BRICS Research Series, Abstracts, 1998

August 5, 2003

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

References

[RS-98-55] *PostScript*, *PDF*, *DVI*.
Andrew D. Gordon, Paul D. Hankin, and Sren 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 *Journal of Functional Programming*, 9(4):373-427, 1999, and Ramesh and Sivakumar, editors, *Foundations of Software Technology and Theoretical Computer Science: 17th Conference*, FST&TCS '97 Proceedings, LNCS 1346, 1997, pages 74–87.
Abstract: We adopt the untyped imperative object calculus of Abadi and Cardelli as a minimal setting in which to study problems of compilation and program equivalence that arise when compiling object-oriented languages. We present both a big-step and a small-step substitution-based operational semantics for the calculus. Our first two results are theorems asserting the equivalence of our substitution-based semantics with a closure-based semantics like that given by Abadi and Cardelli. Our third result is a direct proof of the correctness of compilation to a stack-based abstract machine via a small-step decompilation algorithm. Our fourth result is that contextual equivalence of objects coincides with a form of Mason and Talcott's CIU equivalence; the latter provides a tractable means of establishing operational equivalences. Finally, we prove correct an algorithm, used in our prototype compiler, for statically resolving method offsets. This is the first study of correctness of an object-oriented abstract machine, and of operational equivalence for the imperative object calculus.

[RS-98-54] 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 This report is superseded by the later report BRICS RS-99-27.

Abstract: Lambda-lifting a block-structured program transforms it into a set of recursive equations. We present the symmetric transformation: lambda-dropping. Lambda-dropping a set of recursive equations restores block structure and lexical scope.

For lack of block structure and lexical scope, recursive equations must carry around all the parameters that any of their callees might possibly need. Both lambda-lifting and lambda-dropping thus require one to compute Def/Use paths:

- for lambda-lifting: each of the functions occurring in the path of a free variable is passed this variable as a parameter;
- for lambda-dropping: parameters which are used in the same scope as their definition do not need to be passed along in their path.

A program whose blocks have no free variables is scope-insensitive. Its blocks are then free to float (for lambda-lifting) or to sink (for lambda-dropping) along the vertices of the scope tree.

Our primary application is partial evaluation. Indeed, many partial evaluators for procedural programs operate on recursive equations. To this end, they lambda-lift source programs in a pre-processing phase. But often, partial evaluators [automatically] produce residual recursive equations with dozens of parameters, which most compilers do not handle efficiently. We solve this critical problem by lambda-dropping residual programs in a post-processing phase, which significantly improves both their compile time and their run time.

Lambda-lifting has been presented as an intermediate transformation in compilers for functional languages. We study lambda-lifting and lambda-dropping per se, though lambda-dropping also has a use as an intermediate transformation in a compiler: we noticed that lambda-dropping a program corresponds to transforming it into the functional representation of its optimal SSA form. This observation actually led us to substantially improve our PEPM '97 presentation of lambda-dropping.

[RS-98-53] *PostScript*, *PDF*.
Julian C. Bradfield.
Fixpoint Alternation: Arithmetic, Transition Systems, and the Binary Tree.
December 1998.
20 pp. Appears in *Theoretical Informatics and Applications*, 33:341–356, 1999.

Abstract: We provide an elementary proof of the fixpoint alternation hierarchy in arithmetic, which in turn allows us to simplify the proof of the modal mu-calculus alternation hierarchy. We further show that the alternation hierarchy on the binary tree is strict, resolving a problem of Niwiński.

[RS-98-52] *PostScript*, *PDF*, *DVI*.
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.

Abstract: An interpretation of Abadi and Cardelli's first-order *Imperative Object Calculus* into a typed π-calculus is presented. The interpretation validates the subtyping relation and the typing judgements of the Object Calculus, and is computationally adequate. The proof of computational adequacy makes use of (a π-calculus version) of *ready simulation*, and of a *factorisation* of the interpretation into a functional part and a very simple imperative part. The interpretation can be used to compare and contrast the Imperative and the *Functional* Object Calculi, and to prove properties about them, within a unified framework.

[RS-98-51] *PostScript*, *PDF*.
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.

Abstract: This paper describes ongoing work on the automatic construction of formal models from Real-Time implementations. The model construction is based on measurements of the timed behavior of the threads of an implementation, their causal interaction patterns and external visible events. A specification of the timed behavior is modeled in timed automata and checked against the generated model in order to validate their timed behavior.

[RS-98-50] *PostScript*, *PDF*, *DVI*.
Luca Aceto and Anna Inglfsdttir.
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.

Abstract: This study offers a characterization of the collection of properties expressible in Hennessy-Milner Logic (HML) with recursion that can be tested using finite LTSs. In addition to actions used to probe the behaviour of the tested system, the LTSs that we use as tests will be able to perform a distinguished action `nok` to signal their dissatisfaction during the interaction with the tested process. A process s *passes the test* T iff T does not perform the action `nok` when it interacts with s. A test T tests for a property ϕ in HML with recursion iff it is passed by exactly the states that satisfy ϕ. The paper gives an expressive completeness result offering a characterization of the collection of properties in HML with recursion that are testable in the above sense.

[RS-98-49] *PostScript*, *PDF*, *DVI*.
Luca Aceto, Willem Jan Fokkink, and Anna Inglfsdttir.
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.
Abstract: Prefix iteration is a variation on the original binary version of the Kleene star operation P^*Q, obtained by restricting the first argument to be an atomic action, and yields simple iterative behaviours that can be equationally characterized by means of finite collections of axioms. In this paper, we present axiomatic characterizations for a significant fragment of the notions of equivalence and preorder in van Glabbeek's linear-time/branching-time spectrum over Milner's basic CCS extended with prefix iteration. More precisely, we consider ready simulation, simulation, readiness, trace and language semantics, and provide complete (in)equational axiomatizations for each of these notions over BCCS with prefix iteration. All of the axiom systems we present are finite, if so is the set of atomic actions under consideration.

[RS-98-48] *PostScript*, *PDF*, *DVI*.
Luca Aceto, Patricia Bouyer, Augusto Burgueo, 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.
Abstract: In this paper we provide a complete characterization of the class of properties of (networks of) timed automata for which model checking can be reduced to reachability checking in the context of testing automata.

[RS-98-47] *PostScript*, *PDF*.
Gerd Behrmann, Kim G. Larsen, Justin Pearson, Carsten Weise, and Wang Yi.
Efficient Timed Reachability Analysis using Clock Difference Diagrams.
December 1998.
13 pp. Appears in Halbwachs and Peled, editors, *Computer-Aided Verification: 11th International Conference*, CAV '99 Proceedings, LNCS 1633, 1999, pages 341–353.
Abstract: One of the major problems in applying automatic verification tools to industrial-size systems is the excessive amount of memory required during the state-space exploration of a model. In the setting of real-time, this problem of state-explosion requires extra attention as information must be kept not only on the discrete control structure but also on the values of continuous clock variables.

In this paper, we present Clock Difference Diagrams, CDD's, a BDD-like data-structure for representing and effectively manipulating certain non-convex subsets of the Euclidean space, notably those encountered during verification of timed automata.

A version of the real-time verification tool UPPAAL using CDD's as a compact data-structure for storing explored symbolic states has been implemented. Our experimental results demonstrate significant space-savings: for 8 industrial examples, the savings are between 46% and 99% with moderate increase in runtime.

We further report on how the symbolic state-space exploration itself may be carried out using CDD's.

[RS-98-46] *PostScript*, *PDF*.
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, Turku Centre for Computer Science TUCS General Publications 11, 1998. Appears in *Nordic Journal of Computing*, 6(3):271–298, 1999.

Abstract: We sketch a BDD-like structure for representing unions of simple convex polyhedra, describing the legal values of a set of clocks given bounds on the values of clocks and clock differences.

[RS-98-45] *PostScript*, *PDF*.
Morten Vadskr 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.

Abstract: We present the design and implementation of a high performance layered video codec, designed for deployment in bandwidth heterogeneous networks. The codec combines wavelet based subband decomposition and discrete cosine transforms to facilitate layered spatial and SNR (signal-to-noise ratio) coding for bit-rate adaption to a wide range of receiver capabilities. We show how a test video stream can be partitioned into several distinct layers of increasing visual quality and bandwidth requirements, with the difference between highest and lowest requirement being 47 : 1.

Through the use of the Visual Instruction Set on SUN's UltraSPARC platform we demonstrate how SIMD parallel image processing enables real-time layered encoding and decoding in software. Our $384 \times 320 \times 24$-bit test video stream is partitioned into 21 layers at a speed of 39 frames per second and reconstructed at 28 frames per second. Our VIS accelerated encoder stages are about 3-4 times as fast as an optimized C version. We find that this speedup is well worth the extra implementation effort.

[RS-98-44] *PostScript*, *PDF*.

Brian Nielsen and Gul Agha.
Towards Re-usable Real-Time Objects.
December 1998.
36 pp. Appears in *The Annals of Software Engineering*, 7:257–282, 1999.

Abstract: Large and complex real-time systems can benefit significantly from a component-based development approach where new systems are constructed by composing reusable, documented and previously tested concurrent objects. However, reusing objects which execute under real-time constraints is problematic because application specific time and synchronization constraints are often embedded in the internals of these objects. The tight coupling of functionality and real-time constraints makes objects interdependent, and as a result difficult to reuse in another system.

We propose a model which facilitates separate and modular specification of real-time constraints, and show how separation of real-time constraints and functional behavior is possible. We present our ideas using the *Actor model* to represent untimed objects, and the *Real-time Synchronizers* language to express real-time and synchronization constraints. We discuss specific mechanisms by which *Real-time Synchronizers* can govern the interaction and execution of untimed objects.

We treat our model formally, and succinctly define what effect real-time constraints have on a set of concurrent objects. We briefly discuss how a middleware scheduling and event-dispatching service can use the synchronizers to execute the system.

[RS-98-43] *PostScript*, *PDF*, *DVI.*
Peter D. Mosses.
CASL: A Guided Tour of its Design.
December 1998.
31 pp. Appears in Fiadeiro, editor, *Recent Trends in Algebraic Development Techniques: 13th Workshop*, WADT '98 Selected Papers, LNCS 1589, 1999, pages 216–240.

Abstract: CASL is an expressive language for the specification of functional requirements and modular design of software. It has been designed by CoFI, the international Common Framework Initiative for algebraic specification and development. It is based on a critical selection of features that have already been explored in various contexts, including subsorts, partial functions, first-order logic, and structured and architectural specifications. CASL should facilitate interoperability of many existing algebraic prototyping and verification tools.

This guided tour of the CASL design is based closely on a 1/2-day tutorial held at ETAPS '98 (corresponding slides are available from the CoFI archives). The major issues that had to be resolved in the design process are indicated, and all the main concepts and constructs of CASL are briefly explained and

illustrated—the reader is referred to the CASL Language Summary for further details. Some familiarity with the fundamental concepts of algebraic specification would be advantageous.

[RS-98-42] *PostScript*, *PDF*, *DVI.*
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.

Abstract: A complete formal semantic description of a practical programming language (such as Java) is likely to be a lengthy document, regardless of which semantic framework is being used. Good modularity of the description is important to the person(s) developing it, to facilitate reuse, change, and extension. Unfortunately, the conventional versions of the major semantic frameworks have rather poor modularity.

In this paper, we first recall some approaches that improve the modularity of denotational semantics, namely *action semantics*, *modular monadic semantics*, and a hybrid framework that combines these: *modular monadic action semantics.* We then address the issue of modularity in *operational semantics*, which appears to have received comparatively little attention so far, and report on some preliminary investigations of how one might achieve the same kind of modularity in structural operational semantics as the use of monad transformers can provide in denotational semantics—this is the main technical contribution of the paper. Finally, we briefly consider the representation of structural operational semantics in rewriting logic, and speculate on the possibility of using it to interpret programs in the described language. Providing powerful meta-tools for such semantics-based interpretation is an interesting potential application of rewriting logic; good modularity of the semantic descriptions may be crucial for the practicality of using the tools.

Much of the paper consists of (very) simple examples of semantic descriptions in the various frameworks, illustrating the degree of reformulation needed when extending the described language—a strong indicator of modularity. Throughout, it is assumed that the reader has some familiarity with the concepts and notation of denotational and structural operational semantics. Familiarity with the basic notions of monads and monad transformers is not a prerequisite.

[RS-98-41] *PostScript*, *PDF*, *DVI.*
Ulrich Kohlenbach.
The Computational Strength of Extensions of Weak Knig's Lemma.
December 1998.
23 pp.

Abstract: The weak Knig's lemma WKL is of crucial significance in the study of fragments of mathematics which on the one hand are

mathematically strong but on the other hand have a low proof-theoretic and computational strength. In addition to the restriction to binary trees (or equivalently bounded trees), WKL is also 'weak' in that the tree predicate is quantifier-free. Whereas in general the computational and proof-theoretic strength increases when logically more complex trees are allowed, we show that this is not the case for trees which are given by formulas in a class Φ_∞ where we allow an arbitrary function quantifier prefix over bounded functions in front of a Π^0_1-formula. This results in a schema Φ_∞-WKL.
Another way of looking at WKL is via its equivalence to the principle

$$\forall x \exists y \leq 1 \forall z\, A_0(x,y,z) \rightarrow \exists f \leq \lambda x.1 \forall x,z\, A_0(x,fx,z),$$

where A_0 is a quantifier-free formula (x,y,z are natural number variables). We generalize this to Φ_∞-formulas as well and allow function quantifiers '$\exists g \leq s$' instead of '$\exists y \leq 1$', where $g \leq s$ is defined pointwise. The resulting schema is called Φ_∞-b-AC^{0,1}.
In the absence of functional parameters (so in particular in a second order context), the corresponding versions of Φ_∞-WKL and Φ_∞-b-AC^{0,1} turn out to be equivalent to WKL. This changes completely in the presence of functional variables of type 2 where we get proper hierarchies of principles Φ_n-WKL and Φ_n-b-AC^{0,1}. Variables of type 2 however are necessary for a direct representation of analytical objects and – sometimes – for a faithful representation of such objects at all as we will show in a subsequent paper. By a reduction of Φ_∞-WKL and Φ_∞-b-AC^{0,1} to a non-standard axiom F (introduced in a previous paper) and a new elimination result for F relative to various fragment of arithmetic in all finite types, we prove that Φ_∞-WKL and Φ_∞-b-AC^{0,1} do neither contribute to the provably recursive functionals of these fragments nor to their proof-theoretic strength. In a subsequent paper we will illustrate the greater mathematical strength of these principles (compared to WKL).

[RS-98-40] *PostScript*, *PDF*, *DVI*.
Henrik Reif Andersen, Colin Stirling, and Glynn Winskel.
A Compositional Proof System for the Modal μ-Calculus.
December 1998.
29 pp. Extended abstract appears in *Ninth Annual IEEE Symposium on Logic in Computer Science*, LICS '94 Proceedings, 1994, pages 144–153. Superseeds the earlier BRICS Report RS-94-34.

Abstract: We present a proof system for determining satisfaction between processes in a fairly general process algebra and assertions of the modal μ-calculus. The proof system is compositional in the structure of processes. It extends earlier work on compositional reasoning within the modal μ-calculus and combines it with techniques from work on local model checking. The proof system is sound for all processes and complete for a class of finite-state processes.

[RS-98-39] *PostScript*, *PDF*, *DVI*.
Daniel Fridlender.
An Interpretation of the Fan Theorem in Type Theory.
December 1998.
15 pp. Appears in Altenkirch, Naraschewski and Reus, editors, *International Workshop on Types for Proofs and Programs 1998*, TYPES '98 Selected Papers, LNCS 1657, 1999, pages 93–105.

Abstract: This article presents a formulation of the fan theorem in Martin-Lf's type theory. Starting from one of the standard versions of the fan theorem we gradually introduce reformulations leading to a final version which is easy to interpret in type theory. Finally we describe a formal proof of that final version of the fan theorem.

[RS-98-38] *PostScript*, *PDF*, *DVI*.
Daniel Fridlender and Mia Indrika.
An n-ary `zipWith` *in Haskell.*
December 1998.
12 pp.

Abstract: The aim of this note is to present an alternative definition of the `zipWith` family in the Haskell Library Report. Because of the difficulties in defining a well-typed function with a variable number of arguments, the library presents a family of `zipWith` functions. It provides zip functions $\mathtt{zipWith}_2, \mathtt{zipWith}_3, \ldots, \mathtt{zipWith}_7$. For each n, $\mathtt{zipWith}_n$ zips n lists with a n-ary function. Defining a single `zipWith` function with a variable number of arguments seems to require dependent types. We show, however, how to define such a function in Haskell by means of a binary operator for grouping its arguments. For comparison, we also give definitions of `zipWith` in languages with dependent types.

[RS-98-37] *PostScript*, *PDF*, *DVI*.
Ivan B. Damgrd, Joe Kilian, and Louis Salvail.
On the (Im)possibility of Basing Oblivious Transfer and Bit Commitment on Weakened Security Assumptions.
December 1998.
22 pp. Appears in Stern, editor, *Advances in Cryptology: International Conference on the Theory and Application of Cryptographic Techniques*, EUROCRYPT '99 Proceedings, LNCS 1592, 1999, pages 56–73.

Abstract: We consider the problem of basing Oblivious Transfer (OT) and Bit Commitment (BC), with information theoretic security, on seemingly weaker primitives. We introduce a general model for describing such primitives, called *Weak Generic Transfer* (WGT). This model includes as important special cases *Weak Oblivious Transfer* (WOT), where both the sender and receiver may learn too much about the other party's input, and a new, more realistic model of noisy channels, called *unfair noisy channels.* An unfair noisy channel has a known range of possible noise levels; protocols must work for any level within this range against

adversaries who know the actual noise level.

We give a precise characterization for when one can base OT on WOT. When the deviation of the WOT from the ideal is above a certain threshold, we show that no information-theoretic reductions from OT (even against passive adversaries) and BC exist; when the deviation is below this threshold, we give a reduction from OT (and hence BC) that is information-theoretically secure against active adversaries.

For unfair noisy channels we show a similar threshold phenomenon for bit commitment. If the upper bound on the noise is above a threshold (given as function of the lower bound) then no information-theoretic reduction from OT (even against passive adversaries) or BC exist; when it is below this threshold we give a reduction from BC. As a partial result, we give a reduction from OT to UNC for smaller noise intervals.

[RS-98-36] *PostScript*, *PDF*, *DVI*.
Ronald Cramer, Ivan B. Damgrd, Stefan Dziembowski, Martin Hirt, and Tal Rabin.
Efficient Multiparty Computations with Dishonest Minority.
December 1998.
19 pp. Appears in Stern, editor, *Advances in Cryptology: International Conference on the Theory and Application of Cryptographic Techniques*, EUROCRYPT '99 Proceedings, LNCS 1592, 1999, pages 311–326.

Abstract: We consider verifiable secret sharing (VSS) and multiparty computation (MPC) in the secure channels model, where a broadcast channel is given and a non-zero error probability is allowed. In this model Rabin and Ben-Or proposed VSS and MPC protocols, secure against an adversary that can corrupt any minority of the players. In this paper, we first observe that a subprotocol of theirs, known as weak secret sharing (WSS), is not secure against an adaptive adversary, contrary to what was believed earlier. We then propose new and adaptively secure protocols for WSS, VSS and MPC that are substantially more efficient than the original ones. Our protocols generalize easily to provide security against general Q^2 adversaries.

[RS-98-35] 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.

Abstract: We explore the hierarchy of control induced by successive transformations into continuation-passing style (CPS) in the presence of “control delimiters” and “composable continuations”. Specifically, we investigate the structural operational semantics associated with the CPS hierarchy.

To this end, we characterize an operational notion of continuation semantics. We relate it to the traditional CPS transformation and we use it to account for the control operator shift and the control delimiter reset operationally. We then transcribe the resulting continuation semantics in ML, thus obtaining a native and modular implementation of the entire hierarchy. We illustrate it with a few examples, the most significant of which is layered monads.

[RS-98-34] *PostScript*, *PDF*, *DVI*.
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.
Abstract: We prove that at least $\frac{3k-4}{k(2k-3)}\binom{n}{2} - \mathrm{O}(k)$ equivalence tests and no more than $\frac{2}{k}\binom{n}{2} + \mathrm{O}(n)$ equivalence tests are needed in the worst case to identify the equivalence classes with at least k members in set of n elements. The upper bound is an improvement by a factor 2 compared to known results. For $k = 3$ we give tighter bounds. Finally, for $k > \frac{n}{2}$ we prove that it is necessary and it suffices to make $2n - k - 1$ equivalence tests which generalizes a known result for $k = \lceil \frac{n+1}{2} \rceil$.

[RS-98-33] *PostScript*, *PDF*, *DVI*.
Hans Httel, Josva Kleist, Uwe Nestmann, and Massimo Merro.
Migration = Cloning ; Aliasing (Preliminary Version).
December 1998.
40 pp. Appears in Cardelli, editor, *Foundations of Object-Oriented Languages: 6th International Conference*, FOOL6 Informal Proceedings, 1999.
Abstract: In Obliq, a lexically scoped, distributed, object-based programming language, object migration was suggested as creating a (remote) copy of an object's state at the target site, followed by turning the (local) object itself into an alias, also called *surrogate*, for the just created remote copy.

We consider the creation of object surrogates as an abstraction of the above-mentioned style of migration. We introduce jeblik, a distribution-free subset of Obliq, and provide two formal semantics, one in an intuitive configuration style, the other in terms of a pi-calculus. The intuitive semantics shows why surrogation is neither safe in Obliq, nor can it be so in full generality in Repliq (a repaired Obliq). The pi-calculus semantics allows us to prove that surrogation in jeblik is safe for certain well-identified cases, thus suggesting that migration in Repliq may be safe, accordingly.

[RS-98-32] *PostScript*, *PDF*, *DVI*.
Jan Camenisch and Ivan B. Damgrd.
Verifiable Encryption and Applications to Group Signatures and Signature Sharing.

December 1998.
18 pp. Appears in Okamoto, editor, *Advances in Cryptology: Sixth ASIACRYPT Conference on the Theory and Applications of Cryptologic Techniques*, ASIACRYPT '00 Proceedings, LNCS 1976, 2000, pages 331–345.

Abstract: We generalise and improve the security and efficiency of the verifiable encryption scheme of Asokan et al., such that it can rely on more general assumptions, and can be proven secure without relying on random oracles. We show a new application of verifiable encryption to group signatures with separability, these schemes do not need special purpose keys but can work with a wide range of signature and encryption schemes already in use. Finally, we extend our basic primitive to verifiable threshold and group encryption. By encrypting digital signatures this way, one gets new solutions to the verifiable signature sharing problem.

[RS-98-31] *PostScript*, *PDF*, *DVI*.
Glynn Winskel.
A Linear Metalanguage for Concurrency.
November 1998.
21 pp. Appears in Haeberer, editor, *Algebraic Methodology and Software Technology: 7th International Conference*, AMAST '98 Proceedings, LNCS 1548, 1999, pages 42–58.

Abstract: A metalanguage for concurrent process languages is introduced. Within it a range of process languages can be defined, including higher-order process languages where processes are passed and received as arguments. (The process language has, however, to be linear, in the sense that a process received as an argument can be run at most once, and not include name generation as in the Pi-Calculus.) The metalanguage is provided with two interpretations both of which can be understood as categorical models of a variant of linear logic. One interpretation is in a simple category of nondeterministic domains; here a process will denote its set of traces. The other interpretation, obtained by direct analogy with the nondeterministic domains, is in a category of presheaf categories; the nondeterministic branching behaviour of a process is captured in its denotation as a presheaf. Every presheaf category possesses a notion of (open-map) bisimulation, preserved by terms of the metalanguage. The conclusion summarises open problems and lines of future work.

[RS-98-30] *PostScript*, *PDF*, *DVI*.
Carsten Butz.
Finitely Presented Heyting Algebras.
November 1998.
30 pp.

Abstract: In this paper we study the structure of finitely presented Heyting algebras. Using algebraic techniques (as opposed to techniques from proof-theory) we show that every such Heyting algebra is in fact co-Heyting, improving on a result of Ghilardi who showed that Heyting algebras free on a finite set of generators

are co-Heyting. Along the way we give a new and simple proof of the finite model property.

Our main technical tool is a representation of finitely presented Heyting algebras in terms of a colimit of finite distributive lattices. As applications we construct explicitly the minimal join-irreducible elements (the atoms) and the maximal join-irreducible elements of a finitely presented Heyting algebras in terms of a given presentation. This gives as well a new proof of the disjunction property for intuitionistic propositional logic.

[RS-98-29] *PostScript*, *PDF*, *DVI*.
Jan Camenisch and Markus Michels.
Proving in Zero-Knowledge that a Number is the Product of Two Safe Primes.
November 1998.
19 pp. Appears in Stern, editor, *Advances in Cryptology: International Conference on the Theory and Application of Cryptographic Techniques*, EUROCRYPT '99 Proceedings, LNCS 1592, 1999, pages 106–121.

Abstract: This paper presents the first efficient statistical zero-knowledge protocols to prove statements such as:

- A committed number is a pseudo-prime.
- A committed (or revealed) number is the product of two safe primes, i.e., primes p and q such that $(p-1)/2$ and $(q-1)/2$ are primes as well.
- A given value is of large order modulo a composite number that consists of two safe prime factors.

So far, no methods other than inefficient circuit-based proofs are known for proving such properties. Proving the second property is for instance necessary in many recent cryptographic schemes that rely on both the hardness of computing discrete logarithms and of difficulty computing roots modulo a composite.

The main building blocks of our protocols are statistical zero-knowledge proofs that are of independent interest. Mainly, we show how to prove the correct computation of a modular addition, a modular multiplication, or a modular exponentiation, where all values including the modulus are committed but *not* publicly known. Apart from the validity of the computation, no other information about the modulus (e.g., a generator which order equals the modulus) or any other operand is given. Our technique can be generalized to prove in zero-knowledge that any multivariate polynomial equation modulo a certain modulus is satisfied, where only commitments to the variables of the polynomial and a commitment to the modulus must be known. This improves previous results, where the modulus is publicly known.

We show how a prover can use these building blocks to convince a verifier that a committed number is prime. This finally leads to efficient protocols for proving that a committed (or revealed) number is the product of two safe primes. As a consequence, it can be shown that a given value is of large order modulo a given number that is a product of two safe primes.

[RS-98-28] *PostScript*, *PDF*.
Rasmus Pagh.
Low Redundancy in Dictionaries with $O(1)$ Worst Case Lookup Time.
November 1998.
15 pp. Journal version in *SIAM Journal on Computing* 31:353–363, 2001, and proceedings version in van Emde Boas, Wiedermann and Nielsen, editors, *26th International Colloquium on Automata, Languages, and Programming*, ICALP '99 Proceedings, LNCS 1644, 1999, pages 595–604.
Abstract: A *static dictionary* is a data structure for storing subsets of a finite universe U, so that membership queries can be answered efficiently. We study this problem in a unit cost RAM model with word size $\Omega(\log|U|)$, and show that for n-element subsets, constant worst case query time can be obtained using $B+O(\log\log|U|)+o(n)$ bits of storage, where $B=\lceil\log_2\binom{|U|}{n}\rceil$ is the minimum number of bits needed to represent all such subsets. The solution for dense subsets uses $B+O(\frac{|U|\log\log|U|}{\log|U|})$ bits of storage, and supports constant time rank queries. In a dynamic setting, allowing insertions and deletions, our techniques give an $O(B)$ bit space usage.

[RS-98-27] *PostScript*, *PDF*, *DVI*.
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.
Abstract: The concept of group signatures allows a group member to sign messages anonymously on behalf of the group. However, in the case of a dispute, the identity of a signature's originator can be revealed by a designated entity. In this paper we propose a new group signature scheme that is well suited for large groups, i.e., the length of the group's public key and of signatures do not depend on the size of the group. Our scheme is based on a variation of the RSA problem called strong RSA assumption. It is also more efficient than previous ones satisfying these requirements.

[RS-98-26] *PostScript*, *PDF*, *DVI*.
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.

Abstract: This paper is about the encoding of $p\pi$, the polyadic π-calculus, in $m\pi$, the monadic π-calculus. A type system for $m\pi$ processes is introduced that captures the interaction regime underlying the encoding of $p\pi$ processes respecting a sorting. A full-abstraction result is shown: two $p\pi$ processes are typed barbed congruent iff their $m\pi$ encodings are monadic-typed barbed congruent.

[RS-98-25] *PostScript*, *PDF*, *DVI*.
Devdatt P. Dubhashi.
Talagrand's Inequality in Hereditary Settings.
October 1998.
22 pp.

Abstract: We develop a nicely packaged form of Talagrand's inequality that can be applied to prove concentration of measure for functions defined by hereditary properties. We illustrate the framework with several applications from combinatorics and algorithms. We also give an extension of the inequality valid in spaces satisfying a certain negative dependence property and give some applications.

[RS-98-24] *PostScript*, *PDF*, *DVI*.
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.

Abstract: We illustrate the use of Talagrand's inequality and an extension of it to dependent random variables due to Marton for the analysis of distributed randomised algorithms, specifically, for edge colouring graphs.

[RS-98-23] *PostScript*, *PDF*, *DVI*.
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.

Abstract: We use Martingale inequalities to give a simple and uniform analysis of two families of distributed randomised algorithms for edge colouring graphs.

[RS-98-22] *PostScript*, *PDF*, *DVI*.
Gian Luca Cattani, John Power, and Glynn Winskel.
A Categorical Axiomatics for Bisimulation.
September 1998.
ii+21 pp. Appears in Sangiorgi and de Simone, editors, *Concurrency Theory: 9th International Conference*, CONCUR '98 Proceedings, LNCS 1466, 1998, pages 581–596.

Abstract: We give an axiomatic category theoretic account of bisimulation in process algebras based on the idea of functional bisimulations as open maps. We work with 2-monads, T, on **Cat**. Operations on processes, such as nondeterministic sum, prefixing and parallel composition are modelled using functors in the Kleisli category for the 2-monad T. We may define the notion of open map for any such 2-monad; in examples of interest, that agrees exactly with the usual notion of functional bisimulation. Under a condition on T, namely that it be a dense KZ-monad, which we define, it follows that functors in $Kl(T)$ preserve open maps, i.e., they respect functional bisimulation. We further investigate structures on $Kl(T)$ that exist for axiomatic reasons, primarily because T is a dense KZ-monad, and we study how those structures help to model operations on processes. We outline how this analysis gives ideas for modelling higher order processes. We conclude by making comparison with the use of presheaves and profunctors to model process calculi.

[RS-98-21] *PostScript*, *PDF*, *DVI*.
John Power, Gian Luca Cattani, and Glynn Winskel.
A Representation Result for Free Cocompletions.
September 1998.
16 pp. Appears in *Journal of Pure and Applied Algebra*, 151(3):273–286 (2000).

Abstract: Given a class F of weights, one can consider the construction that takes a small category C to the free cocompletion of C under weighted colimits, for which the weight lies in F. Provided these free F-cocompletions are small, this construction generates a 2-monad on **Cat**, or more generally on V-**Cat** for monoidal biclosed complete and cocomplete V. We develop the notion of a dense 2-monad on V-**Cat** and characterise free F-cocompletions by dense KZ-monads on V-**Cat**. We prove various corollaries about the structure of such 2-monads and their Kleisli 2-categories, as needed for the use of open maps in giving an axiomatic study of bisimulation in concurrency. This requires the introduction of the concept of a pseudo-commutativity for a strong 2-monad on a symmetric monoidal 2-category, and a characterisation of it in terms of structure on the Kleisli 2-category.

[RS-98-20] *PostScript*, *PDF*, *DVI*.
Sren Riis and Meera Sitharam.
Uniformly Generated Submodules of Permutation Modules.
September 1998.
35 pp. Appears in *Annals of Pure and Applied Algebra*, 160(3):285–318, 2001.

Abstract: This paper is motivated by a link between algebraic proof complexity and the representation theory of the finite symmetric groups. Our perspective leads to a series of non-traditional problems in the representation theory of S_n.

Most of our technical results concern the structure of “uniformly” generated submodules of permutation modules. We consider (for

example) sequences W_n of submodules of the permutation modules $M^{(n-k,1^k)}$ and prove that if the modules W_n are given in a uniform way - which we make precise - the dimension $p(n)$ of W_n (as a vector space) is a single polynomial with rational coefficients, for all but finitely many "singular" values of n. Furthermore, we show that $\dim(W_n) < p(n)$ for each singular value of $n \geq 4k$. The results have a non-traditional flavor arising from the study of the irreducible structure of the submodules W_n *beyond* isomorphism types.

We sketch the link between our structure theorems and proof complexity questions, which can be viewed as special cases of the famous NP vs. co-NP problem in complexity theory. In particular, we focus on the efficiency of proof systems for showing membership in polynomial ideals, for example, based on Hilbert's Nullstellensatz.

[RS-98-19] *PostScript*, *PDF*, *DVI*.

Sren Riis and Meera Sitharam.

Generating Hard Tautologies Using Predicate Logic and the Symmetric Group.

September 1998.

13 pp. Abstract appears in *Logic Journal of IGPL*, a special section devoted *the 5th Workshop on Logic, Language, Information and Computation*, WoLLIC '98 (So Paulo, Brazil, July 28–31, 1998), pages 936–937. Full version appears in *Logic Journal of the IGPL*, 8(6):787–795, 2000.

Abstract: We introduce methods to generate uniform families of hard propositional tautologies. The tautologies are essentially generated from a single propositional formula by a natural action of the symmetric group S_n.

The basic idea is that any Second Order Existential sentence Ψ can be systematically translated into a conjunction ϕ of a finite collection of clauses such that the models of size n of an appropriate Skolemization $\tilde{\Psi}$ are in one-to-one correspondence with the satisfying assignments to ϕ_n: the S_n-closure of ϕ, under a natural action of the symmetric group S_n. Each ϕ_n is a CNF and thus has depth at most 2. The size of the ϕ_n's is bounded by a polynomial in n. Under the assumption *NEXPTIME* $\neq$ *co-NEXPTIME,* for any such sequence ϕ_n for which the spectrum $S := \{n : \phi_n \text{ satisfiable}\}$ is *NEXPTIME*-complete, the tautologies $\neg\phi_{n \notin S}$ do not have polynomial length proofs in any propositional proof system.

Our translation method shows that most sequences of tautologies being studied in propositional proof complexity can be systematically generated from Second Order Existential sentences and moreover, many natural mathematical statements can be converted into sequences of propositional tautologies in this manner.

We also discuss algebraic proof complexity issues for such se-

quences of tautologies. To this end, we show that any Second Order Existential sentence Ψ can be systematically translated into a finite collection of polynomial equations $\bar{Q} = 0$ such that the models of size n of an appropriate skolemization $\tilde{\Psi}$ are in one-to-one correspondence with the solutions to $\bar{Q}_n = 0$: the S_n-closure of $\bar{Q} = 0$, under a natural action of the symmetric group S_n. The degree of $\bar{Q}_n$ is the same as that of $\bar{Q}$, and hence is independent of n, and the number of variables is no more than a polynomial in n. Furthermore, we briefly describe how, for the corresponding sequences of tautologies ϕ_n, the rich structure of the S_n closed, uniformly generated, algebraic systems $\bar{Q}_n$ has profound consequences on on the algebraic proof complexity of ϕ_n.

[RS-98-18] *PostScript*, *PDF*, *DVI.*
Ulrich Kohlenbach.
Things that can and things that can't be done in PRA.
September 1998.
24 pp. Appeared in *Annals of Pure and Applied Logic*, 102(3):223-245, 2000.

Abstract: It is well-known by now that large parts of (non-constructive) mathematical reasoning can be carried out in systems $\mathcal{T}$ which are conservative over primitive recursive arithmetic **PRA** (and even much weaker systems). On the other hand there are principles **S** of elementary analysis (like the Bolzano-Weierstra principle, the existence of a limit superior for bounded sequences etc.) which are known to be equivalent to arithmetical comprehension (relative to $\mathcal{T}$) and therefore go far beyond the strength of **PRA** (when added to $\mathcal{T}$).
In this paper we determine precisely the arithmetical and computational strength (in terms of optimal conservation results and subrecursive characterizations of provably recursive functions) of weaker function parameter-free schematic versions $\mathbf{S}^-$ of **S**, thereby exhibiting different levels of strength between these principles as well as a sharp borderline between fragments of analysis which are still conservative over **PRA** and extensions which just go beyond the strength of **PRA**.

[RS-98-17] *PostScript*, *PDF*, *DVI.*
Roberto Bruni, Jos Meseguer, Ugo Montanari, and Vladimiro Sassone.
A Comparison of Petri Net Semantics under the Collective Token Philosophy.
September 1998.
20 pp. Appears in Hsiang and Ohori, editors, *Fourth Asian Computing Science Conference*, ASIAN '98 Proceedings, LNCS 1538, 1998, pages 225–244.

Abstract: In recent years, several semantics for *place/transition Petri nets* have been proposed that adopt the *collective token philosophy.* We investigate distinctions and similarities between three such models, namely *configuration structures*, *concurrent*

transition systems, and *(strictly) symmetric (strict) monoidal categories*. We use the notion of adjunction to express each connection. We also present a purely logical description of the collective token interpretation of net behaviours in terms of theories and theory morphisms in *partial membership equational logic*.

[RS-98-16] Stephen Alstrup, Thore Husfeldt, and Theis Rauhe.
Marked Ancestor Problems.
September 1998.
Extended abstract appears in Motwani, editor, *39th Annual Symposium on Foundations of Computer Science*, FOCS '98 Proceedings, 1998, pages 534–543.

[RS-98-15] *PostScript*, *PDF.*
Jung-taek Kim, Kwangkeun Yi, and Olivier Danvy.
Assessing the Overhead of ML Exceptions by Selective CPS Transformation.
September 1998.
31 pp. Appears in the proceedings of the *1998 ACM SIGPLAN Workshop on ML*, Baltimore, Maryland, September 26, 1998, pages 103–114.

Abstract: ML's exception handling makes it possible to describe exceptional execution flows conveniently, but it also forms a performance bottleneck. Our goal is to reduce this overhead by source-level transformation.

To this end, we transform source programs into continuation-passing style (CPS), replacing handle and raise expressions by continuation-catching and throwing expressions, respectively. CPS-transforming every expression, however, introduces a new cost. We therefore use an exception analysis to transform expressions selectively: if an expression is statically determined to involve exceptions then it is CPS-transformed; otherwise, it is left in direct style.

In this article, we formalize this selective CPS transformation, prove its correctness, and present early experimental data indicating its effect on ML programs.

[RS-98-14] *PostScript*, *PDF*, *DVI.*
Sandeep Sen.
The Hardness of Speeding-up Knapsack.
August 1998.
6 pp.

Abstract: We show that it is not possible to speed-up the Knapsack problem efficiently in the parallel algebraic decision tree model. More specifically, we prove that any parallel algorithm in the fixed degree algebraic decision tree model that solves the decision version of the Knapsack problem requires $\Omega(\sqrt{n})$ rounds even by using $2^{\sqrt{n}}$ processors. We extend the result to the PRAM model without bit-operations. These results are consistent with Mulmuley's recent result on the separation of the strongly-polynomial

class and the corresponding NC class in the arithmetic PRAM model.

[RS-98-13] *PostScript*, *PDF*, *DVI.*
Olivier Danvy and Morten Rhiger.
Compiling Actions by Partial Evaluation, Revisited.
June 1998.
25 pp. Earlier version appears in Vain, editor, *9th Nordic Workshop on Programming Theory*, NWPT '9 Proceedings, Estonian Academy of Sciences Series, 1997.
Abstract: We revisit Bondorf and Palsberg's compilation of actions using the offline syntax-directed partial evaluator Similix (FPCA '93, JFP '96), and we compare it in detail with using an online type-directed partial evaluator. In contrast to Similix, our type-directed partial evaluator is idempotent and requires no "binding-time improvements." It also appears to consume about 7 times less space and to be about 28 times faster than Similix, and to yield residual programs that are perceptibly more efficient than those generated by Similix.

[RS-98-12] *PostScript*, *PDF*, *DVI.*
Olivier Danvy.
Functional Unparsing.
May 1998.
7 pp. This report supersedes the earlier report BRICS RS-98-5. Extended version of an article appearing in *Journal of Functional Programming*, 8(6):621–625, 1998.
Abstract: A string-formatting function such as printf in C seemingly requires dependent types, because its control string determines the rest of its arguments.

Examples:

```
printf ("Hello world.\n");
printf ("The %s is %d.\n", "answer", 42);
```

We show how changing the representation of the control string makes it possible to program printf in ML (which does not allow dependent types). The result is well typed and perceptibly more efficient than the corresponding library functions in Standard ML of New Jersey and in Caml.

[RS-98-11] *PostScript*, *PDF*, *DVI.*
Gudmund Skovbjerg Frandsen, Johan P. Hansen, and Peter Bro Miltersen.
Lower Bounds for Dynamic Algebraic Problems.
May 1998.
30 pp. Revised versions appears in Meinel and Tison, editors, *16th Annual Symposium on Theoretical Aspects of Computer Science Proceedings*, STACS '99 Proceedings, LNCS 1563, 1999, pages 362–372, and in *Information and Computation*, 171:333–349, 2001.
Abstract: We consider dynamic evaluation of algebraic functions (matrix multiplication, determinant, convolution, Fourier trans-

form, etc.) in the model of Reif and Tate; i.e., if $f(x_1, \dots, x_n) = (y_1, \dots, y_m)$ is an algebraic problem, we consider serving on-line requests of the form "change input x_i to value v" or "what is the value of output y_i?". We present techniques for showing lower bounds on the worst case time complexity per operation for such problems. The first gives lower bounds in a wide range of rather powerful models (for instance history dependent algebraic computation trees over any infinite subset of a field, the integer RAM, and the generalized real RAM model of Ben-Amram and Galil). Using this technique, we show optimal $\Omega(n)$ bounds for dynamic matrix-vector product, dynamic matrix multiplication and dynamic discriminant and an $\Omega(\sqrt{n})$ lower bound for dynamic polynomial multiplication (convolution), providing a good match with Reif and Tate's $O(\sqrt{n \log n})$ upper bound. We also show linear lower bounds for dynamic determinant, matrix adjoint and matrix inverse and an $\Omega(\sqrt{n})$ lower bound for the elementary symmetric functions. The second technique is the communication complexity technique of Miltersen, Nisan, Safra, and Wigderson which we apply to the setting of dynamic algebraic problems, obtaining similar lower bounds in the word RAM model. The third technique gives lower bounds in the weaker straight line program model. Using this technique, we show an $\Omega((\log n)^2 / \log \log n)$ lower bound for dynamic discrete Fourier transform. Technical ingredients of our techniques are the incompressibility technique of Ben-Amram and Galil and the lower bound for depth-two superconcentrators of Radhakrishnan and Ta-Shma. The incompressibility technique is extended to arithmetic computation in arbitrary fields.

[RS-98-10] *PostScript*, *PDF*, *DVI*.
Jakob Pagter and Theis Rauhe.
Optimal Time-Space Trade-Offs for Sorting.
May 1998.
12 pp. Appears in Motwani, editor, *39th Annual Symposium on Foundations of Computer Science*, FOCS '98 Proceedings, 1998, pages 264–268.

Abstract: 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. Since then, no progress has been made towards tightening this gap.

The main contribution of this paper is a comparison based sorting algorithm which closes this gap by meeting the lower bound of Beame. The time-space product $O(n^2)$ upper bound holds for the full range of space bounds between $\log n$ and $n / \log n$. Hence in this range our algorithm is optimal for comparison based models as well as for the very powerful general models considered by

Beame.

[RS-98-9] *PostScript*, *PDF*.
Zhe Yang.
Encoding Types in ML-like Languages (Preliminary Version).
April 1998.
32 pp. Extended version of a paper appearing in Hudak and Queinnec, editors, *Proceedings of the third ACM SIGPLAN International Conference on Functional Programming*, 1998, pages 289–300.

Abstract: A Hindley-Milner type system such as ML's seems to prohibit type-indexed values, i.e., functions that map a family of types to a family of values. Such functions generally perform case analysis on the input types and return values of possibly different types. The goal of our work is to demonstrate how to program with type-indexed values within a Hindley-Milner type system.

Our first approach is to interpret an input type as its corresponding value, recursively. This solution is type-safe, in the sense that the ML type system statically prevents any mismatch between the input type and function arguments that depend on this type.

Such specific type interpretations, however, prevent us from combining different type-indexed values that share the same type. To meet this objection, we focus on finding a value-independent type encoding that can be shared by different functions. We propose and compare two solutions. One requires first-class and higher-order polymorphism, and, thus, is not implementable in the core language of ML, but it can be programmed using higher-order functors in Standard ML of New Jersey. Its usage, however, is clumsy. The other approach uses embedding/projection functions. It appears to be more practical.

We demonstrate the usefulness of type-indexed values through examples including type-directed partial evaluation, C printf-like formatting, and subtype coercions. Finally, we discuss the tradeoffs between our approach and some other solutions based on more expressive typing disciplines.

[RS-98-8] *PostScript*, *PDF*, *DVI*.
P. S. Thiagarajan and Jesper G. Henriksen.
Distributed Versions of Linear Time Temporal Logic: A Trace Perspective.
April 1998.
49 pp. Appears as a chapter of Reisig and Rozenberg, editors, *Lectures on Petri Nets I: Basic Models*, LNCS 1491, 1998, pages 643–681.

Abstract: Linear time Temporal Logic (LTL) has become a well established tool for specifying the dynamic behaviour of distributed systems. A basic feature of LTL is that its formulas are interpreted over sequences. Typically, such a sequence will model a computation of a system; a sequence of states visited by the sys-

tem or a sequence of actions executed by the system during the course of the computation.

In many applications the computations of a distributed system will constitute interleavings of the occurrences of causally independent actions. Consequently, the computations can be naturally grouped together into equivalence classes where two computations are equated in case they are two different interleavings of the same partially ordered stretch of behaviour. It turns out that many of the properties expressed as LTL-formulas happen to have the so called "all-or-none" property. Either all members of an equivalence class of computations will have the desired property or none will do ("leads to deadlock" is one such property). For verifying such properties one has to check the property for just one member of each equivalence class. This is the insight underlying many of the partial-order based verification methods in which the computational resources required for the verification task can often be dramatically reduced.

It is often the case that the equivalence classes of computations generated by a distributed system constitute objects called Mazurkiewicz traces. They can be canonically represented as restricted labelled partial orders. This opens up an alternative way of exploiting the non-sequential nature of the computations of a distributed systems and the attendant partial-order based methods. It consists of developing linear time temporal logics that can be directly interpreted over Mazurkiewicz traces. In these logics, every specification is guaranteed to have the "all-or-none" property and hence can take advantage of the partial-order based reduction methods during the verification process. The study of these logics also exposes the richness of the partial-order settings from a logical standpoint and the complications that can arise as a consequence.

Our aim here is to present an overview of linear time temporal logics whose models can be viewed as Mazurkiewicz traces. The presentation is, in principle, self-contained though previous exposure to temporal logics and automata over infinite objects will be very helpful. We have provided net-theoretic examples whenever possible in order to emphasize the broad scope of applicability of the material.

[RS-98-7] *PostScript*, *PDF*, *DVI*.
Stephen Alstrup, Thore Husfeldt, and Theis Rauhe.
Marked Ancestor Problems (Preliminary Version).
April 1998.
36 pp. Superseeded by RS-98-16. Extended abstract in Motwani, editor, *39th Annual Symposium on Foundations of Computer Science*, FOCS '98 Proceedings, 1998, pages 534–543.

Abstract: Consider a rooted tree whose nodes can be marked or unmarked. Given a node, we want to find its nearest marked ancestor. This generalises the well-known predecessor problem, where the tree is a path. We show tight upper and lower bounds for this

problem. The lower bounds are proved in the cell probe model, the upper bounds run on a unit-cost RAM. As easy corollaries 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 intersection problems, proximity problems, and ray shooting. Our upper bounds improve a number of algorithms from various fields, including dynamic dictionary matching and coloured ancestor problems.

Comments: The content of this report is identical with the content of Technical Report DIKU-TR-98/9, Department of Computer Science, University of Copenhagen.

[RS-98-6] *PostScript*, *PDF*, *DVI*.
Kim Sunesen.
Further Results on Partial Order Equivalences on Infinite Systems.
March 1998.
48 pp.

Abstract: In a previous paper [ICATPN'96], we investigated decidability issues for standard language equivalence for process description languages with two generalisations based on traditional approaches for capturing non-interleaving behaviour: *pomset equivalence* reflecting global causal dependency, and *location equivalence* reflecting spatial distribution of events.

In this paper, we continue by investigating the role played by TCSP-style renaming and hiding combinators with respect to decidability. One result of [ICATPN'96] was that in contrast to pomset equivalence, location equvivalence remained decidable for a class of processes consisting of finite sets of BPP processes communicating in a TCSP manner. Here, we show that location equivalence becomes undecidable when either renaming or hiding is added to this class of processes.

Furthermore, we investigate the weak versions of location and pomset equivalences. We show that for BPP with τ prefixing, both weak pomset and weak location equivalence are decidable. Moreover, we show that weak location equivalence is undecidable for BPP semantically extended with CCS communication.

[RS-98-5] *PostScript*, *PDF*, *DVI*.
Olivier Danvy.
Formatting Strings in ML.
March 1998.
3 pp. This report is superseded by the later report BRICS RS-98-12.

Abstract: A string-formatting function such as printf in C seemingly requires dependent types, because its control string specifies the rest of its arguments.

Examples:

```
printf ("Hello world.\n");
printf ("The %s is %d.\n", "answer", 42);
```

We show how changing the representation of the control string makes it possible to program printf in ML (which does not allow dependent types). The result is well typed and perceptibly more efficient than the corresponding library functions in Standard ML of New Jersey and in Caml.

[RS-98-4] *PostScript*, *PDF*.
Mogens Nielsen and Thomas S. Hune.
Timed Bisimulation and Open Maps.
February 1998.
27 pp. Appears in Brim, Gruska and Zlatuška, editors, *Mathematical Foundations of Computer Science: 23rd International Symposium*, MFCS '98 Proceedings, LNCS 1450, 1998, pages 378–387.
Abstract: Formal models for real-time systems have been studied intensively over the past decade. Much of the theory of untimed systems have been lifted to real-time settings. One example is the notion of bisimulation applied to timed transition systems, which is studied here within the general categorical framework of open maps. We define a category of timed transition systems, and show how to characterize standard timed bisimulation in terms of spans of open maps with a natural choice of a path category. This allows us to apply general results from the theory of open maps, e.g. the existence of canonical models and characteristic logics. Also, we obtain here an alternative proof of decidability of bisimulation for finite transition systems, and illustrate the use of open maps in finite presentations of bisimulations.

[RS-98-3] *PostScript*, *PDF*, *DVI*.
Christian N. S. Pedersen, Rune B. Lyngs, and Jotun Hein.
Comparison of Coding DNA.
January 1998.
20 pp. Appears in Farach-Colton, editor, *Combinatorial Pattern Matching: 9th Annual Symposium*, CPM '98 Proceedings, LNCS 1448, 1998, pages 153–173.
Abstract: We discuss a model for the evolutionary distance between two coding DNA sequences which specializes to the DNA/protein model proposed in by Hein in the paper "An algorithm combining DNA and protein alignment". We discuss the DNA/protein model in details and present a quadratic time algorithm that computes an optimal alignment of two coding DNA sequences in the model under the assumption of affine gap cost. We believe that the constant factor of the running time is sufficiently small to make the algorithm feasible in practice.

[RS-98-2] Olivier Danvy.
An Extensional Characterization of Lambda-Lifting and Lambda-Dropping.
January 1998.
This report is superseded by the later report BRICS RS-99-21.
Abstract: Lambda-lifting and lambda-dropping respectively transform a block-structured functional program into recursive equations and vice versa. Lambda-lifting is known since the early

80's, whereas lambda-dropping is more recent. Both are split into an analysis and a transformation. Published work, however, has only concentrated on the analysis part. We focus here on the transformation part and more precisely on its formal correctness, which is an open problem. One of our two main theorems suggests us to define extensional versions of lambda-lifting and lambda-dropping, which we visualize both using ML and using type-directed partial evaluation.

[RS-98-1] *PostScript*, *PDF*, *DVI*.
Olivier Danvy.
A Simple Solution to Type Specialization.
January 1998.
10 pp. Appears in Larsen, Skyum and Winskel, editors, *25th International Colloquium on Automata, Languages, and Programming*, ICALP '98 Proceedings, LNCS 1443, 1998, pages 908–917.

Abstract: Partial evaluation specializes terms, but traditionally this specialization does not apply to the type of these terms. As a result, specializing, e.g., an interpreter written in a typed language, which requires a "universal" type to encode expressible values, yields residual programs with type tags all over. Neil Jones has stated that getting rid of these type tags was an open problem, despite possible solutions such as Torben Mogensen's "constructor specialization." To solve this problem, John Hughes has proposed a new paradigm for partial evaluation, "Type Specialization," based on type inference instead of being based on symbolic interpretation. Type Specialization is very elegant in principle but it also appears non-trivial in practice.

Stating the problem in terms of types instead of in terms of type encodings suggests a very simple type-directed solution, namely, to use a projection from the universal type to the specific type of the residual program. Standard partial evaluation then yields a residual program without type tags, simply and efficiently.