions. Our analysis of the model, however, needs to be more refined than the one provided by Conway *ibidem*.

[RS-96-35] *PostScript*, *DVI*.
Gian Luca Cattani and Glynn Winskel.
Presheaf Models for Concurrency.
October 1996.
16 pp. Appears in van Dalen and Bezem, editors, *European Association for Computer Science Logic: 10th Workshop*, CSL '96 Selected Papers, LNCS 1258, 1997, pages 58–75.
Abstract: This paper studies presheaf models for concurrent computation. An aim is to harness the general machinery around presheaves for the purposes of process calculi. Traditional models like synchronisation trees and event structures have been shown to embed fully and faithfully in particular presheaf models in such a way that bisimulation, expressed through the presence of a span of open maps, is conserved. As is shown in the work of Joyal and Moerdijk, presheaves are rich in constructions which preserve open maps, and so bisimulation, by arguments of a very general nature. This paper contributes similar results but biased towards questions of bisimulation in process calculi. It is concerned with modelling process constructions on presheaves, showing these preserve open maps, and with transferring such results to traditional models for processes. One new result here is that a wide range of left Kan extensions, between categories of presheaves, preserve open maps. As a corollary, this also implies that any colimit-preserving functor between presheaf categories

preserves open maps. A particular left Kan extension is shown to coincide with a refinement operation on event structures. A broad class of presheaf models is proposed for a general process calculus. General arguments are given for why the operations of a presheaf model preserve open maps and why for specific presheaf models the operations coincide with those of traditional models.

[RS-96-34] *PostScript*, *PDF*, *DVI*.
John Hatcliff and Olivier Danvy.
A Computational Formalization for Partial Evaluation (Extended Version).
October 1996.
67 pp. Appears in *Mathematical Structures in Computer Science*, 7(5):507–541, October 1997.

Abstract: We formalize a partial evaluator for Eugenio Moggi's computational metalanguage. This formalization gives an evaluation-order independent view of binding-time analysis and program specialization, including a proper treatment of call unfolding, and enables us to express the essence of "control-based binding-time improvements" for let expressions. Specifically, we prove that the binding-time improvements given by "continuation-based specialization" can be expressed in the metalanguage via monadic laws.

[RS-96-33] *PostScript*.
Jonathan F. Buss, Gudmund Skovbjerg Frandsen, and Jeffrey Outlaw Shallit.
The Computational Complexity of Some Problems of Linear Algebra.
September 1996.
39 pp. Revised version appears in Reischuk and Morvan, editors, *14th Annual Symposium on Theoretical Aspects of Computer Science*, STACS '97: Proceedings, LNCS 1200, 1997, pages 451–462 and later in *Journal of Computer and System Sciences*, 58(3):572–596, 1998.

Abstract: We consider the computational complexity of some problems dealing with matrix rank. Let E, S be subsets of a commutative ring R. Let $x_1, x_2, \ldots, x_t$ be variables. Given a matrix $M = M(x_1, x_2, \ldots, x_t)$ with entries chosen from $E \cup \{x_1, x_2, \ldots, x_t\}$, we want to determine

$$\mathrm{maxrank}_S(M) = \max_{(a_1,a_2,\ldots,a_t)\in S^t} \mathrm{rank}\ M(a_1, a_2, \ldots a_t)$$

and

$$\mathrm{minrank}_S(M) = \min_{(a_1,a_2,\ldots,a_t)\in S^t} \mathrm{rank}\ M(a_1, a_2, \ldots a_t).$$

There are also variants of these problems that specify more about the structure of M, or instead of asking for the minimum or maximum rank, ask if there is some substitution of the variables that makes the matrix invertible or non invertible.

Depending on E, S, and on which variant is studied, the complexity of these problems can range from polynomial-time solvable to

random polynomial-time solvable to *NP*-complete to *PSPACE*-solvable to unsolvable.

[RS-96-32] *PostScript*, *DVI*.
P. S. Thiagarajan.
Regular Trace Event Structures.
September 1996.
34 pp.
Abstract: We propose trace event structures as a starting point for constructing *effective* branching time temporal logics in a non-interleaved setting. As a first step towards achieving this goal, we define the notion of a regular trace event structure. We then provide some simple characterizations of this notion of regularity both in terms of recognizable trace languages and in terms of finite 1-safe Petri nets.

[RS-96-31] *PostScript*, *DVI*.
Ian Stark.
Names, Equations, Relations: Practical Ways to Reason about 'new'.
September 1996.
ii+22 pp. Please refer to the revised version BRICS-RS-97-39.
Abstract: The nu-calculus of Pitts and Stark is a typed lambda-calculus, extended with state in the form of dynamically-generated names. These names can be created locally, passed around, and compared with one another. Through the interaction between names and functions, the language can capture notions of scope, visibility and sharing. Originally motivated by the study of references in Standard ML, the nu-calculus has connections to other kinds of local declaration, and to the mobile processes of the pi-calculus.

This paper introduces a logic of equations and relations which allows one to reason about expressions of the nu-calculus: this uses a simple representation of the private and public scope of names, and allows straightforward proofs of contextual equivalence (also known as observational, or observable, equivalence). The logic is based on earlier operational techniques, providing the same power but in a much more accessible form. In particular it allows intuitive and direct proofs of all contextual equivalences between first-order functions with local names.

[RS-96-30] *PostScript*, *DVI*.
Arne Andersson, Peter Bro Miltersen, and Mikkel Thorup.
Fusion Trees can be Implemented with AC^0 Instructions only.
September 1996.
8 pp. Appears in *Theoretical Computer Science*, 215(1–2):337–344, 1999.
Abstract: Addressing a problem of Fredman and Willard, we implement fusion trees in deterministic linear space using AC^0 instructions only.

[RS-96-29] *PostScript*.
Lars Arge.

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

Abstract: Ordered Binary-Decision Diagrams (OBDD) are the state-of-the-art data structure for boolean function manipulation and there exist several software packages for OBDD manipulation. OBDDs have been successfully used to solve problems in e.g. digital-systems design, verification and testing, in mathematical logic, concurrent system design and in artificial intelligence. The OBDDs used in many of these applications quickly get larger than the avaliable main memory and it becomes essential to consider the problem of minimizing the Input/Output (I/O) communication. In this paper we analyze why existing OBDD manipulation algorithms perform poorly in an I/O environment and develop new I/O-efficient algorithms.

[RS-96-28] *PostScript.*
Lars Arge.
The Buffer Tree: A New Technique for Optimal I/O Algorithms.
August 1996.
34 pp. This report is a revised and extended version of the BRICS Report RS-94-16. An extended abstract appears in Akl, Dehne, Sack and Santoro, editors, *Algorithms and Data Structures: 4th Workshop*, WADS '95 Proceedings, LNCS 955, 1995, pages 334–345.

Abstract: In this paper we develop a technique for transforming an internal-memory tree data structure into an external-memory structure. We show how the technique can be used to develop a search tree like structure, a priority queue, a (one-dimensional) range tree and a segment tree, and give examples of how these structures can be used to develop efficient I/O algorithms. All our algorithms are either extremely simple or straightforward generalizations of known internal-memory algorithms—given the developed external data structures. We believe that algorithms relying on the developed structure will be of practical interest due to relatively small constants in the asymptotic bounds.

[RS-96-27] *PostScript*, *DVI.*
Devdatt P. Dubhashi, Volker Priebe, and Desh Ranjan.
Negative Dependence Through the FKG Inequality.
July 1996.
15 pp.

Abstract: We investigate random variables arising in occupancy problems, and show the variables to be negatively associated, that is, negatively dependent in a strong sense. Our proofs are based on the FKG correlation inequality, and they suggest a useful, general technique for proving negative dependence among

random variables. We also show that in the special case of two binary random variables, the notions of negative correlation and negative association coincide.

[RS-96-26] *PostScript*, *DVI*.
Nils Klarlund and Theis Rauhe.
BDD Algorithms and Cache Misses.
July 1996.
15 pp.

Abstract: Within the last few years, CPU speed has greatly overtaken memory speed. For this reason, implementation of symbolic algorithms—with their extensive use of pointers and hashing—must be reexamined.

In this paper, we introduce the concept of *cache miss complexity* as an analytical tool for evaluating algorithms depending on pointer chasing. Such algorithms are typical of symbolic computation found in verification.

We show how this measure suggests new data structures and algorithms for multi-terminal BDDs. Our ideas have been implemented in a BDD package, which is used in a decision procedure for the Monadic Second-order Logic on strings.

Experimental results show that on large examples involving e.g the verification of concurrent programs, our implementation runs 4 to 5 times faster than a widely used BDD implementation.

We believe that the method of cache miss complexity is of general interest to any implementor of symbolic algorithms used in verification.

[RS-96-25] *PostScript*, *DVI*.
Devdatt P. Dubhashi and Desh Ranjan.
Balls and Bins: A Study in Negative Dependence.
July 1996.
27 pp. Appears in *Random Structures and Algorithms* 13(2):99–124, 1998.

Abstract: This paper investigates the notion of negative dependence amongst random variables and attempts to advocate its use as a simple and unifying paradigm for the analysis of random structures and algorithms.

[RS-96-24] *PostScript*.
Henrik Ejersbo Jensen, Kim G. Larsen, and Arne Skou.
Modelling and Analysis of a Collision Avoidance Protocol using SPIN and UPPAAL.
July 1996.
20 pp. Appears in Grgoire, Holzmann and Peled, editors, *The Spin Verification System: 2nd International SPIN Workshop*, SPIN '96 Proceedings, American Mathematical Society Press (AMS) DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1997, pages 33–47.

Abstract: This paper compares the tools SPIN and UPPAAL by modelling and verifying a Collision Avoidance Protocol for an

Ethernet-like medium. We find that SPIN is well suited for modelling the untimed aspects of the protocol processes and for expressing the relevant (untimed) properties. However, the modelling of the media becomes awkward due to the lack of broadcast communication in the PROMELA language. On the other hand we find it easy to model the timed aspects using the UPPAAL tool. Especially, the notion of *committed locations* supports the modelling of broadcast communication. However, the property language of UPPAAL lacks some expessivity for verification of bounded liveness properties, and we indicate how timed testing automata may be constructed for such properties, inspired by the (untimed) checking automata of SPIN.

[RS-96-23] *PostScript*, *DVI*.
Luca Aceto, Willem Jan Fokkink, and Anna Inglfsdttir.
A Menagerie of Non-Finitely Based Process Semantics over BPA**: From Ready Simulation Semantics to Completed Tracs.*
July 1996.
38 pp.

Abstract: Fokkink and Zantema ((1994) *Computer Journal* **37**:259–267) have shown that bisimulation equivalence has a finite equational axiomatization over the language of Basic Process Algebra with the binary Kleene star operation (BPA*). In light of this positive result on the mathematical tractability of bisimulation equivalence over BPA*, a natural question to ask is whether any other (pre)congruence relation in van Glabbeek's linear time/branching time spectrum is finitely (in)equationally axiomatizable over it. In this paper, we prove that, unlike bisimulation equivalence, none of the preorders and equivalences in van Glabbeek's linear time/branching time spectrum, whose discriminating power lies in between that of ready simulation and that of completed traces, has a finite equational axiomatization. This we achieve by exhibiting a family of (in)equivalences that holds in ready simulation semantics, the finest semantics that we consider, whose instances cannot all be proven by means of any finite set of (in)equations that is sound in completed trace semantics, which is the coarsest semantics that is appropriate for the language BPA*. To this end, for every finite collection of (in)equations that are sound in completed trace semantics, we build a model in which some of the (in)equivalences of the family under consideration fail. The construction of the model mimics the one used by Conway ((1971) *Regular Algebra and Finite Machines*, page 105) in his proof of a result, originally due to Redko, to the effect that infinitely many equations are needed to axiomatize equality of regular expressions.

Our non-finite axiomatizability results apply to the language BPA* over an arbitrary non-empty set of actions. In particular, of the fact we show that completed trace equivalence is not finitely based over BPA* even when the set of actions is a singleton. Our proof of this result may be easily adapted to the standard

language of regular expressions to yield a solution to an open problem posed by Salomaa ((1969) *Theory of Automata*, page 143).

Another semantics that is usually considered in process theory is trace semantics. Trace semantics is, in general, not preserved by sequential composition, and is therefore inappropriate for the language BPA*. We show that, if the set of actions is a singleton, trace equivalence and preorder are preserved by all the operators in the signature of BPA*, and coincide with simulation equivalence and preorder, respectively. In that case, unlike all the other semantics considered in this paper, trace semantics have finite, complete equational axiomatizations over closed terms.

[RS-96-22] *PostScript*, *DVI*.

Luca Aceto and Willem Jan Fokkink.

An Equational Axiomatization for Multi-Exit Iteration.

June 1996.

30 pp. Appears in *Information and Computation*, 137(2):121–158, 1997.

Abstract: This paper presents an equational axiomatization of bisimulation equivalence over the language of Basic Process Algebra (BPA) with multi-exit iteration. Multi-exit iteration is a generalization of the standard binary Kleene star operation that allows for the specification of agents that, up to bisimulation equivalence, are solutions of systems of recursion equations of the form

$$\begin{array}{rcl} X_1 & = & P_1 X_2 + Q_1 \\ & \vdots & \\ X_n & = & P_n X_1 + Q_n \end{array}$$

where n is a positive integer, and the P_i and the Q_i are process terms. The addition of multi-exit iteration to BPA yields a more expressive language than that obtained by augmenting BPA with the standard binary Kleene star (BPA*). As a consequence, the proof of completeness of the proposed equational axiomatization for this language, although standard in its general structure, is much more involved than that for BPA*. An expressiveness hierarchy for the family of k-exit iteration operators proposed by Bergstra, Bethke and Ponse is also offered.

[RS-96-21] *PostScript*, *DVI*.

Dany Breslauer, Tao Jiang, and Zhigen Jiang.

Rotation of Periodic Strings and Short Superstrings.

June 1996.

14 pp. Appears in *Journal of Algorithms*, 24(2):340–353, August 1997.

Abstract: This paper presents two simple approximation algorithms for the shortest superstring problem, with approximation ratios $2\frac{2}{3}$ (≈ 2.67) and $2\frac{25}{42}$ (≈ 2.596), improving the best previously published $2\frac{3}{4}$ approximation. The framework of our improved algorithms is similar to that of previous algorithms in the sense that they construct a superstring by computing some optimal cycle covers on the distance graph of the given strings, and then break and merge the cycles to finally obtain a Hamiltonian path, but we make use of new bounds on the overlap between two strings. We prove that for each periodic semi-infinite string $\alpha = a_1 a_2 \cdots$ of period q, there exists an integer k, such that for *any* (finite) string s of period p which is *inequivalent* to α, the overlap between s and the *rotation* $\alpha[k] = a_k a_{k+1} \cdots$ is at most $p + \frac{1}{2}q$. Moreover, if $p \leq q$, then the overlap between s and $\alpha[k]$ is not larger than $\frac{2}{3}(p+q)$. In the previous shortest superstring algorithms $p+q$ was used as the standard bound on overlap between two strings with periods p and q.

[RS-96-20] *PostScript*, *DVI*.
Olivier Danvy and Julia L. Lawall.
Back to Direct Style II: First-Class Continuations.
June 1996.
36 pp. A preliminary version of this paper appeared in the proceedings of the 1992 ACM Conference on *Lisp and Functional Programming*, William Clinger, editor, LISP Pointers, Vol. V, No. 1, pages 299–310, San Francisco, California, June 1992. ACM Press.

Abstract: The direct-style transformation aims at mapping continuation-passing programs back to direct style, be they originally written in continuation-passing style or the result of the continuation-passing-style transformation. In this paper, we continue to investigate the direct-style transformation by extending it to programs with first-class continuations.

First-class continuations break the stack-like discipline of continuations in that they are sent results out of turn. We detect them syntactically through an analysis of continuation-passing terms. We conservatively extend the direct-style transformation towards call-by-value functional terms (the pure λ-calculus) by translating the declaration of a first-class continuation using the control operator call/cc, and by translating an occurrence of a first-class continuation using the control operator throw. We prove that our extension and the corresponding extended continuation-passing-style transformation are inverses.

Both the direct-style (DS) and continuation-passing-style (CPS) transformations can be generalized to a richer language. These transformations have a place in the programmer's toolbox, and enlarge the class of programs that can be manipulated on a semantic basis. We illustrate both with two applications: the conversion between CPS and DS of an interpreter hand-written in CPS, and the specialization of a coroutine program, where coroutines are implemented using call/cc. The latter example achieves a first: a static coroutine is executed statically and its computational content is inlined in the residual program.

[RS-96-19] *PostScript*, *DVI*.
John Hatcliff and Olivier Danvy.

Thunks and the λ-Calculus.
June 1996.
22 pp. Appears in *Journal of Functional Programming* 7(3):303–319 (1997).

Abstract: Thirty-five years ago, thunks were used to simulate call-by-name under call-by-value in Algol 60. Twenty years ago, Plotkin presented continuation-based simulations of call-by-name under call-by-value and vice versa in the λ-calculus. We connect all three of these classical simulations by factorizing the continuation-based call-by-name simulation $\mathcal{C}_n$ with a thunk-based call-by-name simulation $\mathcal{T}$ followed by the continuation-based call-by-value simulation $\mathcal{C}_v$ extended to thunks.

We show that $\mathcal{T}$ actually satisfies all of Plotkin's correctness criteria for $\mathcal{C}_n$ (i.e., his Indifference, Simulation, and Translation theorems). Furthermore, most of the correctness theorems for $\mathcal{C}_n$ can now be seen as simple corollaries of the corresponding theorems for $\mathcal{C}_v$ and $\mathcal{T}$.

[RS-96-18] *PostScript*, *DVI*.
Thomas Troels Hildebrandt and Vladimiro Sassone.
Comparing Transition Systems with Independence and Asynchronous Transition Systems.
June 1996.
14 pp. Appears in Montanari and Sassone, editors, *Concurrency Theory: 7th International Conference*, CONCUR '96 Proceedings, LNCS 1119, 1996, pages 84–97.

Abstract: Transition systems with independence and asynchronous transition systems are *noninterleaving* models for concurrency arising from the same simple idea of decorating transitions with events. They differ for the choice of a *derived* versus a *primitive* notion of event which induces considerable differences and makes the two models suitable for different purposes. This opens the problem of investigating their mutual relationships, to which this paper gives a fully comprehensive answer.

In details, we characterise the category of *extensional* asynchronous transitions systems as the largest full subcategory of the category of (labelled) asynchronous transition systems which admits TSI, the category of transition systems with independence, as a *coreflective* subcategory. In addition, we introduce *event-maximal* asynchronous transitions systems and we show that their category is *equivalent* to TSI, so providing an exhaustive characterisation of transition systems with independence in terms of asynchronous transition systems.

[RS-96-17] *PostScript*, *DVI*.
Olivier Danvy, Karoline Malmkjr, and Jens Palsberg.
Eta-Expansion Does The Trick (Revised Version).
May 1996.
29 pp. Appears in *ACM Transactions on Programming Languages and Systems*, 8(6):730–751, 1996.

Abstract: Partial-evaluation folklore has it that massaging one's source programs can make them specialize better. In Jones, Gomard, and Sestoft's recent textbook, a whole chapter is dedicated to listing such "binding-time improvements": non-standard use of continuation-passing style, eta-expansion, and a popular transformation called "The Trick". We provide a unified view of these binding-time improvements, from a typing perspective.

Just as a proper treatment of product values in partial evaluation requires partially static values, a proper treatment of disjoint sums requires moving static contexts across dynamic case expressions. This requirement precisely accounts for the non-standard use of continuation-passing style encountered in partial evaluation. Eta-expansion thus acts as a uniform binding-time coercion between values and contexts, be they of function type, product type, or disjoint-sum type. For the latter case, it enables "The Trick".

In this paper, we extend Gomard and Jones's partial evaluator for the λ-calculus, λ-Mix, with products and disjoint sums; we point out how eta-expansion for (finite) disjoint sums enables The Trick; we generalize our earlier work by identifying that eta-expansion can be obtained in the binding-time analysis simply by adding two coercion rules; and we specify and prove the correctness of our extension to λ-Mix.

[RS-96-16] *PostScript*, *DVI.*
Lisbeth Fajstrup and Martin Rauen.
Detecting Deadlocks in Concurrent Systems.
May 1996.
10 pp. Appears in Sangiorgi and de Simone, editors, *Concurrency Theory: 9th International Conference*, CONCUR '98 Proceedings, LNCS 1466, 1998, pages 332–347.

Abstract: We use a geometric description for deadlocks occurring in scheduling problems for concurrent systems to construct a partial order and hence a directed graph, in which the local maxima correspond to deadlocks. Algorithms finding deadlocks are described and assessed.

[RS-96-15] *PostScript*, *DVI.*
Olivier Danvy.
Pragmatic Aspects of Type-Directed Partial Evaluation.
May 1996.
27 pp. Appears in Danvy, Glck and Thiemann, editors, *Partial Evaluation: International Seminar*, PE '96 Selected Papers, LNCS 1110, 1996, pages 73–94.

Abstract: Type-directed partial evaluation stems from the residualization of static values in dynamic contexts, given their type and the type of their free variables. Its algorithm coincides with the algorithm for coercing a subtype value into a supertype value, which itself coincides with Berger and Schwichtenberg's normalization algorithm for the simply typed λ-calculus. Type-directed partial evaluation thus can be used to specialize a compiled,

closed program, given its type.

Since Similix, let-insertion is a cornerstone of partial evaluators for call-by-value procedural languages with computational effects (such as divergence). It prevents the duplication of residual computations, and more generally maintains the order of dynamic side effects in the residual program.

This article describes the extension of type-directed partial evaluation to insert residual let expressions. This extension requires the user to annotate arrow types with effect information. It is achieved by delimiting and abstracting control, comparably to continuation-based specialization in direct style. It enables type-directed partial evaluation of programs with effects (e.g., a definitional lambda-interpreter for an imperative language) that are in direct style. The residual programs are in A-normal form. A simple corollary yields CPS (continuation-passing style) terms instead. We illustrate both transformations with two interpreters for Paulson's Tiny language, a classical example in partial evaluation.

[RS-96-14] Olivier Danvy and Karoline Malmkjr.
On the Idempotence of the CPS Transformation.
May 1996.
17 pp.

Abstract: The CPS (continuation-passing style) transformation on typed λ-terms has an interpretation in many areas of Computer Science, such as programming languages and type theory. Programming intuition suggests that it in effect, it is idempotent, but this does not directly hold for all existing CPS transformations (Plotkin, Reynolds, Fischer, *etc.*).

We rephrase the call-by-value CPS transformation to make it syntactically idempotent, modulo η-reduction of the newly introduced continuation. Type-wise, iterating the transformation corresponds to refining the polymorphic domain of answers.

[RS-96-13] *PostScript*, *DVI.*
Olivier Danvy and Ren Vestergaard.
Semantics-Based Compiling: A Case Study in Type-Directed Partial Evaluation.
May 1996.
28 pp. Appears in Kuchen and Swierstra, editors, *Programming Languages, Implementations, Logics, and Programs: 8th International Symposium*, PLILP '96 Proceedings, LNCS 1140, 1996, pages 182–197.

Abstract: We illustrate a simple and effective solution to semantics-based compiling. Our solution is based on type-directed partial evaluation, where

- our compiler generator is expressed in a few lines, and is efficient;
- its input is a well-typed, purely functional definitional interpreter in the manner of denotational semantics;

- the output of the generated compiler is three-address code, in the fashion and efficiency of the Dragon Book;
- the generated compiler processes several hundred lines of source code per second.

The source language considered in this case study is imperative, block-structured, higher-order, call-by-value, allows subtyping, and obeys stack discipline. It is bigger than what is usually reported in the literature on semantics-based compiling and partial evaluation.

Our compiling technique uses the first Futamura projection, i.e., we compile programs by specializing a definitional interpreter with respect to this program.

Our definitional interpreter is completely straightforward, stack-based, and in direct style. In particular, it requires no clever staging technique (currying, continuations, binding-time improvements, etc.), nor does it rely on any other framework (attribute grammars, annotations, etc.) than the typed lambda-calculus. In particular, it uses no other program analysis than traditional type inference.

The overall simplicity and effectiveness of the approach has encouraged us to write this paper, to illustrate this genuine solution to denotational semantics-directed compilation, in the spirit of Scott and Strachey.

[RS-96-12] *PostScript*.
Lars Arge, Darren Erik Vengroff, and Jeffrey Scott Vitter.
External-Memory Algorithms for Processing Line Segments in Geographic Information Systems.
May 1996.
34 pp. To appear in *Algorithmica*. A shorter version of this paper appears in Spirakis, editor, *Third Annual European Symposiumon on Algorithms*, ESA '95 Proceedings, LNCS 979, 1995, pages 295–310.

Abstract: In the design of algorithms for large-scale applications it is essential to consider the problem of minimizing I/O communication. Geographical information systems (GIS) are good examples of such large-scale applications as they frequently handle huge amounts of spatial data. In this paper we develop efficient new external-memory algorithms for a number of important problems involving line segments in the plane, including trapezoid decomposition, batched planar point location, triangulation, red-blue line segment intersection reporting, and general line segment intersection reporting. In GIS systems, the first three problems are useful for rendering and modeling, and the latter two are frequently used for overlaying maps and extracting information from them.

[RS-96-11] *PostScript*, *DVI*.
Devdatt P. Dubhashi, David A. Grable, and Alessandro Panconesi.
Near-Optimal, Distributed Edge Colouring via the Nibble Method.

May 1996.

17 pp. Appears in Spirakis, editor, *Third Annual European Symposiumon on Algorithms*, ESA '95 Proceedings, LNCS 979, 1995, pages 448–459. Appears in the special issue of *Theoretical Computer Science*, 203(2):225–251, August 1998, devoted to the proceedings of ESA '95.

Abstract: We give a distributed randomized algorithm to edge colour a network. Let G be a graph with n nodes and maximum degree Δ. Here we prove:

- If $\Delta = \Omega(\log^{1+\delta} n)$ for some $\delta > 0$ and $\lambda > 0$ is fixed, the algorithm almost always colours G with $(1+\lambda)\Delta$ colours in time $O(\log n)$.
- If $s > 0$ is fixed, there exists a positive constant k such that if $\Delta = \Omega(\log^k n)$, the algorithm almost always colours G with $\Delta + \Delta/\log^s n = (1+o(1))\Delta$ colours in time $O(\log n + \log^s n \log\log n)$.

By "almost always" we mean that the algorithm may fail, but the failure probability can be made arbitrarily close to 0.

The algorithm is based on the nibble method, a probabilistic strategy introduced by Vojtěch Rdl. The analysis makes use of a powerful large deviation inequality for functions of independent random variables.

[RS-96-10] *PostScript*, *DVI*.

Torben Braner and Valeria de Paiva.

Cut-Elimination for Full Intuitionistic Linear Logic.

April 1996.

27 pp. Also available as Technical Report 395, Computer Laboratory, University of Cambridge.

Abstract: We describe in full detail a solution to the problem of proving the cut elimination theorem for FILL, a variant of (multiplicative and exponential-free) Linear Logic introduced by Hyland and de Paiva. Hyland and de Paiva's work used a term assignment system to describe FILL and barely sketched the proof of cut elimination. In this paper, as well as correcting a small mistake in their paper and extending the system to deal with exponentials, we introduce a different formal system describing the intuitionistic character of FILL and we provide a full proof of the cut elimination theorem. The formal system is based on a notion of dependency between formulae within a given proof and seems of independent interest. The procedure for cut elimination applies to (classical) multiplicative Linear Logic, and we can (with care) restrict our attention to the subsystem FILL. The proof, as usual with cut elimination proofs, is a little involved and we have not seen it published anywhere.

[RS-96-9] *PostScript*, *DVI*.

Thore Husfeldt, Theis Rauhe, and Sren Skyum.

Lower Bounds for Dynamic Transitive Closure, Planar Point Location, and Parentheses Matching.

April 1996.

11 pp. Appears in *Nordic Journal of Computing*, 3(4):323–336, December 1996. Also in Karlson and Lingas, editors, *5th Scandinavian Workshop on Algorithm Theory*, SWAT '96 Proceedings, LNCS 1097, 1996, pages 198–211.

Abstract: We give a number of new lower bounds in the cell probe model with logarithmic cell size, which entails the same bounds on the random access computer with logarithmic word size and unit cost operations.

We study the signed prefix sum problem: given a string of length n of zeroes and signed ones, compute the sum of its ith prefix during updates. We show a lower bound of $\Omega(\log n/\log\log n)$ time per operations, even if the prefix sums are bounded by $\log n/\log\log n$ during all updates. We also show that if the update time is bounded by the product of the worst-case update time and the answer to the query, then the update time must be $\Omega\big(\sqrt{(\log n/\log\log n)}\big)$.

These results allow us to prove lower bounds for a variety of seemingly unrelated 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 the dynamic transitive closure problem 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 $\Omega\big(\sqrt{(\log n/\log\log n)}\big)$ for the dynamic membership problem of any Dyck language with two or more letters. This implies the same lower bound for the dynamic word problem for the free group with k generators. We also give lower bounds for the dynamic prefix majority and prefix equality problems.

[RS-96-8] *PostScript*, *DVI*.

Martin Hansen, Hans Httel, and Josva Kleist.

Bisimulations for Asynchronous Mobile Processes.

April 1996.

18 pp. Abstract appears in BRICS Notes Series NS-94-6, pages 188–189. Appears in *Tbilisi Symposium on Language, Logic, and Computation* (Tbilisi, Republic of Georgia, October 19–22, 1995).

Abstract: Within the past few years there has been renewed interest in the study of value-passing process calculi as a consequence of the emergence of the π-calculus. Here, Milner et. al have determined two variants of the notion of *bisimulation*, *late* and *early* bisimilarity. Most recently Sangiorgi has proposed the new notion of *open* bisimulation equivalence.

In this paper we consider Plain LAL, a mobile process calculus which differs from the π-calculus in the sense that the communication of data values happens *asynchronously.* The surprising result is that in the presence of asynchrony, the open, late and early bisimulation equivalences *coincide* – this in contrast to the π–calculus where they are distinct. The result allows us to for-

mulate a common *equational theory* which is sound and complete for finite terms of Plain LAL.

[RS-96-7] *PostScript*, *DVI*.
Ivan B. Damgrd and Ronald Cramer.
Linear Zero-Knowledge - A Note on Efficient Zero-Knowledge Proofs and Arguments.
April 1996.
17 pp. Appears in *The Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97 Proceedings, 1997, pages 436–445.

Abstract: We present a zero-knowledge proof system for any NP language L, which allows showing that $x \in L$ with error probability less than 2^{-k} using communication corresponding to $O(|x|^c)+k$ *bit commitments*, where c is a constant depending only on L. The proof can be based on any bit commitment scheme with a particular set of properties. We suggest an efficient implementation based on factoring.

We also present a 4-move perfect zero-knowledge interactive argument for any NP-language L. On input $x \in L$, the communication complexity is $O(|x|^c)max(k,l)$ *bits*, where l is the security parameter for the prover (The meaning of l is that if the prover is unable to solve an instance of a hard problem of size l before the protocol is finished, he can cheat with probability at most 2^{-k}). Again, the protocol can be based on any bit commitment scheme with a particular set of properties. We suggest efficient implementations based on discrete logarithms or factoring.

We present an application of our techniques to multiparty computations, allowing for example t committed oblivious transfers with error probability 2^{-k} to be done simultaneously using $O(t+k)$ commitments. Results for general computations follow from this.

As a function of the security parameters, our protocols have the smallest known asymptotic communication complexity among general proofs or arguments for NP. Moreover, the constants involved are small enough for the protocols to be practical in a realistic situation: both protocols are based on a Boolean formula Φ containing and-, or- and not-operators which verifies an NP-witness of membership in L. Let n be the number of times this formula reads an input variable. Then the communication complexity of the protocols when using our concrete commitment schemes can be more precisely stated as at most $4n+k+1$ commitments for the interactive proof and at most $5nl+5l$ bits for the argument (assuming $k \leq l$). Thus, if we use $k=n$, the number of commitments required for the proof is linear in n.

[RS-96-6] *PostScript*, *DVI*.
Mayer Goldberg.
An Adequate Left-Associated Binary Numeral System in the λ-Calculus (Revised Version).

March 1996.
19 pp. Appears in *Journal of Functional Programming* 10(6):607–623, 2000. This report is a revision of the BRICS Report RS-95-42.
Abstract: This paper introduces a sequence of λ-expressions modelling the binary expansion of integers. We derive expressions computing the test for zero, the successor function, and the predecessor function, thereby showing the sequence to be an adequate numeral system. These functions can be computed efficiently. Their complexity is independent of the order of evaluation.

[RS-96-5] *PostScript*, *DVI*.
Mayer Goldberg.
Gdelisation in the λ-Calculus (Extended Version).
March 1996.
10 pp Appears in *Information Processing Letters* 75(1–2):13–16, 2000. Earlier version Also available as BRICS Report RS-95-38.
Abstract: Gdelisation is a meta-linguistic encoding of terms in a language. While it is impossible to define an operator in the λ-calculus which encodes all closed λ-expressions, it is possible to construct restricted versions of such an encoding operator modulo normalisation. In this paper, we propose such an encoding operator for proper combinators.

[RS-96-4] *PostScript*, *DVI*.
Jrgen H. Andersen, Ed Harcourt, and K. V. S. Prasad.
A Machine Verified Distributed Sorting Algorithm.
February 1996.
21 pp. Abstract appeared in Bjerner, Larsson and Nordstrm, editors, *7th Nordic Workshop on Programming Theory*, NWPT '7 Proceedings, Programming Methodology Group's Report Series Report 86, January 1996, 1995, pages 62–82.
Abstract: We present a verification of a distributed sorting algorithm in ALF, an implementation of Martin Lf's type theory. The implementation is expressed as a program in a priortized version of CBS, (the Calculus of Broadcasting Systems) which we have implemented in ALF. The specification is expressed in terms of an ALF type which represents the set of all sorted lists and an HML (Hennesey–Milner Logic) formula which expresses that the sorting program will input any number of data until it hears a value triggering the program to begin outputting the data in a sorted fashion. We gain expressive power from the type theory by inheriting the language of data, state expressions, and propositions.

[RS-96-3] *PostScript*, *DVI*.
Jaap van Oosten.
The Modified Realizability Topos.
February 1996.
17 pp. Appears in *Journal of Pure and Applied Algebra*, 116:273–289, 1997 (the special Freyd issue).

Abstract: The modified realizability topos is the semantic (and higher order) counterpart of a variant of Kreisel's modified realizability (1957). These years, this realizability has been in the limelight again because of its possibilities for modelling type theory (Streicher, Hyland-Ong-Ritter) and strong normalization.

In this paper this topos is investigated from a general logical and topos-theoretic point of view. It is shown that Mod (as we call the topos) is the *closed complement* of the effective topos inside another one; this turns out to have some logical consequences. Some important subcategories of Mod are described, and a general logical principle is derived, which holds in the larger topos and implies the well-known Independence of Premiss principle.

[RS-96-2] *PostScript*, *DVI*.
Allan Cheng and Mogens Nielsen.
Open Maps, Behavioural Equivalences, and Congruences.
January 1996.
25 pp. A short version of this paper appeared in Kirchner, editor, *Trees in Algebra and Programming: 21st International Colloquium*, CAAP '96 Proceedings, LNCS 1059, 1996, pages 257–272, and a full version appears in *Theoretical Computer Science*, 190(1):87–112, January 1998.

Abstract: Spans of open maps have been proposed by Joyal, Nielsen, and Winskel as a way of adjoining an abstract equivalence, $\mathcal{P}$-bisimilarity, to a category of models of computation $\mathcal{M}$, where $\mathcal{P}$ is an arbitrary subcategory of observations. Part of the motivation was to recast and generalise Milner's well-known strong bisimulation in this categorical setting. An issue left open was the *congruence properties* of $\mathcal{P}$-bisimilarity. We address the following fundamental question: given a category of models of computation $\mathcal{M}$ and a category of observations $\mathcal{P}$, are there any conditions under which algebraic constructs viewed as functors preserve $\mathcal{P}$-bisimilarity? We define the notion of functors being $\mathcal{P}$*-factorisable*, show how this ensures that $\mathcal{P}$-bisimilarity is a congruence with respect to such functors. Guided by the definition of $\mathcal{P}$-factorisability we show how it is possible to parametrise proofs of functors being $\mathcal{P}$-factorisable with respect to the category of observations $\mathcal{P}$, i.e., with respect to a behavioural equivalence.

[RS-96-1] *PostScript*, *DVI*.
Gerth Stlting Brodal and Thore Husfeldt.
A Communication Complexity Proof that Symmetric Functions have Logarithmic Depth.
January 1996.
3 pp.

Abstract: We present a direct protocol with logarithmic communication that finds an element in the symmetric difference of two sets of different size. This yields a simple proof that symmetric functions have logarithmic circuit depth.