BRICS Research Series, Abstracts, 2003

March 26, 2004

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

References

[RS-03-53] *PostScript*, *PDF*, *DVI*.
Kyung-Goo Doh and Peter D. Mosses.
Composing Programming Languages by Combining Action-Semantics Modules.
December 2003.
39 pp. Appears in *Science of Computer Programming*, 47(1):2–36, 2003.

Abstract: This article demonstrates a method for composing a programming language by combining action-semantics modules. Each module is defined separately, and then a programming-language module is defined by combining existing modules. This method enables the language designer to gradually develop a language by defining, selecting and combining suitable modules. The resulting modular structure is substantially different from that previously employed in action-semantic descriptions.

It also discusses how to resolve the conflicts that may arise when combining modules, and indicates some advantages that action semantics has over other approaches in this respect.

[RS-03-52] *PostScript*, *PDF*, *DVI*.
Peter D. Mosses.
Pragmatics of Modular SOS.
December 2003.
22 pp. Invited paper, published in Kirchner and Ringeissen, editors, *Algebraic Methodology and Software Technology: 9th International Conference*, AMAST '02 Proceedings, LNCS 2422, 2002, pages 21–40.

Abstract: Modular SOS is a recently-developed variant of Plotkin's Structural Operational Semantics (SOS) framework. It has several pragmatic advantages over the original framework—the most significant being that rules specifying the semantics of individual language constructs can be given *definitively*, once and for all.

Modular SOS is being used for teaching operational semantics at the undergraduate level. For this purpose, the meta-notation for modular SOS rules has been made more user-friendly, and derivation of computations according to the rules is simulated using Prolog.

After giving an overview of the foundations of Modular SOS, this paper gives some illustrative examples of the use of the framework, and discusses various pragmatic aspects.

[RS-03-51] *PostScript*, *PDF*, *DVI*.

Ulrich Kohlenbach and Branimir Lambov.

Bounds on Iterations of Asymptotically Quasi-Nonexpansive Mappings.

December 2003.

24 pp.

Abstract: This paper establishes explicit quantitative bounds on the computation of approximate fixed points of asymptotically (quasi-) nonexpansive mappings f by means of iterative processes. Here $f : C \to C$ is a selfmapping of a convex subset $C \subseteq X$ of a uniformly convex normed space X. We consider general Krasnoselski-Mann iterations with and without error terms. As a consequence of our quantitative analysis we also get new qualitative results which show that the assumption on the existence of fixed points of f can be replaced by the existence of approximate fixed points only. We explain how the existence of effective uniform bounds in this context can be inferred already a-priorily by a logical metatheorem recently proved by the first author. Our bounds were in fact found with the help of the general logical machinery behind the proof of this metatheorem. The proofs we present here are, however, completely selfcontained and do not require any tools from logic.

[RS-03-50] *PostScript*, *PDF*, *DVI*.

Branimir Lambov.

A Two-Layer Approach to the Computability and Complexity of Real Numbers.

December 2003.

16 pp.

Abstract: We present a new approach to computability of real numbers in which real functions have type-1 representations, which also includes the ability to reason about the complexity of real numbers and functions. We discuss how this allows efficient implementations of exact real numbers and also present a new real number system that is based on it.

[RS-03-49] *PostScript*, *PDF*.

Marius Mikucionis, Kim G. Larsen, and Brian Nielsen.

Online On-the-Fly Testing of Real-time Systems.

December 2003.

14 pp.

Abstract: In this paper we present a framework, an algorithm and a new tool for online testing of real-time systems based on symbolic techniques used in UPPAAL model checker. We extend UPPAAL timed automata network model to a test specification which is used to generate test primitives and to check the correctness of system responses including the timing aspects. We use timed trace inclusion as a conformance relation between system and specification to draw a test verdict. The test generation and execution algorithm is implemented as an extension to UPPAAL and experiments carried out to examine the correctness and performance of the tool. The experiment results are promising.

[RS-03-48] *PostScript*, *PDF.*
Kim G. Larsen, Ulrik Larsen, Brian Nielsen, Arne Skou, and Andrzej Wasowski.
Danfoss EKC Trial Project Deliverables.
December 2003.
53 pp.
Abstract: This report documents the results of the Danfoss EKC trial project on model based development using IAR visualState. We present a formal state-model of an refrigeration controller based on a specification given by Danfoss. We report results on modeling, verification, simulation, and code-generation. It is found that the IAR visualState is a promising tool for this application domain, but that improvements must be done to code-generation and automatic test generation.

[RS-03-47] *PostScript*, *PDF*, *DVI.*
Hans Httel and Jiř Srba.
Recursive Ping-Pong Protocols.
December 2003.
21 pp. To appear in the proceedings of 2004 IFIP WG 1.7, ACM SIGPLAN and GI FoMSESS Workshop on Issues in the Theory of Security (WITS'04).
Abstract: This paper introduces a process calculus with recursion which allows us to express an unbounded number of runs of the ping-pong protocols introduced by Dolev and Yao. We study the decidability issues associated with two common approaches to checking security properties, namely reachability analysis and bisimulation checking. Our main result is that our channel-free and memory-less calculus is Turing powerful, assuming that at least three principals are involved. We also investigate the expressive power of the calculus in the case of two participants. Here, our main results are that reachability and, under certain conditions, also strong bisimilarity become decidable.

[RS-03-46] *PostScript*, *PDF*, *DVI.*
Philipp Gerhardy.
The Role of Quantifier Alternations in Cut Elimination.
December 2003.
10 pp. Extends paper appearing in Baaz and Makowsky, editors, *European Association for Computer Science Logic: 17th International Workshop*, CSL '03 Proceedings, LNCS 2803, 2003, pages 212-225.
Abstract: Extending previous results from the author's master's thesis, subsequently published in the proceedings of CSL 2003, on the complexity of cut elimination for the sequent calculus

LK, we discuss the role of quantifier alternations and develope a measure to describe the complexity of cut elimination in terms of quantifier alternations in cut formulas and contractions on such formulas.

[RS-03-45] *PostScript*, *PDF*, *DVI*.
Peter Bro Miltersen, Jaikumar Radhakrishnan, and Ingo Wegener.
On converting CNF to DNF.
December 2003.
11 pp. A preliminary version appeared in Rovan and Vojts, editors, *Mathematical Foundations of Computer Science: 28th International Symposium*, MFCS '03 Proceedings, LNCS 2747, 2003, pages 612–621.

Abstract: We study how big the blow-up in size can be when one switches between the CNF and DNF representations of boolean functions. For a function $f : \{0,1\}^n \to \{0,1\}$, $\mathsf{cnfsize}(f)$ denotes the minimum number of clauses in a CNF for f; similarly, $\mathsf{dnfsize}(f)$ denotes the minimum number of terms in a DNF for f. For $0 \le m \le 2^{n-1}$, let $\mathsf{dnfsize}(m,n)$ be the maximum $\mathsf{dnfsize}(f)$ for a function $f : \{0,1\}^n \to \{0,1\}$ with $\mathsf{cnfsize}(f) \le m$. We show that there are constants $c_1, c_2 \ge 1$ and $\epsilon > 0$, such that for all large n and all $m \in [\frac{1}{\epsilon}n, 2^{\epsilon n}]$, we have

$$2^{n - c_1 \frac{n}{\log(m/n)}} \;\le\; \mathsf{dnfsize}(m,n) \;\le\; 2^{n - c_2 \frac{n}{\log(m/n)}}.$$

In particular, when m is the polynomial n^c, we get $\mathsf{dnfsize}(n^c, n) = 2^{n - \theta(c^{-1}\frac{n}{\log n})}$.

[RS-03-44] *PostScript*, *PDF*, *DVI*.
Anna Gl and Peter Bro Miltersen.
The Cell Probe Complexity of Succinct Data Structures.
December 2003.
17 pp. Appears in Lingas and Nilsson, editors, *Fundamentals of Computation Theory: 14th International Conference*, FCT '03 Proceedings, LNCS 2751, 2003, pages 442–453. An early version of this paper appeared in Baeten, Lenstra, Parrow and Woeginger, editors, *30th International Colloquium on Automata, Languages, and Programming*, ICALP '03 Proceedings, LNCS 2719, 2003, pages 332–344.

Abstract: In the cell probe model with word size 1 (the bit probe model), a static data structure problem is given by a map $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$, where $\{0,1\}^n$ is a set of possible data to be stored, $\{0,1\}^m$ is a set of possible queries (for natural problems, we have mn) and $f(x,y)$ is the answer to question y about data x.

A solution is given by a representation $\phi : \{0,1\}^n \to \{0,1\}^s$ and a query algorithm q so that $q(\phi(x), y) = f(x,y)$. The time t of the query algorithm is the number of bits it reads in $\phi(x)$.

In this paper, we consider the case of *succinct* representations where $s = n + r$ for some *redundancy* rn. For a boolean version of the problem of polynomial evaluation with preprocessing of

coefficients, we show a lower bound on the redundancy-query time tradeoff of the form

$$(r+1)t \geq \Omega(n/\log n).$$

In particular, for very small redundancies r, we get an almost optimal lower bound stating that the query algorithm has to inspect almost the entire data structure (up to a logarithmic factor). We show similar lower bounds for problems satisfying a certain combinatorial property of a coding theoretic flavor. Previously, no $\omega(m)$ lower bounds were known on t in the general model for explicit functions, even for very small redundancies.

By restricting our attention to *systematic* or *index* structures ϕ satisfying $\phi(x) = x\phi^*(x)$ for some map ϕ^* (where denotes concatenation) we show similar lower bounds on the redundancy-query time tradeoff for the natural data structuring problems of Prefix Sum and Substring Search.

[RS-03-43] *PostScript*, *PDF*, *DVI*.
Mikkel Nygaard and Glynn Winskel.
Domain Theory for Concurrency.
December 2003.
45 pp. To appear in a *Theoretical Computer Science* special issue on Domain Theory.

Abstract: A simple domain theory for concurrency is presented. Based on a categorical model of linear logic and associated comonads, it highlights the role of linearity in concurrent computation. Two choices of comonad yield two expressive metalanguages for higher-order processes, both arising from canonical constructions in the model. Their denotational semantics are fully abstract with respect to contextual equivalence. One language derives from an exponential of linear logic; it supports a straightforward operational semantics with simple proofs of soundness and adequacy. The other choice of comonad yields a model of affine-linear logic, and a process language with a tensor operation to be understood as a parallel composition of independent processes. The domain theory can be generalised to presheaf models, providing a more refined treatment of nondeterministic branching. The article concludes with a discussion of a broader programme of research, towards a fully fledged domain theory for concurrency.

[RS-03-42] *PostScript*, *PDF*, *DVI*.
Mikkel Nygaard and Glynn Winskel.
Full Abstraction for HOPLA.
December 2003.
25 pp. Appears in Amadio and Lugiez, editors, *Concurrency Theory: 14th International Conference*, CONCUR '03 Proceedings, LNCS 2761, 2003, pages 383–398.

Abstract: A fully abstract denotational semantics for the higher-order process language HOPLA is presented. It characterises contextual and logical equivalence, the latter linking up with simulation. The semantics is a clean, domain-theoretic description of processes as downwards-closed sets of computation paths: the operations of HOPLA arise as syntactic encodings of canonical constructions on such sets; full abstraction is a direct consequence of expressiveness with respect to computation paths; and simple proofs of soundness and adequacy shows correspondence between the denotational and operational semantics.

[RS-03-41] *PostScript*, *PDF*, *DVI*.
Malgorzata Biernacka, Dariusz Biernacki, and Olivier Danvy.
An Operational Foundation for Delimited Continuations.
December 2003.
21 pp.

Abstract: We derive an abstract machine that corresponds to a definitional interpreter for the control operators shift and reset. Based on this abstract machine, we construct a syntactic theory of delimited continuations.

Both the derivation and the construction scale to the family of control operators shift_n and reset_n. The definitional interpreter for shift_n and reset_n has $n+1$ layers of continuations, the corresponding abstract machine has $n+1$ layers of control stacks, and the corresponding syntactic theory has $n+1$ layers of evaluation contexts.

[RS-03-40] *PostScript*, *PDF*, *DVI*.
Andrzej Filinski and Henning Korsholm Rohde.
A Denotational Account of Untyped Normalization by Evaluation.
December 2003.
29 pp.

Abstract: We show that the standard normalization-by-evaluation construction for the simply-typed $\lambda_{\beta\eta}$-calculus has a natural counterpart for the untyped λ_{β}-calculus, with the central type-indexed logical relation replaced by a "recursively defined" *invariant relation*, in the style of Pitts. In fact, the construction can be seen as generalizing a computational-adequacy argument for an untyped, call-by-name language to normalization instead of evaluation.

In the untyped setting, not all terms have normal forms, so the normalization function is necessarily partial. We establish its correctness in the senses of *soundness* (the output term, if any, is β-equivalent to the input term); *standardization* (β-equivalent terms are mapped to the same result); and *completeness* (the function is defined for all terms that do have normal forms). We also show how the semantic construction enables a simple yet formal correctness proof for the normalization algorithm, expressed as a functional program in an ML-like call-by-value language.

[RS-03-39] *PostScript*, *PDF*.
Jrg Abendroth.
Applying π-Calculus to Practice: An Example of a Unified Security Mechanism.

November 2003.
35 pp.

Abstract: The Pi-calculus has been developed to reason about behavioural equivalence. Different notions of equivalence are defined in terms of process interactions, as well as the context of processes. There are various extensions of the Pi-calculus, such as the SPI calculus, which has primitives to facilitate security protocol design.

Another area of computer security is access control research, which includes problems of access control models, policies and access control mechanism. The design of a unified framework for access control requires that all policies are supported and different access control models are instantiated correctly.

In this paper we will utilise the Pi calculus to reason about access control policies and mechanism. An equivalence of different policy implementations, as well as access control mechanism will be shown. Finally some experiences regarding the use of Pi-calculus are presented.

[RS-03-38] *PostScript*, *PDF*.
Henning Bttger, Anders Mller, and Michael I. Schwartzbach.
Contracts for Cooperation between Web Service Programmers and HTML Designers.
November 2003.
23 pp.

Abstract: Interactive Web services consist of a mixture of HTML fragments and program code. The fragments, which are maintained by designers, are combined to form HTML pages that are shown to the clients. The code, which is maintained by programmers, is executed on the server to handle the business logic. Current Web service frameworks provide little help in separating these constituents, which complicates cooperation between programmers and HTML designers.

We propose a system based on XML templates and formalized contracts allowing a flexible separation of concerns. The contracts act as interfaces between the programmers and the HTML designers and permit tool support for statically checking that both parties fulfill their obligations. This ensures that (1) programmers and HTML designers work more independently focusing on their own expertises, (2) the Web service implementation is better structured and thus easier to develop and maintain, (3) it is guaranteed that only valid HTML is sent to the clients even though it is constructed dynamically, (4) the programmer uses the XML templates consistently, and (5) the form input fields being sent to the client always match the code receiving those values. Additionally, we describe tools that aid in the construction and management of contracts and XML templates.

[RS-03-37] *PostScript*, *PDF*.
Claude Crpeau, Paul Dumais, Dominic Mayers, and Louis Salvail.

Computational Collapse of Quantum State with Application to Oblivious Transfer.
November 2003.
30 pp.

Abstract: Quantum 2-party cryptography differs from its classical counterpart in at least one important way: Given blak-box access to a perfect commitment scheme there exists a secure $1-2$ *quantum* oblivious transfer. This reduction proposed by Crpeau and Kilian was proved secure against any receiver by Yao, in the case where perfect commitments are used. However, quantum commitments would normally be based on computational assumptions. A natural question therefore arises: What happens to the security of the above reduction when computationally secure commitments are used instead of perfect ones?

In this paper, we address the security of $1-2$ QOT when computationally binding string commitments are available. In particular, we analyse the security of a primitive called *Quantum Measurement Commitment* when it is constructed from unconditionally concealing but computationally binding commitments. As measuring a quantum state induces an irreversible collapse, we describe a QMC as an instance of "computational collapse of a quantum state". In a QMC a state appears to be collapsed to a polynomial time observer who cannot extract full information about the state without breaking a computational assumption.

We reduce the security of QMC to a *weak* binding criteria for the string commitment. We also show that *secure* QMCs implies QOT using a straightforward variant of the reduction above.

[RS-03-36] *PostScript*, *PDF*.
Ivan B. Damgrd, Serge Fehr, Kirill Morozov, and Louis Salvail.
Unfair Noisy Channels and Oblivious Transfer.
November 2003.
26 pp. Appears in Naor, editor, *The first Theory of Cryptography Conference*, TCC '04 Proceedings, LNCS 2951, 2004, pages 355–373.

Abstract: In a paper from EuroCrypt'99, Damgrd, Kilian and Salvail show various positive and negative results on constructing Bit Commitment (BC) and Oblivious Transfer (OT) from Unfair Noisy Channels (UNC), i.e., binary symmetric channels where the error rate is only known to be in a certain interval $[\gamma..\delta]$ and can be chosen adversarially. They also introduce a related primitive called $PassiveUNC$. We prove in this paper that any OT protocol that can be constructed based on a $PassiveUNC$ and is secure against a passive adversary can be transformed using a generic "compiler" into an OT protocol based on a UNC which is secure against an active adversary. Apart from making positive results easier to prove in general, this also allows correcting a problem in the EuroCrypt'99 paper: There, a positive result was claimed on constructing from UNC an OT that is secure against active cheating. We point out that the proof sketch given

for this was incomplete, and we show that a correct proof of a much stronger result follows from our general compilation result and a new technique for transforming between weaker versions of OT with different parameters.

[RS-03-35] *PostScript*, *PDF*, *DVI*.
Mads Sig Ager, Olivier Danvy, and Jan Midtgaard.
A Functional Correspondence between Monadic Evaluators and Abstract Machines for Languages with Computational Effects.
November 2003.
31 pp.

Abstract: We extend our correspondence between evaluators and abstract machines from the pure setting of the lambda-calculus to the impure setting of the computational lambda-calculus. Specifically, we show how to derive new abstract machines from monadic evaluators for the computational lambda-calculus. Starting from a monadic evaluator and a given monad, we inline the components of the monad in the evaluator and we derive the corresponding abstract machine by closure-converting, CPS-transforming, and defunctionalizing this inlined interpreter. We illustrate the construction first with the identity monad, obtaining yet again the CEK machine, and then with a state monad, an exception monad, and a combination of both.

In addition, we characterize the tail-recursive stack inspection presented by Clements and Felleisen at ESOP 2003 as a canonical state monad. Combining this state monad with an exception monad, we construct an abstract machine for a language with exceptions and properly tail-recursive stack inspection. The construction scales to other monads—including one more properly dedicated to stack inspection than the state monad—and other monadic evaluators.

[RS-03-34] *PostScript*, *PDF*, *DVI*.
Luca Aceto, Willem Jan Fokkink, Anna Inglfsdttir, and Bas Luttik.
CCS with Hennessy's Merge has no Finite Equational Axiomatization.
November 2003.
37 pp.

Abstract: This paper confirms a conjecture of Bergstra and Klop's from 1984 by establishing that the process algebra obtained by adding an auxiliary operator proposed by Hennessy in 1981 to the recursion free fragment of Milner's Calculus of Communicationg Systems is not finitely based modulo bisimulation equivalence. Thus Hennessy's merge cannot replace the left merge and communication merge operators proposed by Bergstra and Klop, at least if a finite axiomatization of parallel composition is desired.

[RS-03-33] *PostScript*, *PDF*, *DVI*.
Olivier Danvy.
A Rational Deconstruction of Landin's SECD Machine.
October 2003.
32 pp. This report supersedes the earlier BRICS report RS-02-53.

Abstract: Landin's SECD machine was the first abstract machine for the λ-calculus viewed as a programming language. Both theoretically as a model of computation and practically as an idealized implementation, it has set the tone for the subsequent development of abstract machines for functional programming languages. However, and even though variants of the SECD machine have been presented, derived, and invented, the precise rationale for its architecture and modus operandi has remained elusive. In this article, we deconstruct the SECD machine into a λ-interpreter, i.e., an evaluation function, and we reconstruct λ-interpreters into a variety of SECD-like machines. The deconstruction and reconstructions are transformational: they are based on equational reasoning and on a combination of simple program transformations—mainly closure conversion, transformation into continuation-passing style, and defunctionalization.

The evaluation function underlying the SECD machine provides a precise rationale for its architecture: it is an environment-based eval-apply evaluator with a callee-save strategy for the environment, a data stack of intermediate results, and a control delimiter. Each of the components of the SECD machine (stack, environment, control, and dump) is therefore rationalized and so are its transitions.

The deconstruction and reconstruction method also applies to other abstract machines and other evaluation functions. It makes it possible to systematically extract the denotational content of an abstract machine in the form of a compositional evaluation function, and the (small-step) operational content of an evaluation function in the form of an abstract machine.

[RS-03-32] *PostScript*, *PDF*, *DVI*.
Philipp Gerhardy and Ulrich Kohlenbach.
Extracting Herbrand Disjunctions by Functional Interpretation.
October 2003.
17 pp.

Abstract: Carrying out a suggestion by Kreisel, we adapt Gdel's functional interpretation to ordinary first-order predicate logic(PL) and thus devise an algorithm to extract Herbrand terms from PL-proofs. The extraction is carried out in an extension of PL to higher types. The algorithm consists of two main steps: first we extract a functional realizer, next we compute the β-normal-form of the realizer from which the Herbrand terms can be read off. Even though the extraction is carried out in the extended language, the terms are ordinary PL-terms. In contrast to approaches to Herbrand's theorem based on cut elimination or ε-elimination this extraction technique is, except for the normalization step, of low polynomial complexity, fully modular and furthermore allows an analysis of the structure of the Herbrand terms, in the spirit of Kreisel, already prior to the normalization step. It is expected that the implementation of functional interpretation in Schwichtenberg's MINLOG system can be adapted

to yield an efficient Herbrand-term extraction tool.

[RS-03-31] *PostScript*, *PDF*, *DVI*.
Stephen Lack and Paweł Sobociński.
Adhesive Categories.
October 2003.
25 pp.
Abstract: We introduce adhesive categories, which are categories with structure ensuring that pushouts along monomorphisms are well-behaved. Many types of graphical structures used in computer science are shown to be examples of adhesive categories. Double-pushout graph rewriting generalises well to rewriting on arbitrary adhesive categories.

[RS-03-30] *PostScript*, *PDF*.
Jesper Makholm Byskov, Bolette Ammitzbll Madsen, and Bjarke Skjernaa.
New Algorithms for Exact Satisfiability.
October 2003.
31 pp.
Abstract: The Exact Satisfiability problem is to determine if a CNF-formula has a truth assignment satisfying exactly one literal in each clause; Exact 3-Satisfiability is the version in which each clause contains at most three literals. In this paper, we present algorithms for Exact Satisfiability and Exact 3-Satisfiability running in time $O(2^{0.2325n})$ and $O(2^{0.1379n})$, respectively. The previously best algorithms have running times $O(2^{0.2441n})$ for Exact Satisfiability (Monien, Speckenmeyer and Vornberger (1981)) and $O(2^{0.1626n})$ for Exact 3-Satisfiability (Kulikov and independently Porschen, Randerath and Speckenmeyer (2002)). We extend the case analyses of these papers and observe, that a formula not satisfying any of our cases has a small number of variables, for which we can try all possible truth assignments and for each such assignment solve the remaining part of the formula in polynomial time.

[RS-03-29] *PostScript*, *PDF*.
Aske Simon Christensen, Christian Kirkegaard, and Anders Mller.
A Runtime System for XML Transformations in Java.
October 2003.
15 pp.
Abstract: We show that it is possible to extend a general-purpose programming language with a convenient high-level data-type for manipulating XML documents while permitting (1) precise static analysis for guaranteeing validity of the constructed XML documents relative to the given DTD schemas, and (2) a runtime system where the operations can be performed efficiently. The system, named XACT, is based on a notion of immutable XML templates and uses XPath for deconstructing documents. A companion paper presents the program analysis; this paper focuses on the efficient runtime representation.

[RS-03-28] *PostScript*, *PDF*, *DVI*.
Zoltn sik and Kim G. Larsen.
Regular Languages Definable by Lindstrm Quantifiers.
August 2003.
82 pp. This report supersedes the earlier BRICS report RS-02-20.

Abstract: In our main result, we establish a formal connection between Lindstrm quantifiers with respect to regular languages and the double semidirect product of finite monoids with a distinguished set of generators. We use this correspondence to characterize the expressive power of Lindstrm quantifiers associated with a class of regular languages.

[RS-03-27] *PostScript*, *PDF*, *DVI*.
Luca Aceto, Willem Jan Fokkink, Rob J. van Glabbeek, and Anna Inglfsdttir.
Nested Semantics over Finite Trees are Equationally Hard.
August 2003.
31 pp.

Abstract: This paper studies nested simulation and nested trace semantics over the language BCCSP, a basic formalism to express finite process behaviour. It is shown that none of these semantics affords finite (in)equational axiomatizations over BCCSP. In particular, for each of the nested semantics studied in this paper, the collection of sound, closed (in)equations over a singleton action set is not finitely based.

[RS-03-26] *PostScript*, *PDF*, *DVI*.
Olivier Danvy and Ulrik P. Schultz.
Lambda-Lifting in Quadratic Time.
August 2003.
23 pp. Extended version of a paper appearing in Hu and Rodrguez-Artalejo, editors, *Sixth International Symposium on Functional and Logic Programming*, FLOPS '02 Proceedings, LNCS 2441, 2002, pages 134–151. This report supersedes the earlier BRICS report RS-02-30.

Abstract: Lambda-lifting is a program transformation used in compilers and in partial evaluators and that operates in cubic time. In this article, we show how to reduce this complexity to quadratic time, and we present a flow-sensitive lambda-lifter that also works in quadratic time.

Lambda-lifting transforms a block-structured program into a set of recursive equations, one for each local function in the source program. Each equation carries extra parameters to account for the free variables of the corresponding local function *and of all its callees*. It is the search for these extra parameters that yields the cubic factor in the traditional formulation of lambda-lifting, which is due to Johnsson. This search is carried out by a transitive closure.

Instead, we partition the call graph of the source program into strongly connected components, based on the simple observation that *all functions in each component need the same extra param-*

eters and thus a transitive closure is not needed. We therefore simplify the search for extra parameters by treating each strongly connected component instead of each function as a unit, thereby reducing the time complexity of lambda-lifting from $\mathcal{O}(n^3 \log n)$ to $\mathcal{O}(n^2 \log n)$, where n is the size of the program.

Since a lambda-lifter can output programs of size $\mathcal{O}(n^2)$, we believe that our algorithm is close to optimal.

[RS-03-25] Biernacki Dariusz and Danvy Olivier.
From Interpreter to Logic Engine by Defunctionalization.
June 2003.
13 pp. Presented at the International Symposium on Logic Based Program Development and Transformation, LOPSTR 2003, Uppsala, Sweden, August 25–27, 2003. This report is superseded by the later report BRICS RS-04-5.

Abstract: Starting from a continuation-based interpreter for a simple logic programming language, propositional Prolog with cut, we derive the corresponding logic engine in the form of an abstract machine. The derivation originates in previous work (our article at PPDP 2003) where it was applied to the lambda-calculus. The key transformation here is Reynolds's defunctionalization that transforms a tail-recursive, continuation-passing interpreter into a transition system, i.e., an abstract machine. Similar denotational and operational semantics were studied by de Bruin and de Vink in previous work (their article at TAPSOFT 1989), and we compare their study with our derivation. Additionally, we present a direct-style interpreter of propositional Prolog expressed with control operators for delimited continuations.

[RS-03-24] *PostScript*, *PDF*, *DVI*.
Mads Sig Ager, Olivier Danvy, and Jan Midtgaard.
A Functional Correspondence between Call-by-Need Evaluators and Lazy Abstract Machines.
June 2003.
13 pp.

Abstract: We bridge the gap between compositional evaluators and abstract machines for the lambda-calculus, using closure conversion, transformation into continuation-passing style, and defunctionalization of continuations. This article is a spin-off of our article at PPDP 2003, where we consider call by name and call by value. Here, however, we consider call by need.

We derive a lazy abstract machine from an ordinary call-by-need evaluator that threads a heap of updatable cells. In this resulting abstract machine, the continuation fragment for updating a heap cell naturally appears as an 'update marker', an implementation technique that was invented for the Three Instruction Machine and subsequently used to construct lazy variants of Krivine's abstract machine. Tuning the evaluator leads to other implementation techniques such as unboxed values and the push-enter model. The correctness of the resulting abstract machines is a corollary

of the correctness of the original evaluators and of the program transformations used in the derivation.

[RS-03-23] *PostScript*, *PDF*, *DVI*.
Margarita Korovina.
Recent Advances in Σ-definability over Continuous Data Types.
June 2003.
24 pp.
Abstract: The purpose of this paper is to survey our recent research in computability and definability over continuous data types such as the real numbers, real-valued functions and functionals. We investigate the expressive power and algorithmic properties of the language of Σ-formulas intended to represent computability over the real numbers. In order to adequately represent computability we extend the reals by the structure of hereditarily finite sets. In this setting it is crucial to consider the real numbers without equality since the equality test is undecidable over the reals. We prove Engeler's Lemma for Σ-definability over the reals without the equality test which relates Σ-definability with definability in the constructive infinitary language $L_{\omega_1\omega}$. Thus, a relation over the real numbers is Σ-definable if and only if it is definable by a disjunction of a recursively enumerable set of quantifier free formulas. This result reveals computational aspects of Σ-definability and also gives topological characterisation of Σ-definable relations over the reals without the equality test. We also illustrate how computability over the real numbers can be expressed in the language of Σ-formulas.

[RS-03-22] *PostScript*, *PDF*, *DVI*.
Ivan B. Damgrd and Mads J. Jurik.
Scalable Key-Escrow.
May 2003.
15 pp.
Abstract: We propose a cryptosystem that has an inherent key escrow mechanism. This leads us to propose a session based public verifiable key escrow system that greatly improves the amount of key material the escrow servers has to keep in order to decrypt an encryption. In our scheme the servers will only have a single secret sharing, as opposed to a single key from every escrowed player. This is done while still having the properties: 1) public verifiable: the user proves to everyone that the encryption can indeed be escrowed, and 2) no secret leakage: no matter how many decryptions a law enforcement agency is presented, it will gain no more information on the users private key, than it couldn't have calculated itself.

[RS-03-21] *PostScript*, *PDF*, *DVI*.
Ulrich Kohlenbach.
Some Logical Metatheorems with Applications in Functional Analysis.
May 2003.

55 pp. Slighly revised and extended version to appear in *Transactions of the American Mathematical Society*.

Abstract: In previous papers we have developed proof-theoretic techniques for extracting effective uniform bounds from large classes of ineffective existence proofs in functional analysis. 'Uniform' here means independence from parameters in compact spaces. A recent case study in fixed point theory systematically yielded uniformity even w.r.t. parameters in metrically bounded (but noncompact) subsets which had been known before only in special cases. In the present paper we prove general logical metatheorems which cover these applications to fixed point theory as special cases but are not restricted to this area at all. Our theorems guarantee under general logical conditions such strong uniform versions of non-uniform existence statements. Moreover, they provide algorithms for actually extracting effective uniform bounds and transforming the original proof into one for the stronger uniformity result. Our metatheorems deal with general classes of spaces like metric spaces, hyperbolic spaces, normed linear spaces, uniformly convex spaces as well as inner product spaces.

[RS-03-20] *PostScript*, *PDF*.
Mads Sig Ager, Olivier Danvy, and Henning Korsholm Rohde.
Fast Partial Evaluation of Pattern Matching in Strings.
May 2003.
16 pp. Final version in Leuschel, editor, *ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, PEPM '03 Proceedings, 2003, pages 3–9. This report supersedes the earlier BRICS report RS-03-11.

Abstract: We show how to obtain all of Knuth, Morris, and Pratt's linear-time string matcher by partial evaluation of a quadratic-time string matcher with respect to a pattern string. Although it has been known for 15 years how to obtain this linear matcher by partial evaluation of a quadratic one, how to obtain it *in linear time* has remained an open problem.

Obtaining a linear matcher by partial evaluation of a quadratic one is achieved by performing its backtracking at specialization time and memoizing its results. We show (1) how to rewrite the source matcher such that its static intermediate computations can be shared at specialization time and (2) how to extend the memoization capabilities of a partial evaluator to static functions. Such an extended partial evaluator, if its memoization is implemented efficiently, specializes the rewritten source matcher in linear time.

[RS-03-19] *PostScript*, *PDF*.
Christian Kirkegaard, Anders Mller, and Michael I. Schwartzbach.
Static Analysis of XML Transformations in Java.
May 2003.
29 pp.

Abstract: XML documents generated dynamically by programs

are typically represented as text strings or DOM trees. This is a low-level approach for several reasons: 1) Traversing and modifying such structures can be tedious and error prone; 2) Although schema languages, e.g. DTD, allow classes of XML documents to be defined, there are generally no automatic mechanisms for statically checking that a program transforms from one class to another as intended. We introduce XACT, a high-level approach for Java using XML templates as a first-class data type with operations for manipulating XML values based on XPath. In addition to an efficient runtime representation, the data type permits static type checking using DTD schemas as types. By specifying schemas for the input and output of a program, our algorithm will statically verify that valid input data is always transformed into valid output data and that no errors occur during processing.

[RS-03-18] *PostScript*, *PDF*, *DVI*.
Bartek Klin and Paweł Sobociński.
Syntactic Formats for Free: An Abstract Approach to Process Equivalence.
April 2003.
41 pp. Appears in Amadio and Lugiez, editors, *Concurrency Theory: 14th International Conference*, CONCUR '03 Proceedings, LNCS 2761, 2003, pages 72–86.
Abstract: A framework of Plotkin and Turis, originally aimed at providing an abstract notion of bisimulation, is modified to cover other operational equivalences and preorders. Combined with bialgebraic methods, it yields a technique for deriving syntactic formats for transition system specifications, which guarantee operational preorders to be precongruences. The technique is applied to the trace prorder, the completed trace preorder and the failures preorder. In the latter two cases, new syntactic formats guaranteeing precongruence properties are introduced.

[RS-03-17] *PostScript*, *PDF*.
Luca Aceto, Jens Alsted Hansen, Anna Inglfsdttir, Jacob Johnsen, and John Knudsen.
The Complexity of Checking Consistency of Pedigree Information and Related Problems.
March 2003.
31 pp. This paper supersedes BRICS Report RS-02-42.
Abstract: Consistency checking is a fundamental computational problem in genetics. Given a pedigree and information on the genotypes (of some) of the individuals in it, the aim of consistency checking is to determine whether these data are consistent with the classic Mendelian laws of inheritance. This problem arose originally from the geneticists' need to filter their input data from erroneous information, and is well motivated from both a biological and a sociological viewpoint. This paper shows that consistency checking is NP-complete, even with focus on a single gene and in the presence of three alleles. Several other results on the computational complexity of problems from genetics that are

related to consistency checking are also offered. In particular, it is shown that checking the consistency of pedigrees over two alleles, and of pedigrees without loops, can be done in polynomial time.

[RS-03-16] *PostScript*, *PDF*, *DVI*.
Ivan B. Damgrd and Mads J. Jurik.
A Length-Flexible Threshold Cryptosystem with Applications.
March 2003.
31 pp.

Abstract: We propose a public-key cryptosystem which is derived from the Paillier cryptosystem. The scheme inherits the attractive homomorphic properties of Paillier encryption. In addition, we achieve two new properties: First, all users can use the same modulus when generating key pairs, this allows more efficient proofs of relations between different encryptions. Second, we can construct a threshold decryption protocol for our scheme that is length flexible, i.e., it can handle efficiently messages of arbitrary length, even though the public key and the secret key shares held by decryption servers are of fixed size. We show how to apply this cryptosystem to build:

1) a self-tallying election scheme with perfect ballot secrecy. This is a small voting system where the result can be computed from the submitted votes without the need for decryption servers. The votes are kept secret unless the cryptosystem can be broken, regardless of the number of cheating parties. This is in contrast to other known schemes that usually require a number of decryption servers, the majority of which must be honest.

2) a length-flexible mix-net which is universally verifiable, where the size of keys and ciphertexts do not depend on the number of mix servers, and is robust against a corrupt minority. Mix-nets can provide anonymity by shuffling messages to provide a random permutation of input ciphertexts to the output plaintexts such that no one knows which plaintexts relate to which ciphertexts. The mix-net inherits several nice properties from the underlying cryptosystem, thus making it useful for a setting with small messages or high computational power, low-band width and that anyone can verify that the mix have been done correctly.

[RS-03-15] *PostScript*, *PDF*, *DVI*.
Anna Inglfsdttir.
A Semantic Theory for Value–Passing Processes Based on the Late Approach.
March 2003.
48 pp.

Abstract: A general class of languages for value-passing calculi based on the late semantic approach is defined and a concrete instantiation of the general syntax is given. This is a modification of the standard CCS according to the late approach. Three kinds of semantics are given for this language. First a Plotkin style operational semantics by means of an applicative labelled

transition system is introduced. This is a modification of the standard labelled transition system that caters for value-passing according to the late approach. As an abstraction, late bisimulation preorder is given. Then a general class of denotational models for the late semantics is defined. A denotational model for the concrete language is given as an instantiation of the general class. Two equationally based proof systems are defined. The first one, which is value-finitary, i. e. only reasons about a finite number of values at each time, is shown to be sound and complete with respect to this model. The second proof system, a value-infinitary one, is shown to be sound with respect to the model, whereas the completeness is proven later. The operational and the denotational semantics are compared and it is shown that the bisimulation preorder is finer than the preorder induced by the denotational model. We also show that in general the ω-bisimulation preorder is strictly included in the model induced preorder. Finally a value-finitary version of the bisimulation preorder is defined and the full abstractness of the denotational model with respect to it is shown. It is also shown that for CCS_L the ω-bisimulation preorder coincides with the preorder induced by the model. From this we can conclude that if we allow for parameterized recursion in our language, we may express processes which coincide in any algebraic domain but are distinguished by the ω-bisimulation. This shows that if we extend CCS_L in this way we obtain a strictly more expressive language.

[RS-03-14] *PostScript*, *PDF*, *DVI*.

Mads Sig Ager, Dariusz Biernacki, Olivier Danvy, and Jan Midtgaard.

From Interpreter to Compiler and Virtual Machine: A Functional Derivation.

March 2003.

36 pp.

Abstract: We show how to derive a compiler and a virtual machine from a compositional interpreter. We first illustrate the derivation with two evaluation functions and two normalization functions. We obtain Krivine's machine, Felleisen et al.'s CEK machine, and a generalization of these machines performing strong normalization, which is new. We observe that several existing compilers and virtual machines—e.g., the Categorical Abstract Machine (CAM), Schmidt's VEC machine, and Leroy's Zinc abstract machine—are already in derived form and we present the corresponding interpreter for the CAM and the VEC machine. We also consider Hannan and Miller's CLS machine and Landin's SECD machine.

We derived Krivine's machine via a call-by-name CPS transformation and the CEK machine via a call-by-value CPS transformation. These two derivations hold both for an evaluation function and for a normalization function. They provide a non-trivial illustration of Reynolds's warning about the evaluation order of a meta-language.

[RS-03-13] *PostScript*, *PDF*, *DVI*.

Mads Sig Ager, Dariusz Biernacki, Olivier Danvy, and Jan Midtgaard.

A Functional Correspondence between Evaluators and Abstract Machines.

March 2003.

28 pp.

Abstract: We bridge the gap between functional evaluators and abstract machines for the lambda-calculus, using closure conversion, transformation into continuation-passing style, and defunctionalization of continuations.

We illustrate this bridge by deriving Krivine's abstract machine from an ordinary call-by-name evaluator and by deriving an ordinary call-by-value evaluator from Felleisen et al.'s CEK machine. The first derivation is strikingly simpler than what can be found in the literature. The second one is new. Together, they show that Krivine's abstract machine and the CEK machine correspond to the call-by-name and call-by-value facets of an ordinary evaluator for the lambda-calculus.

We then reveal the denotational content of Hannan and Miller's CLS machine and of Landin's SECD machine. We formally compare the corresponding evaluators and we illustrate some relative degrees of freedom in the design spaces of evaluators and of abstract machines for the lambda-calculus with computational effects.

For the purpose of this work, we distinguish between virtual machines, which have an instruction set, and abstract machines, which do not. The Categorical Abstract Machine, for example, has an instruction set, but Krivine's machine, the CEK machine, the CLS machine, and the SECD machine do not; they directly operate on lambda-terms instead. We present the abstract machine that corresponds to the Categorical Abstract Machine.

[RS-03-12] *PostScript*, *PDF*, *DVI*.

Mircea-Dan Hernest and Ulrich Kohlenbach.

A Complexity Analysis of Functional Interpretations.

February 2003.

70 pp.

Abstract: We give a quantitative analysis of Gdel's functional interpretation and its monotone variant. The two have been used for the extraction of programs and numerical bounds as well as for conservation results. They apply both to (semi-)intuitionistic as well as (combined with negative translation) classical proofs. The proofs may be formalized in systems ranging from weak base systems to arithmetic and analysis (and numerous fragments of these). We give upper bounds in basic proof data on the depth, size, maximal type degree and maximal type arity of the extracted terms as well as on the depth of the verifying proof. In

all cases terms of size linear in the size of the proof at input can be extracted and the corresponding extraction algorithms have cubic worst-time complexity. The verifying proofs have depth linear in the depth of the proof at input and the maximal size of a formula of this proof.

[RS-03-11] Mads Sig Ager, Olivier Danvy, and Henning Korsholm Rohde.
Fast Partial Evaluation of Pattern Matching in Strings.
February 2003.
14 pp. This report is superseded by the later report BRICS RS-03-20.
Abstract: We show how to obtain all of Knuth, Morris, and Pratt's linear-time string matcher by partial evaluation of a quadratic-time string matcher with respect to a pattern string. Although it has been known for 15 years how to obtain this linear matcher by partial evaluation of a quadratic one, how to obtain it *in linear time* has remained an open problem.

Obtaining a linear matcher by partial evaluation of a quadratic one is achieved by performing its backtracking at specialization time and memoizing its results. We show (1) how to rewrite the source matcher such that its static intermediate computations can be shared at specialization time and (2) how to extend the memoization capabilities of a partial evaluator to static functions. Such an extended partial evaluator, if its memoization is implemented efficiently, specializes the rewritten source matcher in linear time.

[RS-03-10] *PostScript*, *PDF*.
Federico Crazzolara and Giuseppe Milicia.
Wireless Authentication in χ-Spaces.
February 2003.
20 pp.
Abstract: The χ-Spaces framework provides a set of tools to support every step of the security protocol's life-cycle. The framework includes a simple, yet powerful programming language which is an implementation of the Security Protocol Language (SPL). SPL is a formal calculus designed to model security protocols and prove interesting properties about them. In this paper we take an authentication protocol suited for low-power wireless devices and derive a χ-Spaces implementation from its SPL model. We study the correctness of the resulting implementation using the underlying SPL semantics of χ-Spaces.

[RS-03-9] *PostScript*, *PDF*.
Ivan B. Damgrd and Gudmund Skovbjerg Frandsen.
An Extended Quadratic Frobenius Primality Test with Average and Worst Case Error Estimates.
February 2003.
53 pp. Appears in Lingas and Nilsson, editors, *Fundamentals of Computation Theory: 14th International Conference*, FCT '03 Proceedings, LNCS 2751, 2003, pages 118–131.
Abstract: We present an Extended Quadratic Frobenius Primality Test (EQFT), which is related to the Miller-Rabin test and the

Quadratic Frobenius test (QFT) by Grantham. EQFT takes time about equivalent to 2 Miller-Rabin tests, but has much smaller error probability, namely $256/331776^t$ for t iterations of the test in the worst case. EQFT extends QFT by verifying additional algebraic properties related to the existence of elements of order dividing 24. We also give bounds on the average-case behaviour of the test: consider the algorithm that repeatedly chooses random odd k bit numbers, subjects them to t iterations of our test and outputs the first one found that passes all tests. We obtain numeric upper bounds for the error probability of this algorithm as well as a general closed expression bounding the error. For instance, it is at most 2^{-143} for $k = 500$.

[RS-03-8] *PostScript*, *PDF*, *DVI*.
Ivan B. Damgrd and Gudmund Skovbjerg Frandsen.
Efficient Algorithms for gcd and Cubic Residuosity in the Ring of Eisenstein Integers.
February 2003.
11 pp. Appears in Lingas and Nilsson, editors, *Fundamentals of Computation Theory: 14th International Conference*, FCT '03 Proceedings, LNCS 2751, 2003, pages 109–117.

Abstract: We present simple and efficient algorithms for computing gcd and cubic residuosity in the ring of Eisenstein integers, $\mathbf{Z}[\zeta]$, i.e. the integers extended with ζ, a complex primitive third root of unity. The algorithms are similar and may be seen as generalisations of the binary integer gcd and derived Jacobi symbol algorithms. Our algorithms take time $O(n^2)$ for n bit input. This is an improvement from the known results based on the Euclidian algorithm, and taking time $O(nM(n))$, where $M(n)$ denotes the complexity of multipliplying n bit integers. The new algorithms have applications in practical primality tests and the implementation of cryptographic protocols. The technique underlying our algorithms can be used to obtain equally fast algorithms for gcd and quartic residuosity in the ring of Gaussian integers, $\mathbf{Z}[i]$.

[RS-03-7] *PostScript*, *PDF*, *DVI*.
Claus Brabrand, Michael I. Schwartzbach, and Mads Vanggaard.
The METAFRONT System: Extensible Parsing and Transformation.
February 2003.
24 pp.

Abstract: We present the metafront tool for specifying flexible, safe, and efficient syntactic transformations between languages defined by context-free grammars. The transformations are guaranteed to terminate and to map grammatically legal input to grammatically legal output.

We rely on a novel parser algorithm that is designed to support gradual extensions of a grammar by allowing productions to remain in a natural style and by statically reporting ambiguities and errors in terms of individual productions as they are being added.

Our tool may be used as a parser generator in which the resulting parser automatically supports a flexible, safe, and efficient macro processor, or as an extensible lightweight compiler generator for domain-specific languages. We show substantial examples of both kinds.

[RS-03-6] *PostScript*, *PDF.*
Giuseppe Milicia and Vladimiro Sassone.
Jeeg: Temporal Constraints for the Synchronization of Concurrent Objects.
February 2003.
41 pp. Short version appears in Fox and Getov, editors, *Joint ACM-ISCOPE Conference on Java Grande*, JGI '02 Proceedings, 2002, pages 212–221.
Abstract: We introduce Jeeg, a dialect of Java based on a declarative replacement of the synchronization mechanisms of Java that results in a complete decoupling of the 'business' and the 'synchronization' code of classes. Synchronization constraints in Jeeg are expressed in a linear temporal logic which allows to effectively limit the occurrence of the inheritance anomaly that commonly affects concurrent object oriented languages. Jeeg is inspired by the current trend in aspect oriented languages. In a Jeeg program the sequential and concurrent aspects of object behaviors are decoupled: specified separately by the programmer these are then weaved together by the Jeeg compiler.

[RS-03-5] *PostScript*, *PDF.*
Aske Simon Christensen, Anders Mller, and Michael I. Schwartzbach.
Precise Analysis of String Expressions.
February 2003.
15 pp.
Abstract: We perform static analysis of Java programs to answer a simple question: which values may occur as results of string expressions? The answers are summarized for each expression by a regular language that is guaranteed to contain all possible values. We present several applications of this analysis, including statically checking the syntax of dynamically generated expressions, such as SQL queries. Our analysis constructs flow graphs from class files and generates a context-free grammar with a nonterminal for each string expression. The language of this grammar is then widened into a regular language through a variant of an algorithm previously used for speech recognition. The collection of resulting regular languages is compactly represented as a special kind of multi-level automaton from which individual answers may be extracted. If a program error is detected, examples of invalid strings are automatically produced. We present extensive benchmarks demonstrating that the analysis is efficient and produces results of useful precision.

[RS-03-4] *PostScript*, *PDF*, *DVI.*
Marco Carbone, Mogens Nielsen, and Vladimiro Sassone.

A Formal Model for Trust in Dynamic Networks.
January 2003.
18 pp. Appears in Cerone and Lindsay, editors, *1st International Conference on Software Engineering and Formal Methods*, SEFM '03 Proceedings, 2003, pages 54–61.
Abstract: We propose a formal model of trust informed by the Global Computing scenario and focusing on the aspects of trust formation, evolution, and propagation. The model is based on a novel notion of trust structures which, building on concepts from trust management and domain theory, feature at the same time a trust and an information partial order.

[RS-03-3] *PostScript*, *PDF*.
Claude Crpeau, Paul Dumais, Dominic Mayers, and Louis Salvail.
On the Computational Collapse of Quantum Information.
January 2003.
31 pp.
Abstract: We analyze the situation where computationally binding string commitment schemes are used to force the receiver of a BB84 encoding of a classical bitstring to measure upon reception. Since measuring induces an irreversible collapse to the received quantum state, even given extra information after the measurement does not allow the receiver to evaluate reliably some predicates apply to the classical bits encoded in the state. This fundamental quantum primitive is called quantum measure commitment (QMC) and allows for secure two-party computation of classical functions. An adversary to QMC is one that can both provide valid proof of having measured the received states while still able to evaluate a predicate applied to the classical content of the encoding. We give the first quantum black-box reduction for the security of QMC to the binding property of the string commitment. We characterize a class of quantum adversaries against QMC that can be transformed into adversaries against a weak form for the binding property of the string commitment. Our result provides a construction for $1-2$-oblivious transfer that is computationally secure against the receiver and unconditionally secure against the sender from any string commitment scheme satisfying a weak binding property.

[RS-03-2] *PostScript*, *PDF*, *DVI*.
Olivier Danvy and Pablo E. Martnez Lpez.
Tagging, Encoding, and Jones Optimality.
January 2003.
Appears in Degano, editor, *Programming Languages and Systems: Twelfth European Symposium on Programming*, ESOP '03 Proceedings, LNCS 2618, 2003, pages 335–347.
Abstract: A partial evaluator is said to be Jones-optimal if the result of specializing a self-interpreter with respect to a source program is textually identical to the source program, modulo renaming. Jones optimality has already been obtained if the self-interpreter is untyped. If the self-interpreter is typed, however,

residual programs are cluttered with type tags. To obtain the original source program, these tags must be removed.

A number of sophisticated solutions have already been proposed. We observe, however, that with a simple representation shift, ordinary partial evaluation is already Jones-optimal, modulo an encoding. The representation shift amounts to reading the type tags as constructors for higher-order abstract syntax. We substantiate our observation by considering a typed self-interpreter whose input syntax is higher-order. Specializing this interpreter with respect to a source program yields a residual program that is textually identical to the source program, modulo renaming.

[RS-03-1] *PostScript*, *PDF*, *DVI*.
Vladimiro Sassone and Paweł Sobocinski.
Deriving Bisimulation Congruences: 2-Categories vs. Precategories.
January 2003.
28 pp. Appears in Gordon, editor, *Foundations of Software Science and Computation Structures*, FoSSaCS '03 Proceedings, LNCS 2620, 2003, pages 409–424.

Abstract: G-relative pushouts (GRPOs) have recently been proposed by the authors as a new foundation for Leifer and Milner's approach to deriving labelled bisimulation congruences from reduction systems. This paper develops the theory of GRPOs further, arguing that they provide a simple and powerful basis towards a comprehensive solution. As an example, we construct GRPOs in a category of 'bunches and wirings.' We then examine the approach based on Milner's precategories and Leifer's functorial reactive systems, and show that it can be recast in a much simpler way into the 2-categorical theory of GRPOs.