BRICS Dissertation Series, Abstracts, 1996–05

January 16, 2007

This document is also available as *PostScript*, *DVI*,

References

[DS-05-9] Henning Korsholm Rohde.
Formal Aspects of Partial Evaluation.
December 2005.
PhD thesis. xii+188.
Comments: Defence: December 19, 2005.

[DS-05-8] Dariusz Biernacki.
The Theory and Practice of Programming Languages with Delimited Continuations.
December 2005.
PhD thesis.
Comments: Defence: December 15, 2005.

[DS-05-7] Kasper Dupont.
Disk Encryption, Group Identification, Byzantine Agreement, and Threshold RSA.
December 2005.
PhD thesis.
Comments: Defence: December 12, 2005.

[DS-05-6] Saurabh Agarwal.
GCD Algorithms for Quadratic Number Rings.
October 2005.
PhD thesis.
Comments: Defence: October 10, 2005.

[DS-05-5] Jesus Fernando Almansa Guerra.
A Study for Cryptologic Protocols.
September 2005.
PhD thesis.
Comments: Defence: September 12, 2005.

[DS-05-4] Bolette Ammitzbøll Madsen.
Exact Algorithms and Exact Satisfiability.
September 2005.
PhD thesis.
Comments: Defence: September 9, 2005.

[DS-05-3] Marco Carbone.
Trust and Mobility.
2005.
PhD thesis.
Comments: Defence: June 20, 2005.

[DS-05-2] Jørgen Iversen.
Formalisms and tools supporting Constructive Action Semantics.
May 2005.
PhD thesis. xii+189 pp.
Comments: Defence: May 12, 2005.

[DS-05-1] *PostScript*, *PDF*.
Kirill Morozov.
On Cryptographic Primitives Based on Noisy Channels.
March 2005.
PhD thesis. xii+102 pp.

Abstract: The primitives of Oblivious Transfer (OT) and Bit Commitment (BC) are fundamental in the cryptographic protocol design. OT is a complete primitive for secure two-party computation, while zero-knowledge proofs are based on BC. In this work, the implementations of OT and BC with unconditional security for both parties are considered. The security of these protocols does not depend on unproven intractability assumptions. Instead, we assume that the players are connected by noisy channels. This is a very natural assumption since noise is inherently present in the real communication channels.

We present and prove secure a protocol for OT based on a Discrete Memoryless Channel (DMC) with probability transition matrix of a general form. The protocol is secure for any nontrivial DMC. Some generalisations to this protocol for the particular case of Binary Symmetric Channel (BSC) are presented and their asymptotic behaviour is analysed.

The security of OT and BC based on BSC is also analysed in the non-asymptotic case. We derive relations for the failure probabilities depending on the number of channel uses establishing trade-offs between their communication complexity on the one hand and the security on the other hand.

We consider a modification to the Universally Composable (UC) framework for the case of unconditional two-party protocols. We argue that this modification is valid hereby preparing a ground for our results concerning OT based on Unfair Noisy Channels (UNC).

In contrast to the noise models mentioned above, a corrupted party is given a partial control over the randomness introduced by UNC. We introduce a generic "compiler" which transforms any protocol implementing OT from a passive version of UNC and secure against passive cheating into a protocol that uses UNC for communications and builds an OT secure against active cheating. We exploit this result and a new technique for trans-

forming between the weaker versions of OT in order to obtain new possibility results for OT based on UNC.

Comments: Defence: March 11, 2005.

[DS-04-6] *PostScript*, *PDF*.

Paweł Sobocinski.

Deriving Process Congruences from Reaction Rules.

December 2004.

PhD thesis. xii+216 pp.

Abstract: This thesis is concerned with the development of a theory which, given a formalism with a reduction semantics, allows the derivation of a canonical labelled transition system on which bisimilarity as well as other other equivalences are congruences; provided that the contexts of the formalism form a category which has certain colimits.

We shall begin by extending Leifer and Milner's theory of reactive systems to a 2-categorical setting. This development is motivated by the common situation in which the contexts of a reactive system contain non-trivial algebraic structure with an associated notion of context isomorphism. Forgetting this structure often leads to problems and we shall show that the theory can be extended smoothly, retaining this useful information as well as the congruence theorems.

Technically, the generalisation includes defining the central notion of groupoidal-relative-pushout (GRPO) (categorically: a bipushout in a pseudoslice category), which turns out to provide a suitable generalisation of Leifer and Milner's relative pushout (RPO). The congruence theorems are then reproved in this more general setting. We shall also show how previously introduced alternative solutions to the problem of forgetting the two-dimensional structure can be reduced to the 2-categorical approach.

Secondly, we shall construct GRPOs in settings which are general enough to allow the theory to be applied to useful, previously studied examples. We shall begin by constructing GRPOs in a category whose arrows correspond closely to the contexts a simple process calculus, and extend this construction further to cover the category of bunch contexts, studied previously by Leifer and Milner. The constructions use the structure of extensive categories.

We shall argue that cospans provide an interesting notion of "generalised contexts". In an effort to find a natural class of categories which allows the construction of GRPOs in the corresponding cospan bicategory, we shall introduce the class of adhesive categories. As extensive categories have well-behaved coproducts, so adhesive categories have well-behaved pushouts along monomorphisms. Adhesive categories also turn out to be a useful tool in the study and generalisation of the theory of double-pushout graph transformation systems, indeed, such systems have a rich rewriting theory when defined over adhesive

categories.

Armed with the theory of adhesive categories, we shall present a construction of GRPOs in input-linear cospan bicategories. As an immediate application, we shall shed light on as well as extend the theory of rewriting via borrowed contexts, due to Ehrig and König. Secondly, we shall examine the implications of the construction for Milner's bigraphs.

Comments: Defence: December 3, 2004.

[DS-04-5] *PostScript*, *PDF*.

Bjarke Skjernaa.

Exact Algorithms for Variants of Satisfiability and Colouring Problems.

November 2004.

PhD thesis. x+112 pp.

Abstract: This dissertation studies exact algorithms for various hard problems and give an overview of not only results but also the techniques used.

In the first part we focus on Satisfiability and several variants of Satisfiability. We present some historical techniques and results. Our main focus in the first part is on a particular variant of Satisfiability, Exact Satisfiability. Exact Satisfiability is the variant of Satisfiability where a clause is satisfied if it contains exactly one *true* literal. We present an algorithm solving Exact Satisfiability in time $O(2^{0.2325n})$, and an algorithm solving Exact 3-Satisfiability, the variant of Exact Satisfiability where each clause contains at most three literals, in time $O(2^{0.1379n})$. In these algorithms we use a new concept of *k-sparse* formulas, and we present a new technique called *splitting loosely connected formulas*, although we do not use the technique in the algorithms.

We present a new program which generates algorithms and corresponding upper bound proofs for variants of Satisfiability, and in particular Exact Satisfiability. The program uses several new techniques which we describe and compare to the techniques used in three other programs generating algorithms and upper bound proofs.

In the second part we focus on another class of NP-complete problems, colouring problems. We present several algorithms for 3-Colouring, and discuss k-Colouring in general. We also look at the problem of determining the chromatic number of a graph, which is the minimum number, k, such that the graph is k-colourable. We present a technique using maximal bipartite subgraphs to solve k-Colouring, and prove two bounds on the maximum possible number of maximal bipartite subgraphs of a graph with n vertices: a lower bound of $\Omega(1.5926^n)$ and an upper bound of $O(1.8612^n)$. We also present a recent improvement of the upper bound by Byskov and Eppstein, and show a number of applications of this in colouring problems.

In the last part of this dissertation we look at a class of prob-

lems unrelated to the above, Graph Distinguishability problems. DIST_k is the problem of determining if a graph can be coloured with k colours, such that no non-trivial automorphism of the graph preserves the colouring. We present some results on determining the distinguishing number of a graph, which is the minimum k, such that the graph is in DIST_k. We finish by proving a new result which shows that DIST_k can be reduced to DIST_l for various values of k and l.

Comments: Defence: November 11, 2004.

[DS-04-4] Jesper Makholm Byskov.
Exact Algorithms for Graph Colouring and Exact Satisfiability.
November 2004.
PhD thesis.

Abstract: This dissertation contains results within the field of exact algorithms for NP-hard problems. We give an overview of the major results within the field as well as the main techniques used. Then we present new exact algorithms for various graph colouring problems and for two variants of EXACT SATISFIABILITY (XSAT).

In graph colouring, we first look at the problem of enumerating all maximal independent sets of a graph. This is not an NP-hard problem, but algorithms for this problem are useful for solving other graph colouring problems. Moon and Moser (1965) have given tight upper bounds on the number of maximal independent sets in a graph, and there exists a number of algorithms for enumerating them all. We look at the extended problems of enumerating all maximal independent sets of size equal to k, of size at most k and of size at least k for $1 \leq k \leq n$. We give tight upper bounds on the number of such sets and give algorithms for enumerating them all in time within a polynomial factor of these upper bounds, thereby extending earlier work of Eppstein (2003a). Then we look at algorithms for enumerating all maximal bipartite subgraphs of a graph. We give upper and lower bounds on the number of maximal bipartite subgraphs and an algorithm for enumerating all such subgraphs, but the bounds are not tight. The work on maximal bipartite subgraphs is from two different papers, one with B. A. Madsen and B. Skjernaa and the other with D. Eppstein. Finally, we give algorithms for enumerating all maximal k-colourable subgraphs of a graph for $k \geq 3$. These algorithms all run in time $\omega(2^n)$ and we have no $o(2^n)$ upper bound on the number of maximal k-colourable subgraphs for $k \geq 3$. The algorithm for $k = 3$ is from joint work with D. Eppstein. We use all the above-mentioned algorithms in new algorithms for 4-COLOURING, 5-COLOURING and 6-COLOURING using ideas of Lawler (1976). Some of these algorithms are also joint work with D. Eppstein. We also improve an algorithm of Eppstein (2003a) for the CHROMATIC NUMBER problem.

We also give new, improved algorithms for XSAT as well as X3SAT, the variant where each clause contains at most three

literals. Our improvements come from three sources. First, we discover new reduction rules which allow us to get rid of more formulas by replacing them by simpler formulas. Next, we extend the case analyses of previous algorithms to distinguish between more cases. Finally, we introduce a new concept called sparse formulas. XSAT can be solved in polynomial time if no variable occurs more than twice in the formula. We call a formula sparse, if the number of variables occurring more than twice is small. Such formulas we solve by trying all possible truth assignments to the variables occurring more than twice. Then the remaining parts of the formulas contain only variables occurring at most twice, so they can be solved in polynomial time. Both algorithms are joint work with B. A. Madsen and B. Skjernaa.

Comments: Defence: November 11, 2004.

[DS-04-3] *PostScript*, *PDF*.
Jens Groth.
Honest Verifier Zero-knowledge Arguments Applied.
October 2004.
PhD thesis. xii+119 pp.

Abstract: We apply honest verifier zero-knowledge arguments to cryptographic protocols for non-malleable commitment, voting, anonymization and group signatures. For the latter three the random oracle model can be used to make them non-interactive. Unfortunately, the random oracle model is in some ways unreasonable, we present techniques to reduce the dependence on it.

Several voting schemes have been suggested in the literature, among them a class of efficient protocols based on homomorphic threshold encryption. Voters encrypt their votes, and using the homomorphic property of the cryptosystem we can get an encryption of the result. A group of authorities now makes a threshold decryption of this ciphertext to get the result. We have to protect against voters that cheat by encrypting something that is not a valid vote. We therefore require that they make a non-interactive zero-knowledge argument of correctness of the encrypted vote. Using integer-commitments, we suggest efficient honest verifier zero-knowledge arguments for correctness of a vote, which can be made non-interactive using the Fiat-Shamir heuristic. We suggest arguments for single-voting, multi-way voting, approval voting and shareholder voting, going from the case where the voter has a single vote to the case where a voter may cast millions of votes at once.

For anonymization purposes, one can use a mix-net. One way to construct a mix-net is to let a set of mixers take turns in permuting and reencrypting or decrypting the inputs. If at least one of the mixers is honest, the input data and the output data can no longer be linked. For this anonymization guarantee to hold we do need to ensure that all mixers act according to the protocol. For use in reencrypting mix-nets we suggest an honest verifier zero-knowledge argument of a correct reencrypting shuffle that can be used with a large class of homomorphic cryptosystems. For use in decrypting mix-nets, we offer an honest verifier zero-knowledge argument for a decrypting shuffle.

For any NP-language there exists, based on one-way functions, a 3-move honest verifier zero-knowledge argument. If we pick a suitable hard statement and a suitable 3-move argument for this statement, we can use the protocol as a commitment scheme. A commitment to a message is created by using a special honest verifier zero-knowledge property to simulate an argument with the message as challenge. We suggest picking the statement to be proved in a particular way related to a signature scheme, which is secure against known message attack. This way we can create many commitments that can be trapdoor opened, yet other commitments chosen by an adversary remain binding. This property is known as simulation soundness. Simulation soundness implies non-malleability, so we obtain a non-malleable commitment scheme based on any one-way function. We also obtain efficient non-malleable commitment schemes based on the strong RSA assumption.

We investigate the use of honest verifier zero-knowledge arguments in signature schemes. By carefully selecting the statement to be proven and finding an efficient protocol, we obtain a group signature scheme that is very efficient and has only weak set-up assumptions. Security is proved in the random oracle model. The group signature scheme can be extended to allow for dynamic join of members, revocation of members and can easily be transformed into a traceable signature scheme. We observe that traceable signatures can be built from one-way functions and non-interactive zero-knowledge arguments, which seems to be weaker than the assumptions needed to build group signatures.

Comments: Defence: October 15, 2004.

[DS-04-2] Alex Rune Berg.
Rigidity of Frameworks and Connectivity of Graphs.
July 2004.
PhD thesis. xii+173 pp.

Abstract: This dissertation contains results on three different but connected areas. The first area is on rigidity of frameworks. A framework consists of rigid bars and rotatable joints. A framework is rigid if it has no deformation. A graph is generically rigid if every framework obtained as a 'generic embedding' of the graph in the plane is rigid. A graph is redundantly rigid if it is generically rigid after removing an arbitrary edge. A graph $G = (V, E)$ is called a **generic circuit** if $|E| = 2|V| - 2$ and G is redundantly rigid. The operation **extension** subdivides an edge uw of a graph by a new vertex v and adds a new edge vz for some vertex $z \neq u, w$. R. Connelly conjectured that every 3-connected generic circuit can be obtained from K_4 by a sequence of extensions. We prove this conjecture. All new results

presented in this dissertation is joint work with T. Jordán. A new and simple algorithm for finding the maximal generically rigid subgraphs of a graph has also been developed. The running time matches that of previous algorithms, namely $O(n^2)$. Furthermore a polynomial algorithm has been found for computing maximal generically uniquely realizable graphs in dimension two (also called globally rigid graphs). The algorithm relies heavily on a result of Jackson and Jordán.

Theorems on detachments in k-edge-connected directed graphs and splitting off in directed hypergraphs have been obtained. The work described in this paragraph is also joint with B. Jackson. The splitting off operation in a directed graph replaces an arc $\overrightarrow{su}$ and arcs $\overrightarrow{v_1s}, \ldots, \overrightarrow{v_ts}$ by one directed hyperedge having tail vertices $v_1, \ldots, v_t$ and one head vertex u. We present a theorem detaching hyperedges preserving k-edge-connectivity of the resulting directed hypergraph (for an appropriate definition of k-edge-connectivity). The detachment operation in a directed graph 'detaches' a vertex s into several vertices ('pieces') $s_1, \ldots, s_t$ and then distributes the arcs of s among the vertices $s_1, \ldots, s_t$ and finally deletes s. We present a theorem stating when it is possible to detach a vertex into r pieces preserving k-edge-connectivity of the resulting graph. This detachment theorem correspond to a theorem by Nash-Williams on undirected graphs.

Orientation results preserving k-edge-connectivity have been developed by Nash-Williams. Frank conjectured that a graph $G = (V, E)$, $|V| \geq k + 1$, has a k-vertex-connected orientation if and only if $G - X$ is $(2k - 2|X|)$-edge-connected for all $X \subseteq V$ with $|X| \leq k - 1$. Gerards verified Frank's conjecture graphs G where every vertex has degree four and with $k = 2$. In this dissertation Frank's conjecture is verified for the Eulerian graphs and still with $k = 2$. An algorithm is presented for finding 2-vertex-connected orientations of Eulerian graphs satisfying Frank's conjecture for $k = 2$.

Comments: Defence: July 7, 2004.

[DS-04-1] *PostScript*, *PDF*.

Bartosz Klin.

An Abstract Coalgebraic Approach to Process Equivalence for Well-Behaved Operational Semantics.

May 2004.

PhD thesis. x+152 pp.

Abstract: This thesis is part of the programme aimed at finding a mathematical theory of well-behaved structural operational semantics. General and basic results shown in 1997 in a seminal paper by Turi and Plotkin are extended in two directions, aiming at greater expressivity of the framework.

The so-called bialgebraic framework of Turi and Plotkin is an abstract generalization of the well-known structural operational semantics format GSOS, and provides a theory of operational semantic rules for which bisimulation equivalence is a congruence.

The first part of this thesis aims at extending that framework to cover other operational equivalences and preorders (e.g. trace equivalence), known collectively as the van Glabbeek spectrum. To do this, a novel coalgebraic approach to relations on processes is desirable, since the usual approach to coalgebraic bisimulations as spans of coalgebras does not extend easily to other known equivalences on processes. Such an approach, based on fibrations of test suites, is presented. Based on this, an abstract characterization of congruence formats is given, parametrized by the relation on processes that is expected to be compositional. This abstract characterization is then specialized to the case of trace equivalence, completed trace equivalence and failures equivalence. In the two latter cases, novel congruence formats are obtained, extending the current state of the art in this area of research.

The second part of the thesis aims at extending the bialgebraic framework to cover a general class of recursive language constructs, defined by (possibly unguarded) recursive equations. Since unguarded equations may be a source of divergence, the entire framework is interpreted in a suitable domain category, instead of the category of sets and functions. It is shown that a class of recursive equations called regular equations can be merged seamlessly with GSOS operational rules, yielding well-behaved operational semantics for languages extended with recursive constructs.

Comments: Defence: May 28, 2004.

[DS-03-14] *PostScript*, *PDF*.
Daniele Varacca.
Probability, Nondeterminism and Concurrency: Two Denotational Models for Probabilistic Computation.
November 2003.
PhD thesis. xii+163 pp.

Abstract: Nondeterminism is modelled in domain theory by the notion of a powerdomain, while probability is modelled by that of the probabilistic powerdomain. Some problems arise when we want to combine them in order to model computation in which both nondeterminism and probability are present. In particular there is no categorical *distributive law* between them. We introduce the *powerdomain of indexed valuations* which modifies the usual probabilistic powerdomain to take more detailed account of where probabilistic choices are made. We show the existence of a distributive law between the powerdomain of indexed valuations and the nondeterministic powerdomain. By means of an equational theory we give an alternative characterisation of indexed valuations and the distributive law. We study the relation between valuations and indexed valuations. Finally we use indexed valuations to give a semantics to a programming language. This semantics reveals the computational intuition lying behind the mathematics.

In the second part of the thesis we provide an operational reading of continuous valuations on certain domains (the distributive concrete domains of Kahn and Plotkin) through the model of *probabilistic event structures*. Event structures are a model for concurrent computation that account for causal relations between events. We propose a way of adding probabilities to confusion free event structures, defining the notion of probabilistic event structure. This leads to various ideas of a run for probabilistic event structures. We show a confluence theorem for such runs. Configurations of a confusion free event structure form a distributive concrete domain. We give a representation theorem which characterises completely the powerdomain of valuations of such concrete domains in terms of probabilistic event structures.

Comments: Defence: November 24, 2003.

[DS-03-13] *PostScript*, *PDF*.
Mikkel Nygaard.
Domain Theory for Concurrency.
November 2003.
PhD thesis. xiii+161 pp.

Abstract: Concurrent computation can be given an abstract mathematical treatment very similar to that provided for sequential computation by domain theory and denotational semantics of Scott and Strachey.

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, called HOPLA for Higher-Order Process LAnguage, derives from an exponential of linear logic. It can be viewed as an extension of the simply-typed lambda calculus with CCS-like nondeterministic sum and prefix operations, in which types express the form of computation path of which a process is capable. HOPLA can directly encode calculi like CCS, CCS with process passing, and mobile ambients with public names, and it can be given a straightforward operational semantics supporting a standard bisimulation congruence. The denotational and operational semantics are related with simple proofs of soundness and adequacy. Full abstraction implies that contextual equivalence coincides with logical equivalence for a fragment of Hennessy-Milner logic, linking up with simulation equivalence.

The other language is called Affine HOPLA and is based on a weakening comonad that yields a model of affine-linear logic. This language adds to HOPLA an interesting tensor operation at the price of linearity constraints on the occurrences of variables. The tensor can be understood as a juxtaposition of independent processes, and allows Affine HOPLA to encode processes of the

kind found in treatments of nondeterministic dataflow.

The domain theory can be generalised to presheaf models, providing a more refined treatment of nondeterministic branching and supporting notions of bisimulation. The operational semantics for HOPLA is guided by the idea that derivations of transitions in the operational semantics should correspond to elements of the presheaf denotations. Similar guidelines lead to an operational semantics for the first-order fragment of Affine HOPLA. An extension of the operational semantics to the full language is based on a stable denotational semantics which associates to each computation the minimal input necessary for it. Such a semantics is provided, based on event structures; it agrees with the presheaf semantics at first order and exposes the tensor operation as a simple parallel composition of event structures.

The categorical model obtained from presheaves is very rich in structure and points towards more expressive languages than HOPLA and Affine HOPLA—in particular concerning extensions to cover independence models. The thesis concludes with a discussion of related work towards a fully fledged domain theory for concurrency.

Comments: Defence: November 21, 2003.

[DS-03-12] *PostScript*, *PDF*.
Paulo B. Oliva.
Proof Mining in Subsystems of Analysis.
September 2003.
PhD thesis. xii+198 pp.

Abstract: This dissertation studies the use of methods of proof theory in extracting new information from proofs in subsystems of classical analysis. We focus mainly on ineffective proofs, i.e. proofs which make use of ineffective principles ranging from weak König's lemma to full comprehension. The main contributions of the dissertation can be divided into four parts:

1. A discussion of how monotone functional interpretation provides the right notion of "numerical implication" in analysis. We show among other things that monotone functional interpretation naturally creates well-studied moduli when applied to various classes of statements (e.g. uniqueness, convexity, contractivity, continuity and monotonicity) and that the interpretation of implications between those statements corresponds to translations between the different moduli.

2. A case study in L_1-approximation, in which we analyze Cheney's proof of Jackson's theorem, concerning uniqueness of the best approximation, w.r.t. L_1-norm, of continuous functions $f \in C[0,1]$ by polynomials of bounded degree. The result of our analysis provides the first effective modulus of uniqueness for L_1-approximation. Moreover, using this modulus we give the first complexity analysis of the se-

quence of best L_1-approximations for polynomial-time computable functions $f \in C[0,1]$.

3. A comparison between three different forms of bar recursion, in which we show among other things that the type structure of strongly majorizable functionals is a model of modified bar recursion, that modified bar recursion is not S1–S9 computable over the type structure of total continuous functions and that modified bar recursion de nes (primitive recursively, in the sense of Kleene) Spector's bar recursion.

4. An adaptation of functional interpretation to handle ine ective proofs in feasible analysis, which provides the first modular procedure for extracting polynomial-time realizers from ineffective proofs (i.e. proofs involving weak König's lemma) of $\prod_2^0$-theorems in feasible analysis.

.

Comments: Defence: September 26, 2003.

[DS-03-11] Maciej Koprowski.
Cryptographic Protocols Based on Root Extracting.
August 2003.
PhD thesis. xii+138 pp.

Abstract: In this thesis we design new cryptographic protocols, whose security is based on the hardness of root extracting or more speci cally the RSA problem.

First we study the problem of root extraction in nite Abelian groups, where the group order is unknown. This is a natural generalization of the, heavily employed in modern cryptography, RSA problem of decrypting RSA ciphertexts. We examine the complexity of this problem for generic algorithms, that is, algorithms that work for any group and do not use any special properties of the group at hand. We prove an exponential lower bound on the generic complexity of root extraction, even if the algorithm can choose the "public exponent" itself. In other words, both the standard and the strong RSA assumption are provably true w.r.t. generic algorithms. The results hold for arbitrary groups, so security w.r.t. generic attacks follows for any cryptographic construction based on root extracting.

As an example of this, we modify Cramer-Shoup signature scheme such that it becomes a generic algorithm. We discuss then implementing it in RSA groups without the original restriction that the modulus must be a product of safe primes. It can also be implemented in class groups. In all cases, security follows from a well de ned complexity assumption (the strong root assumption), without relying on random oracles.

A *smooth* natural number has no big prime factors. The probability, that a random natural number not greater than x has all prime factors smaller than $x^{1/u}$ with $u > 1$, is asymptotically close to the Dickman function $\rho(u)$. By careful analysis of

Hildebrand's asymptotic smoothness results and applying new techniques, we derive concrete bounds on the smoothness probabilities assuming the Riemann hypothesis. We show how our results can be applied in order to suggest realistic values of security parameters in Gennaro-Halevi-Rabin signature scheme.

We introduce a modi ed version of Fischlin's signature scheme where we substitute prime exponents with random numbers. Equipped with our exact smoothness bounds, we can compute concrete sizes of security parameters, providing a currently acceptable level of security.

This allows us to propose the rst practical blind signature scheme provably secure, without relying on heuristics called random oracle model (ROM). We obtain the protocol for issuing blind signatures by implementing our modi ed Fischlin's signing algorithm as a secure two-party computation. The security of our scheme assumes the hardness of the strong RSA problem and decisional composite residuosity.

We discuss a security aw in the e cient statefull variant of Fischlin's signature scheme.

Next, we introduce a threshold RSA scheme which is almost as e cient as the fastest previous threshold RSA scheme (by Shoup), but where two assumptions needed in Shoup's and in previous schemes can be dropped, namely that the modulus must be a product of safe primes and that a trusted dealer generates the keys. The robustness (but not the unforgeability) of our scheme depends on a new intractability assumption, in addition to security of the underlying standard RSA scheme.

Finally, we propose the concept of ne-grained forward secure signature schemes, which not only provides non-repudiation w.r.t. past time periods the way ordinary forward secure signature schemes do, but in addition allows the signer to specify which signatures of the current time period remain valid in case the public key needs to be revoked. This is an important advantage if the signer processes many signatures per time period as otherwise the signer would have to re-issue those signatures (and possibly re-negotiate the respective messages) with a new key. Apart from a formal model we present practical instantiations and prove them secure under the strong RSA assumption only, i.e., we do not resort to the random oracle model to prove security. As a sideresult, we provide an ordinary forward-secure scheme whose key-update time is signi cantly smaller than for other known schemes that do not assume random oracle model (ROM).

Comments: Defence: August 8, 2003.

[DS-03-10] *PostScript*, *PDF*.
Serge Fehr.
Secure Multi-Player Protocols: Fundamentals, Generality, and Efficiency.

August 2003.
PhD thesis. xii+125 pp.

Abstract: While classically cryptography is concerned with the problem of private communication among two entities, say *players*, in modern cryptography *multi*-player protocols play an important role. And among these, it is probably fair to say that *secret sharing*, and its stronger version *verifiable secret sharing (VSS)*, as well as *multi-party computation* (MPC) belong to the most appealing and/or useful ones. The former two are basic tools to achieve better robustness of cryptographic schemes against malfunction or misuse by "decentralizing" the security from one single to a whole group of individuals (captured by the term *threshold* cryptography). The latter allows—at least in principle—to execute *any* collaboration among a group of players in a *secure* way that guarantees the correctness of the outcome but simultaneously respects the privacy of the participants.

In this work, we study three aspects of secret sharing, VSS and MPC, which we denote by *fundamentals*, *generality*, and *efficiency*. By fundamentals we mean the quest for understanding *why* a protocol works and is secure in abstract (and hopefully simple) mathematical terms. By generality we mean generality with respect to the underlying mathematical structure, in other words, minimizing the mathematical *axioms* required to do some task. And efficiency of course deals with the improvement of protocols with respect to some meaningful complexity measure.

We briefly summarize our main results. (1) We give a complete characterization of *black-box* secret sharing in terms of simple algebraic conditions on the integer sharing coefficients, and we propose a black-box secret sharing scheme with minimal expansion factor. Note that, in contrast to the classical field-based secret sharing schemes, a black-box secret sharing scheme allows to share a secret sampled from an arbitrary Abelian group and requires only black-box access to the group operations and to random group elements. Such a scheme may be very useful in the construction of threshold cryptosystems based on groups with secret order (like RSA). (2) We show that without loss of efficiency, MPC can be based on arbitrary finite rings. This is in sharp contrast to the literature where essentially all MPC protocols require a much stronger mathematical structure, namely a field. Apart from its theoretical value, this can lead to efficiency improvements since it allows a greater freedom in the (mathematical) representation of the task that needs to be securely executed. (3) We propose a unified treatment of perfectly secure linear VSS and distributed commitments (a weaker version of the former), and we show that the security of such a scheme can be reduced to a linear algebra condition. The security of all known schemes follows as corollaries whose proofs are pure linear algebra arguments, in contrast to some hybrid arguments used in the literature. (4) We construct a new unconditionally secure

VSS scheme with minimized reconstruction complexity in the setting of a dishonest minority. This improves on the reconstruction complexity of the previously best known scheme without affecting the (asymptotic) complexity of the sharing phase. In the context of MPC with *pre-processing*, our scheme allows to improve the computation phase of earlier protocols without increasing the complexity of the pre-processing. (5) Finally, we consider *commitment multiplication proofs*, which allow to prove a multiplicative relation among three commitments, and which play a crucial role in computationally secure MPC. We study a *non-interactive* solution which works in a *distributed-verifier* setting and essentially consists of a few executions of Pedersen's VSS. We simplify and improve the protocol, and we point out a (previously overlooked) property which helps to construct non-interactive proofs of *partial knowledge* in this setting. This allows for instance to prove the knowledge of ℓ out of m given secrets, without revealing which ones. We also construct a non-interactive distributed-verifier proof of *circuit satisfiability*, which—in principle—allows to prove *anything* that can be proven without giving away the proof.

Comments: Defence: August 8, 2003.

[DS-03-9] *PostScript*, *PDF*.

Mads J. Jurik.

Extensions to the Paillier Cryptosystem with Applications to Cryptological Protocols.

August 2003.

PhD thesis. xii+117 pp.

Abstract: The main contribution of this thesis is a simplification, a generalization and some modifications of the homomorphic cryptosystem proposed by Paillier in 1999, and several cryptological protocols that follow from these changes.

The Paillier cryptosystem is an additive homomorphic cryptosystem, meaning that one can combine ciphertexts into a new ciphertext that is the encryption of the sum of the messages of the original ciphertexts. The cryptosystem uses arithmetic over the group $\mathbf{Z}^*_{n^2}$ and the cryptosystem can encrypt messages from the group $\mathbf{Z}_n$. In this thesis the cryptosystem is generalized to work over the group $\mathbf{Z}^*_{n^{s+1}}$ for any integer $s > 0$ with plaintexts from the group $\mathbf{Z}_{n^s}$. This has the advantage that the ciphertext is only a factor of $(s+1)/s$ longer than the plaintext, which is an improvement to the factor of 2 in the Paillier cryptosystem. The generalized cryptosystem is also simplified in some ways, which results in a threshold decryption that is conceptually simpler than other proposals. Another cryptosystem is also proposed that is length-flexible, i.e. given a fixed public key, the sender can choose the s when the message is encrypted and use the message space of $\mathbf{Z}_{n^s}$. This new system is modified using some El Gamal elements to create a cryptosystem that is both length-flexible and has an efficient threshold decryption. This new system has

the added feature, that with a globally setup RSA modulus n, provers can efficiently prove various relations on plaintexts inside ciphertexts made using different public keys. Using these cryptosystems several multi-party protocols are proposed:

- A mix-net, which is a tool for making an unknown random permutation of a list of ciphertext. This makes it a useful tool for achieving anonymity.
- Several voting systems:
  - An efficient large scale election system capable of handling large elections with many candidates.
  - Client/server trade-offs: 1) a system where vote size is within a constant of the minimal size, and 2) a system where a voter is protected even when voting from a hostile environment (i.e. a Trojan infested computer). Both of these improvements are achieved at the cost of some extra computations at the server side.
  - A small scale election with perfect ballot secrecy (i.e. any group of persons only learns what follows directly from their votes and the final result) usable e.g. for board room election.
- A key escrow system, which allows an observer to decrypt any message sent using any public key set up in the defined way. This is achieved even though the servers only store a constant amount of key material.

The last contribution of this thesis is a petition system based on the modified Weil pairing. This system greatly improves the naive implementations using normal signatures from using an order of $\mathcal{O}(tk)$ group operations to using only $\mathcal{O}(t+k)$, where t is the number of signatures checked, and k is the security parameter.

Comments: Defence: August 7, 2003.

[DS-03-8] *PostScript*, *PDF*.
Jesper Buus Nielsen.
On Protocol Security in the Cryptographic Model.
August 2003.
PhD thesis. xiv+341 pp.

Abstract: It seems to be a generally acknowledged fact that you should never trust a computer and that you should trust the person operating the computer even less. This in particular becomes a problem when the party that you do not trust is one which is separated from you and is one on which you depend, e.g. because that party is the holder of some data or some capability that you need in order to operate correctly. How do you perform a given task involving interaction with other parties while allowing these parties a minimal influence on you and, if privacy is an issue, allowing them to learn as little about you as possible. This is the general problem of secure *multiparty computation*. The usual way of formalizing the problem is to say that a number of parties who do not trust each other wish to compute some function of their

local inputs, while keeping their inputs as secret as possible and guaranteeing the correctness of the output. Both goals should be obtained even if some parties stop participating or some malevolent coalition of the parties start deviating arbitrarily from the agreed protocol. The task is further complicated by the fact that besides their mutual distrust, nor do the parties trust the channels by which they communicate. A general solution to the secure multiparty computation problem is a compiler which given any feasible function describes an efficient protocol which allows the parties to compute the function securely on their local inputs over an open network.

Over the past twenty years the secure multiparty computation problem has been the subject of a large body of research, both research into the models of multiparty computation and research aimed at realizing general secure multiparty computation. The main approach to realizing secure multiparty computation has been based on verifiable secret sharing as computation, where each party holds a secret share of each input and during the execution computes secret shares of all intermediate values. This approach allows the parties to keep all inputs and intermediate values secret and to pool the shares of the output values to learn exactly these values.

More recently an approach based on joint homomorphic encryption was introduced, allowing for an efficient realization of general multiparty computation secure against an eavesdropper. A joint encryption scheme is an encryption scheme where all parties can encrypt, but where it takes the participation of all parties to open an encryption. The computation then starts by all parties broadcasting encryptions of their inputs and progresses through computing encryptions of the intermediary values using the homomorphic properties of the encryption scheme. The encryptions of the outputs can then be jointly decrypted.

In this dissertation we extend this approach by using threshold homomorphic encryption to provide full-fledged security against an active attacker which might control some of the parties and which might take over parties during the course of the protocol execution. As opposed to a joint encryption scheme a threshold encryption scheme only requires that a given number of the parties are operating correctly for decryption to be possible. In this dissertation we show that threshold homomorphic encryption allows to solve the secure general multiparty computation problem more efficiently than previous approaches to the problem.

Starting from an open point-to-point network there is a long way to general secure multiparty computation. The dissertation contains contributions at several points along the way. In particular we investigate how to realize *secure channels*. We also show how threshold signature schemes can be used to efficiently realize a *broadcast channel* and how threshold cryptography can be used to provide the parties of the network with a source of *shared ran-*

domness. All three tools are well-known to be extremely powerful resources in a network, and our principle contribution is to provide efficient realizations.

The dissertation also contains contributions to the definitional part of the field of multiparty computation. Most significantly we augment the recent model of *universally composable security* with a formal notion of what it means for a protocol to only realize a given problem under a given *restricted input-output behavior* of the environment. This e.g. allows us to give the first specification of a *universally composable zero-knowledge proof of membership* which does not at the same time require the proof system to be a proof of knowledge.

Comments: Defence: August 7, 2003.

[DS-03-7] *PostScript*, *PDF*.
Mario José Cáccamo.
A Formal Calculus for Categories.
June 2003.
PhD thesis. xiv+151.

Abstract: This dissertation studies the logic underlying category theory. In particular we present a formal calculus for reasoning about universal properties. The aim is to systematise judgements about *functoriality* and *naturality* central to categorical reasoning. The calculus is based on a language which extends the typed lambda calculus with new binders to represent universal constructions. The types of the languages are interpreted as locally small categories and the expressions represent functors.

The logic supports a syntactic treatment of universality and duality. Contravariance requires a definition of universality generous enough to deal with functors of mixed variance. *Ends* generalise limits to cover these kinds of functors and moreover provide the basis for a very convenient algebraic manipulation of expressions.

The equational theory of the lambda calculus is extended with new rules for the definitions of universal properties. These judgements express the existence of natural isomorphisms between functors. The isomorphisms allow us to formalise in the calculus results like the Yoneda lemma and Fubini theorem for ends. The definitions of limits and ends are given as *representations* for special **Set**-valued functors where **Set** is the category of sets and functions. This establishes the basis for a more calculational presentation of category theory rather than the traditional diagrammatic one.

The presence of structural rules as primitive in the calculus together with the rule for duality give rise to issues concerning the *coherence* of the system. As for every well-typed expression-in-context there are several possible derivations it is sensible to ask whether they result in the same interpretation. For the functoriality judgements the system is coherent up to natural isomorphism. In the case of naturality judgements a simple example shows its incoherence. However in many situations to know there

exists a natural isomorphism is enough. As one of the contributions in this dissertation, the calculus provides a useful tool to verify that a functor is continuous by just establishing the existence of a certain natural isomorphism.

Finally, we investigate how to generalise the calculus to enriched categories. Here we lose the ability to manipulate contexts through weakening and contraction and conical limits are not longer adequate.

Comments: Defence: June 23, 2003.

[DS-03-6] *PostScript*, *PDF*.

Rasmus K. Ursem.

Models for Evolutionary Algorithms and Their Applications in System Identification and Control Optimization.

June 2003.

PhD thesis. xiv+183 pp.

Abstract: In recent years, optimization algorithms have received increasing attention by the research community as well as the industry. In the area of evolutionary computation (EC), inspiration for optimization algorithms originates in Darwin's ideas of evolution and survival of the fittest. Such algorithms simulate an evolutionary process where the goal is to evolve solutions by means of crossover, mutation, and selection based on their quality (fitness) with respect to the optimization problem at hand. Evolutionary algorithms (EAs) are highly relevant for industrial applications, because they are capable of handling problems with non-linear constraints, multiple objectives, and dynamic components — properties that frequently appear in real-world problems.

This thesis presents research in three fundamental areas of EC; fitness function design, methods for parameter control, and techniques for multimodal optimization. In addition to general investigations in these areas, I introduce a number of algorithms and demonstrate their potential on real-world problems in system identification and control. Furthermore, I investigate dynamic optimization problems in the context of the three fundamental areas as well as control, which is a field where real-world dynamic problems appear.

Regarding fitness function design, smoothness of the fitness landscape is of primary concern, because a too rugged landscape may disrupt the search and lead to premature convergence at local optima. Rugged fitness landscapes typically arise from imprecisions in the fitness calculation or low relatedness between neighboring solutions in the search space. The imprecision problem was investigated on the Runge-Kutta-Fehlberg numerical integrator in the context of non-linear differential equations. Regarding the relatedness problem for the search space of arithmetic functions, Thiemo Krink and I suggested the smooth operator genetic programming algorithm. This approach improves the smoothness of fitness function by allowing a gradual change between traditional

operators such as multiplication and division.

In the area of parameter control, I investigated the so-called self-adaptation technique on dynamic problems. In self-adaptation, the genome of the individual contains the parameters that are used to modify the individual. Self-adaptation was developed for static problems; however, the parameter control approach requires a significant number of generations before superior parameters are evolved. In my study, I experimented with two artificial dynamic problems and showed that the technique fails on even rather simple time-varying problems. In a different study on static problems, Thiemo Krink and I suggested the terrain-based patchwork model, which is a fundamentally new approach to parameter control based on agents moving in a spatial grid world.

For multimodal optimization problems, algorithms are typically designed with two objectives in mind. First, the algorithm shall find the global optimum and avoid stagnation at local optima. Additionally, the algorithm shall preferably find several candidate solutions, and thereby allow a final human decision among the found solutions. For this objective, I created the multinational EA that employs a self-organizing population structure grouping the individuals into a number of subpopulations located in different parts of the search space. In a related study, I investigated the robustness of the widely used sharing technique. Surprisingly, I found that this algorithm is extremely sensitive to the range of fitness values. In a third investigation, I introduced the diversity-guided EA, which uses a population diversity measure to guide the search. The potential of this algorithm was demonstrated on parameter identification of two induction motor models, which are used in the pumps produced by the Danish pump manufacturer Grundfos.

The field of dynamic optimization has received significant attention since 1990. However, most research performed in an EC-context has focused on artificial dynamic problems. In a fundamental study, Thiemo Krink, Mikkel T. Jensen, Zbigniew Michalewicz, and I investigated artificial dynamic problems and found no clear connection (if any) to real-world dynamic problems. In conclusion, a large part of this research field's foundation, i.e., the test problems, is highly questionable. In continuation of this, Thiemo Krink, Bogdan Filipič, and I investigated online control problems, which have the special property that the search changes the problem. In this context, we examined how to utilize the available computation time in the best way between updates of the control signals. For an EA, the available time can be a trade-off between population size and number of generations. From our experiments, we concluded that the best approach was to have a small population and many generations, which essentially turns the problem into a series of related static problems. To our surprise, the control problem could easily be solved when optimized like this. To further examine this, we compared the EA with a particle swarm and a local search approach, which we developed for dynamic optimization in general. The three algorithms had matching performance when properly tuned. An interesting result from this investigation was that different step-sizes in the local search algorithm induced different control strategies, i.e., the search strategy lead to the emergence of alternative paths of the moving optima in the dynamic landscape. This observation is captured in the novel concept of *optima in time*, which we introduced as a temporal version of the usual optima the in search space.

Comments: Defence: June 20, 2003.

[DS-03-5] Giuseppe Milicia.

Applying Formal Methods to Programming Language Design and Implementation.

June 2003.

PhD thesis. xvi+211.

Abstract: This thesis concerns the design and evolution of two programming languages. The two contributions, unrelated with each other, are treated separately. The underlying theme of this work is the application of *formal methods* to the task of programming language design and implementation.

Jeeg

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.

χ-Spaces

It is of paramount importance that a security protocol effectively enforces the desired security requirements. The apparent simplicity of informal protocol descriptions hides the complex interactions of a security protocol which often invalidate informal correctness arguments and justify the effort of formal protocol verification. Verification, however, is usually carried out on an abstract model not at all related with a protocol's implementation. Experience shows that security breaches introduced in implementations of successfully verified models are rather common. The χ-Spaces framework is a tool meant to support every step of the protocol's life-cycle in a uniform manner. The core of the framework is a domain specific programming language based on SPL (Security Protocol Language) a formal model for studying security protocols. This simple, yet powerful, language allows the

implementation of security protocols in a concise an precise manner. Its underlying formal model, based on SPL, gives the power to prove interesting properties of the actual protocol implementation. Simulation, benchmarking and testing of the protocols is immediate using the χ-Sim tool.

Comments: Defence: June 16, 2003.

[DS-03-4] *PostScript*, *PDF*.

Federico Crazzolara.

Language, Semantics, and Methods for Security Protocols.

May 2003.

PhD thesis. xii+160.

Abstract: Security protocols help in establishing secure channels between communicating systems. Great care needs therefore to be taken in developing and implementing robust protocols. The complexity of security-protocol interactions can hide, however, security weaknesses that only a formal analysis can reveal. The last few years have seen the emergence of successful intensional, event-based, formal approaches to reasoning about security protocols. The methods are concerned with reasoning about the events that a security protocol can perform, and make use of a causal dependency that exists between events. Methods like strand spaces and the inductive method of Paulson have been designed to support an intensional, event-based, style of reasoning. These methods have successfully tackled a number of protocols though in an *ad hoc* fashion. They make an informal spring from a protocol to its representation and do not address how to build up protocol representations in a compositional fashion.

This dissertation presents a new, event-based approach to reasoning about security protocols. We seek a broader class of models to show how event-based models can be structured in a compositional way and so used to give a formal semantics to security protocols which supports proofs of their correctness. More precisely, we give a compositional event-based semantics to an economical, but expressive, language for describing security protocols (SPL); so the events and dependency of a wide range of protocols are determined once and for all. The net semantics allows the derivation of general properties and proof principles the use of which is demonstrated in establishing security properties for a number of protocols. The NSL public-key protocol, the ISO 5-pass authentication and the Π_3 key-translation protocols are analysed for their level of secrecy and authentication and for their robustness in runs where some session-keys are compromised. Particularly useful in the analysis are general results that show which events of a protocol run can not be those of a spy.

The net semantics of SPL is formally related to a transition semantics which can easily be implemented. (An existing SPL implementation bridges the gap that often exists between abstract protocol models on which verification of security properties is carried out and the implemented protocol.) SPL-nets are a par-

ticular kind of contextual Petri-nets. They have persistent conditions and as we show in this thesis, unfold under reasonable assumptions to a more basic kind of nets. We relate SPL-nets to strand spaces and inductive rules, as well as trace languages and event structures so unifying a range of approaches, as well as providing conditions under which particular, more limited, models are adequate for the analysis of protocols. The relations that are described in this thesis help integrate process calculi and algebraic reasoning with event-based methods to obtain a more accurate analysis of security properties of protocols.

Comments: Defence: May 15, 2003.

[DS-03-3] *PostScript*, *PDF*.

Jiří Srba.

Decidability and Complexity Issues for Infinite-State Processes.

2003.

PhD thesis. xii+171 pp.

Abstract: This thesis studies decidability and complexity borderlines for algorithmic verification of infinite-state systems. Verification techniques for finite-state processes are a successful approach for the validation of reactive programs operating over finite domains. When infinite data domains are introduced, most of the verification problems become in general undecidable. By imposing certain restrictions on the considered models (while still including infinite-state spaces), it is possible to provide algorithmic decision procedures for a variety of properties. Hence the models of infinite-state systems can be classified according to which verification problems are decidable, and in the positive case according to complexity considerations.

This thesis aims to contribute to the problems mentioned above by studying decidability and complexity questions of equivalence checking problems, i.e., the problems whether two infinite-state processes are equivalent with regard to some suitable notion of behavioural equivalence. In particular, our interest is focused on strong and weak bisimilarity checking within classical models of infinite-state processes.

Throughout the thesis, processes are modelled by the formalism of process rewrite systems which provides a uniform classification of all the considered classes of infinite-state processes.

We begin our exposition of infinite-state systems by having a closer look at the very basic model of labelled transition systems. We demonstrate a simple geometrical encoding of the labelled systems into unlabelled ones by transforming labels into linear paths of different lengths. The encoding preserves the answers to the strong bisimilarity checking and is shown to be effective e.g. for the classes of pushdown processes and Petri nets. This gives several decidability and complexity corollaries about unlabelled variants of the models.

We continue by developing a general decidability theorem for commutative process algebras like deadlock-sensitive BPP (basic

parallel processes), lossy BPP, interrupt BPP and timed-arc BPP nets. The introduced technique relies on the tableau-based proof of decidability of strong bisimilarity for BPP and we extend it in several ways to provide a wider applicability range of the technique. This, in particular, implies that all the mentioned classes of commutative process algebras allow for algorithmic checking of strong bisimilarity.

Another topic studied in the thesis deals with finding tight complexity estimates for strong and weak bisimilarity checking. A novel technique called existential quantification is explicitly described and used to show that the strong bisimilarity and regularity problems for basic process algebra and basic parallel processes are PSPACE-hard. In case of weak bisimilarity checking of basic parallel processes the problems are shown to be again PSPACE-hard — this time, however, also for the normed subclass.

Finally we consider the problems of weak bisimilarity checking for the classes of pushdown processes and PA-processes. Negative answers to the problems are given — both problems are proven to be undecidable. The proof techniques moreover solve some other open problems. For example the undecidability proof for pushdown processes implies also undecidability of strong bisimilarity for prefix-recognizable graphs.

The thesis is concluded with an overview of the state-of-the-art for strong and weak bisimilarity problems within the classes from the hierarchy of process rewrite systems.

Comments: Defence: April 14, 2003.

[DS-03-2] Frank D. Valencia.
Temporal Concurrent Constraint Programming.
February 2003.
PhD thesis. xvii+174.

Abstract: *Concurrent constraint programming* (ccp) is a formalism for concurrency in which agents interact with one another by telling (adding) and asking (reading) information in a shared medium. Temporal ccp extends ccp by allowing agents to be constrained by time conditions. This dissertation studies temporal ccp as a model of concurrency for discrete-timed systems. The study is conducted by developing a process calculus called `ntcc`.

The `ntcc` calculus generalizes the tcc model, the latter being a temporal ccp model for *deterministic* and *synchronous timed reactive* systems. The calculus is built upon few basic ideas but it captures several aspects of timed systems. As tcc, `ntcc` can model unit delays, time-outs, pre-emption and synchrony. Additionally, it can model *unbounded but finite delays, bounded eventuality, asynchrony* and *nondeterminism.* The applicability of the calculus is illustrated with several interesting examples of discrete-time systems involving mutable data structures, robotic devices, multi-agent systems and music applications.

The calculus is provided with a *denotational semantics* that captures the reactive computations of processes in the presence of arbitrary environments. The denotation is proven to be *fully-abstract* for a substantial fragment of the calculus. This dissertation identifies the exact technical problems (arising mainly from allowing nondeterminism, locality and time-outs in the calculus) that makes it impossible to obtain a fully abstract result for the full language of `ntcc`.

Also, the calculus is provided with a process logic for expressing *temporal specifications* of `ntcc` processes. This dissertation introduces a (*relatively*) *complete inference system* to prove that a given `ntcc`process satisfies a given formula in this logic. The denotation, process logic and inference system presented in this dissertation significantly extend and strengthen similar developments for tcc and (untimed) ccp.

This dissertation addresses the *decidability* of various behavioral equivalences for the calculus and *characterizes* their corresponding induced congruences. The equivalences (and their associated congruences) are proven to be decidable for a significant fragment of the calculus. The decidability results involve a systematic translation of processes into finite state Büchi automata. To the author's best knowledge the study of decidability for ccp equivalences is original to this work.

Furthermore, this dissertation deepens the understanding of previous ccp work by establishing an *expressive power hierarchy* of several temporal ccp languages which were proposed in the literature by other authors. These languages, represented in this dissertation as *variants* of `ntcc` , differ in their way of defining infinite behavior (i.e., *replication* or *recursion*) and the scope of variables (i.e., *static* or *dynamic scope*). In particular, it is shown that (1) recursive procedures with parameters can be encoded into parameterless recursive procedures with dynamic scoping, and vice-versa; (2) replication can be encoded into parameterless recursive procedures with static scoping, and vice-versa; (3) the languages from (1) are *strictly more expressive* than the languages from (2). (Thus, in this family of languages recursion is more expressive than replication and dynamic scope is more expressive than static scope.) Moreover, it is shown that behavioral equivalence is *undecidable* for the languages from (l), but *decidable* for the languages from (2). Interestingly, the undecidability result holds even if the process variables take values from a *fixed finite domain* whilst the decidability holds for *arbitrary domains*.

Both the expressive power hierarchy and decidability/undecidability results give a clear distinction among the various temporal ccp languages. Also, the distinction between static and dynamic scoping helps to clarify the situation in the (untimed) ccp family of languages. Moreover, the methods used in the decidability results may provide a framework to perform further systematic investigations of temporal ccp languages.

Comments: Defence: February 5, 2003.

[DS-03-1] *PostScript*, *PDF*.
Claus Brabrand.
Domain Specific Languages for Interactive Web Services.
January 2003.
PhD thesis. xiv+214 pp.

Abstract: This dissertation shows how domain specific languages may be applied to the domain of interactive Web services to obtain flexible, safe, and efficient solutions.

We show how each of four key aspects of interactive Web services involving sessions, dynamic creation of HTML/XML documents, form field input validation, and concurrency control, may benefit from the design of a dedicated language.

Also, we show how a notion of metamorphic syntax macros facilitates integration of these individual domain specific languages into a complete language.

The result is a domain specific language, !{<bigwig>}!,that supports virtually all aspects of the development of interactive Web services and provides flexible, safe, and efficient solutions.

[DS-02-5] *PostScript*, *PDF*.
Rasmus Pagh.
Hashing, Randomness and Dictionaries.
October 2002.
PhD thesis. x+167 pp.

Abstract: This thesis is centered around one of the most basic information retrieval problems, namely that of storing and accessing the elements of a set. Each element in the set has some associated information that is returned along with it. The problem is referred to as the *dictionary* problem, due to the similarity to a bookshelf dictionary, which contains a set of words and has an explanation associated with each word. In the *static* version of the problem the set is fixed, whereas in the *dynamic* version, insertions and deletions of elements are possible.

The approach taken is that of the theoretical algorithms community. We work (almost) exclusively with a *model*, a mathematical object that is meant to capture essential aspects of a real computer. The main model considered here (and in most of the literature on dictionaries) is a unit cost RAM with a word size that allows a set element to be stored in one word.

We consider several variants of the dictionary problem, as well as some related problems. The problems are studied mainly from an upper bound perspective, i.e., we try to come up with algorithms that are as efficient as possible with respect to various computing resources, mainly computation time and memory space. To some extent we also consider lower bounds, i.e., we attempt to show limitations on how efficient algorithms are possible. A central theme in the thesis is *randomness*. Randomized algorithms play an important role, in particular through the key technique of *hashing*. Additionally, probabilistic methods are used in several proofs.

Comments: Defence: October 11, 2002.

[DS-02-4] *PostScript*, *PDF*.
Anders Møller.
Program Verification with Monadic Second-Order Logic & Languages for Web Service Development.
September 2002.
PhD thesis. xvi+337 pp.

Abstract: Domain-specific formal languages are an essential part of computer science, combining theory and practice. Such languages are characterized by being tailor-made for specific application domains and thereby providing expressiveness on high abstraction levels and allowing specialized analysis and verification techniques. This dissertation describes two projects, each exploring one particular instance of such languages: monadic second-order logic and its application to program verification, and programming languages for construction of interactive Web services. Both program verification and Web service development are areas of programming language research that have received increased attention during the last years.

We first show how the logic Weak monadic Second-order Logic on Strings and Trees can be implemented efficiently despite an intractable theoretical worst-case complexity. Among several other applications, this implementation forms the basis of a verification technique for imperative programs that perform data-type operations using pointers. To achieve this, the basic logic is extended with layers of language abstractions. Also, a language for expressing data structures and operations along with correctness specifications is designed. Using Hoare logic, programs are split into loop-free fragments which can be encoded in the logic. The technique is described for recursive data types and later extended to the whole class of graph types. As an example application, we verify correctness properties of an implementation of the insert procedure for red-black search trees.

We then show how Web service development can benefit from high-level language support. Existing programming languages for Web services are typically general-purpose languages that provide only low-level primitives for common problems, such as maintaining session state and dynamically producing HTML or XML documents. By introducing explicit language-based mechanisms for those issues, we liberate the Web service programmer from the tedious and error-prone alternatives. Specialized program analyses aid the programmer by verifying at compile time that only valid HTML documents are ever shown to the clients at runtime and that the documents are constructed consistently. In addition, the language design provides support for declarative form-field validation, caching of dynamic documents, concurrency control based on temporal-logic specifications, and syntax-level macros for making additional language extensions. In its newest version, the programming language is designed as an extension of Java. To describe classes of XML documents, we introduce a novel XML schema language aiming to both simplify and generalize existing proposals. All parts are implemented and tested in practice.

Both projects involve design of high-level languages and specialized analysis and verification techniques, supporting the thesis that the domain-specific paradigm can provide a versatile and productive approach to development of formal languages.

Comments: Defence: September 9, 2002.

[DS-02-3] *PostScript*, *PDF*.
Riko Jacob.
Dynamic Planar Convex hull.
May 2002.
PhD thesis. xiv+112 pp.

Abstract: We determine the computational complexity of the dynamic convex hull problem in the planar case. We present a data structure that maintains a finite set of points in the plane under insertion and deletion of points in amortized $O(\log n)$ time. Here n denotes the number of points in the set. The data structure supports the reporting of the extreme point of the set in some direction in worst-case and amortized $O(\log n)$ time. The space usage of the data structure is $O(n)$. We give a lower bound on the amortized asymptotic time complexity that matches the performance of this data structure.

Comments: Defence: May 31, 2002.

[DS-02-2] *PostScript*, *PDF*.
Stefan Dantchev.
On Resolution Complexity of Matching Principles.
May 2002.
PhD thesis. xii+68 pp.

Abstract: Studying the complexity of mathematical proofs is important not only for automated theorem proving, but also for Mathematics as a whole. Each significant result in this direction would potentially have a great impact on Foundations of mathematics.

Surprisingly enough, the general Proof Complexity is closely related to Propositional Proof Complexity. The latter area was founded by Cook and Reckhow in 1979, and enjoyed quite a fast development since then. One of the main research directions is finding the precise complexity of some natural combinatorial principle within a relatively weak propositional proof system. The results in the thesis fall in this category. We study the Resolution complexity of some Matching Principles. The three major contributions of the thesis are as follows.

Firstly, we develop a general technique of proving resolution lower bounds for the perfect matchingprinciples based on regular planar graphs. No lower bounds for these were known prior to our work. As a matter of fact, one such problem, the Mutilated Chessboard, was suggested as hard to automated theorem provers in 1964, and remained open since then. Our technique proves a tight resolution lower bound for the Mutilated Chessboard as well as for Tseitin tautologies based on rectangular grid graph. We reduce these problems to Tiling games, a concept introduced by us, which may be of interest on its own.

Secondly, we find the exact Tree-Resolution complexity of the Weak Pigeon-Hole Principle. It is the most studied combinatorial principle, but even its Tree-Resolution complexity was unknown prior to our work. We develop a new, more general method for proving Tree-Resolution lower bounds. We also define and prove non-trivial upper bounds on worst-case proofs of the Weak Pigeon-Hole Principle. The worst-case proofs are first introduced by us, as a concept opposite to the optimal proofs.

Thirdly, we prove Resolution width-size trade-offs for the Pigeon-Hole Principle. Proving the size lower bounds via the width lower bounds was known since the seminal paper of Haken, who first proved an exponential lower bound for the ordinary Pigeon-Hole Principle. The width-size trade-offs however were not studied at all prior to our work. Our result gives an optimal width-size trade-off for resolution in general.

Comments: Defence: May 10, 2002.

[DS-02-1] *PDF*.
M. Oliver Möller.
Structure and Hierarchy in Real-Time Systems.
April 2002.
PhD thesis. xvi+228 pp.

Abstract: The development of digital systems is particularly challenging, if their correctness depends on the right timing of operations. One approach to enhance the reliability of such systems is model-based development. This allows for a formal analysis throughout all stages of design.

Model-based development is hindered mainly by the lack of adequate modeling languages and the high computational cost of the analysis. In this thesis we improve the situation along both axes.

First, we bring the mathematical model closer to the human designer. This we achieve by casting hierarchical structures—as known from statechart-like formalisms—into a formal timed model. This shapes a high-level language, which allows for fully automated verification.

Second, we use sound approximations to achieve more efficient automation in the verification of timed properties. We present two novel state-based techniques that have the potential to reduce time and memory consumption drastically.

The first technique makes use of structural information, in particular loops, to exploit regularities in the reachable state space. Via shortcut-like additions to the model we avoid repetition of similar states during an exhaustive state space exploration.

The second technique applies the abstract interpretation framework to a real-time setting. We preserve the control structure and approximate only the more expensive time component of a state. The verification of infinite behavior, also known as liveness properties, requires an assumption on the progress of time. We incorporate this assumption by means of limiting the behavior of the model with respect to delay steps. For a next-free temporal logic, this modification does not change the set of valid properties.

We supplement our research with experimental run-time data. This data is gathered via prototype implementations on top of the model checking tools UPPAAL and MOCHA.

[DS-01-10] *PostScript*, *PDF*.
Mikkel T. Jensen.
Robust and Flexible Scheduling with Evolutionary Computation.
November 2001.
PhD thesis. xii+299 pp.

Abstract: Over the last ten years, there have been numerous applications of evolutionary algorithms to a variety of scheduling problems. Like most other research on heuristic scheduling, the primary aim of the research has been on deterministic formulations of the problems. This is in contrast to real world scheduling problems which are usually not deterministic. Usually at the time the schedule is made some information about the problem and processing environment is available, but this information is uncertain and likely to change during schedule execution. Changes frequently encountered in scheduling environments include machine breakdowns, uncertain processing times, workers getting sick, materials being delayed and the appearance of new jobs. These possible environmental changes mean that a schedule which was optimal for the information available at the time of scheduling can end up being highly suboptimal when it is implemented and subjected to the uncertainty of the real world. For this reason it is very important to find methods capable of creating *robust* schedules (schedules expected to perform well after a minimal amount of modification when the environment changes) or *flexible* schedules (schedules expected to perform well after some degree of modification when the environment changes).

This thesis presents two fundamentally different approaches for scheduling job shops facing machine breakdowns. The first method is called *neighbourhood based robustness* and is based on an idea of minimising the cost of a neighbourhood of schedules. The scheduling algorithm attempts to find a small set of schedules with an acceptable level of performance. The approach is demonstrated to significantly improve the robustness and flexi-

bility of the schedules while at the same time producing schedules with a low implementation cost if no breakdown occurs. The method is compared to a state of the art method for stochastic scheduling and concluded to have the same level of performance, but a wider area of applicability. The current implementation of the method is based on an evolutionary algorithm, but since the real contribution of the method is a new performance measure, other implementations could be based on tabu search, simulated annealing or other powerful "blind" optimisation heuristics.

The other method for stochastic scheduling uses the idea of coevolution to create schedules with a guaranteed worst case performance for a known set of scenarios. The method is demonstrated to improve worst case performance of the schedules when compared to ordinary scheduling; it substantially reduces program running times when compared to a more standard approach explicitly considering all scenarios. Schedules based on worst case performance measures often have suboptimal performance if no disruption happens, so the coevolutionary algorithm is combined with a multi-objective algorithm which optimises worst case performance as well as performance without disruptions.

The coevolutionary worst case algorithm is also combined with another algorithm to create schedules with a guaranteed level of *worst deviation performance*. In worst deviation performance the objective is to minimise the highest possible performance difference from the schedule optimal for the scenario that actually takes place. Minimising this kind of performance measure involves solving a large number of related scheduling problems in one run, so a new evolutionary algorithm for this kind of problem is suggested.

Other contributions of the thesis include a new coevolutionary algorithm for minimax problems. The new algorithm is capable of solving problems with an asymmetric property that causes previously published algorithms to fail. Also, a new algorithm to solve the economic lot and delivery scheduling problem is presented. The new algorithm is guaranteed to solve the problem to optimality in polynomial time, something previously published algorithms have not been able to do.

[DS-01-9] *PostScript*, *PDF*.
Flemming Friche Rodler.
Compression with Fast Random Access.
November 2001.
PhD thesis. xiv+124 pp.

Abstract: The main topic of this dissertation is the development and use of methods for space efficient storage of data combined with fast retrieval. By fast retrieval we mean that a single data element can be randomly selected and decoded efficiently. In particular, the focus will be on compression of volumetric data with fast random access to individual voxels, decoded from the compressed data.

Volumetric data is finding widespread use in areas such as medical imaging, scientific visualization, simulations and fluid dynamics. Because of the size of modern volumes, using such data sets in uncompressed form is often only possible on computers with extensive amounts of memory. By designing compression methods for volumetric data that support fast random access this problem might be overcome. Since lossless compression of three-dimensional data only provides relatively small compression ratios, lossy techniques must be used. When designing compression methods with fast random access there is a choice between designing general schemes that will work for a wide range of applications or to tailor the compression algorithm to a specific application at hand. This dissertation we be concerned with designing general methods and we present two methods for volumetric compression with fast random access. The methods are compared to other existing schemes showing them to be quite competitive.

The first compression method that we present is suited for online compression. That is, the data can be compressed as it is downloaded utilizing only a small buffer. Inspired by video coding the volume is considered as a stack of slices. To remove redundancies between slices a simple "motion estimation" technique is used. Redundancies are further reduced by wavelet transforming each slice using a two-dimensional wavelet transform. Finally, the wavelet data is thresholded and the resulting sparse representation is quantized and encoded using a nested block indexing scheme, which allows for efficient retrieval of coefficients. While being only slightly slower than other existing schemes the method improves the compression ratio by about 50%.

As a tool for constructing fast and efficient compression methods that support fast random access we introduce the concept of lossy dictionaries and show how to use it to achieve significant improvements in compressing volumetric data. The dictionary data structure is widely used in computer science as a tool for space efficient storage of sparse sets. The main performance parameters of dictionaries are space, lookup time and update time. In this dissertation we present a new and efficient dictionary scheme, based on hashing, called CUCKOO HASHING. CUCKOO HASHING achieves worst case constant lookup time, expected amortized constant update time and space usage of three words per element stored in the dictionary, i.e., the space usage is similar to that of binary search trees. Running extensive experiments we show CUCKOO HASHING to be very competitive to the most commonly used dictionary methods. Since these methods have nontrivial worst case lookup time CUCKOO HASHING is useful in time critical systems where such a guarantee is mandatory.

Even though, time efficient dictionaries with a reasonable space usage exist, the space usage of these are still too large to be of use in lossy compression. However, if the dictionary is allowed to store a slightly different set than intended, new and interesting possibilities originate. Very space efficient and fast data structures can be constructed by allowing the dictionary to discard some elements of the set (false negatives) and also allowing it to include elements not in the set (false positives). The lossy dictionary we present in this dissertation is a variant of CUCKOO HASHING which results in fast lookup time. We show that our data structure is nearly optimal with respect to space usage. Experimentally, we find that the data structure has very good behavior with respect to keeping the most important elements of the set which is partially explained by theoretical considerations. Besides working well in compression our lossy dictionary data structure might find use in applications such as web cache sharing and differential files.

. In the second volumetric compression method with fast random access that we present in this dissertation, we look at a completely different and rather unexploited way of encoding wavelet coefficients. In wavelet based compression it is common to store the coefficients of largest magnitude, letting all other coefficients be zero. However, the new method presented allows a slightly different set of coefficients to be stored. The foundation of the method is a three-dimensional wavelet transform of the volume and the lossy dictionary data structure that we introduce. Comparison to other previously suggested schemes in the literature, including the two-dimensional scheme mentioned above, shows an improvement of up to 80% in compression ratio while the time for accessing a random voxel is considerably reduced compared to the first method.

[DS-01-8] Niels Damgaard.
Using Theory to Make Better Tools.
October 2001.
PhD thesis.

Abstract: In this dissertation four tools based on theory will be presented. The tools fall in four quite different areas.

The first tool is an extension of parser generators, enabling them to handle side constraints along with grammars. The side constraints are specified in a first-order logic on parse trees. The constraints are compiled into equivalent tree automata, which result in good runtime performance of the generated checkers. The tool is based on language theory, automata theory and logic.

The second tool is a static checker of assembly code for interrupt-based microcontrollers. The tool gives lower and upper bounds on the stack height, checks an implicit type-system and calculates the worst-case delay from an interrupt fires until it is handled. The tool is based on the theory of data-flow analysis.

The third 'tool' is a security extension of a system (`<bigwig>`) for making interactive web-services. The extension provides confidentiality, integrity and authenticity in the communication between the client and the server. Furthermore, some primitives are

introduced to ease the programming of cryptological protocols, like digital voting and e-cash, in the `<bigwig>` language. The extension uses the theory of cryptography.

The fourth tool is a black-box system for finding neural networks good at solving a given classification problem. Back-propagation is used to train the neural network, but the rest of the parameters are found by a distributed genetic-algorithm. The tool uses the theory of neural networks and genetic algorithms.

[DS-01-7] Lasse R. Nielsen.
A Study of Defunctionalization and Continuation-Passing Style.
August 2001.
PhD thesis. iv+280 pp.

Abstract: We show that defunctionalization and continuations unify a variety of independently developed concepts in programming languages.

Defunctionalization was introduced by Reynolds to solve a specific need—representing higher-order functions in a first-order definitional interpreter—but it is a general method for transforming programs from higher-order programs to first-order. We formalize this method and give it a partial proof of correctness. We use defunctionalization to connect independent first-order and higher-order specifications and proofs by, e.g., defunctionalizing continuations.

The canonical specification of a continuation-passing style (CPS) transformation introduces many administrative redexes. To show a simulation theorem, Plotkin used a so-called colon-translation to bypass administrative reductions and relate the reductions common to the transformed and the un-transformed program. This allowed him to make a proof by structural induction on the source program. We extend the colon-translation and Plotkin's proof to show simulation theorems for a higher-order CPS transformation, written in a two-level language, and for a selective CPS transformation, keeping parts of the source program in direct style.

Using, e.g., Plotkin's CPS transformation and then performing all possible administrative reductions gives a so-called two-pass CPS transformation. A one-pass CPS transformation gives the same result but without traversing the output. Several different one-pass CPS transformations exist, and we describe, evaluate, and compare three of these: (1) A higher-order CPS transformation à la Danvy and Filinski, (2) a syntactic-theory based CPS transformation derived from Sabry and Felleisen's CPS transformation, and (3) a look-ahead based CPS transformation which extends the colon-translation to a full one-pass CPS transformation and is new. This look-ahead based CPS transformation is not compositional, but it allows reasoning by structural induction, which we use to prove the correctness of a direct-style transformation.

Syntactic theories are a framework for specifying operational semantics that uses evaluation contexts to define a reduction relation. The two primary operations needed to implement syntactic theories are decomposing an expression into an evaluation context and a redex, and plugging an expression into an evaluation context to generate an expression. If implemented separately, these operations each take time proportional to the size of the evaluation context, which can cause a quadratic overhead on the processing of programs. However, in actual use of syntactic theories, the two operations mostly occur consecutively. We define a single function, 'refocus', that combines these operations and that avoids unnecessary overhead, thereby deforesting the composition of plugging and decomposition. We prove the correctness of refocusing and we show how to automatically generate refocus functions from a syntactic theory with unique decomposition.

We connect defunctionalization and Church encoding. Church encoding represents data-structures as functions and defunctionalization represents functions as data-structures, and we show to which extent they are inverses of each other.

We use defunctionalization to connect a higher-order CPS transformation and a syntactic-theory based CPS transformation. Defunctionalizing the static continuations of the higher-order transformation gives exactly the evaluation contexts of the syntactic theory.

We combine CPS transformation and defunctionalization to automatically generate data structures implementing evaluation contexts.Defunctionalizing the continuations of a CPS-transformed reduction function yields the evaluation contexts and their plug function—corresponding to the traditional view of continuations as a representation of the context. Defunctionalizing the continuations of a CPS-transformed evaluation function yields the evaluation contexts and the refocus functions—corresponding to the traditional view of continuations as a representation of the remainder of the computation.

[DS-01-6] *PostScript*, *PDF*.
Bernd Grobauer.
Topics in Semantics-based Program Manipulation.
August 2001.
PhD thesis. ii+x+186 pp.

Abstract: Programming is at least as much about manipulating existing code as it is about writing new code. Existing code is *modified*, for example to make inefficient code run faster, or to accommodate for new features when reusing code; existing code is *analyzed*, for example to verify certain program properties, or to use the analysis information for code modifications. Semantics-based program manipulation addresses methods for program modifications and program analyses that are formally defined and therefore can be verified with respect to the programming-language semantics. This dissertation comprises four articles in the field

of semantics-based techniques for program manipulation: three articles are about *partial evaluation*, a method for program specialization; the fourth article treats an approach to *automatic cost analysis*.

Partial evaluation optimizes programs by specializing them with respect to parts of their input that are already known: Computations that depend only on known input are carried out during partial evaluation, whereas computations that depend on unknown input give rise to residual code. For example, partially evaluating an interpreter with respect to a program written in the interpreted language yields code that carries out the computations described by that program; partial evaluation is used to remove interpretive overhead. In effect, the partial evaluator serves as a compiler from the interpreted language into the implementation language of the interpreter. Compilation by partial evaluation is known as the first *Futamura projection*. The second and third Futamura projection describe the use of partial evaluation for compiler generation and compiler-generator generation, respectively; both require the partial evaluator that is employed to be *self applicable*.

The first article in this dissertation describes how the second Futamura projection can be achieved for type-directed partial evaluation (TDPE), a relatively recent approach to partial evaluation: We derive an ML implementation of the second Futamura projection for TDPE. Due to the differences between 'traditional', syntax-directed partial evaluation and TDPE, this derivation involves several conceptual and technical steps. These include a suitable formulation of the second Futamura projection and techniques for making TDPE amenable to self-application.

In the second article, compilation by partial evaluation plays a central role for giving a unified approach to goal-directed evaluation, a programming-language paradigm that is built on the notions of backtracking and of generating successive results. Formulating the semantics of a small goal-directed language as a *monadic* semantics—a generic approach to structuring denotational semantics—allows us to relate various possible semantics to each other both conceptually and formally. We thus are able to explain goal-directed evaluation using an intuitive list-based semantics, while using a continuation semantics for semantics-based compilation through partial evaluation. The resulting code is comparable to that produced by an optimized compiler described in the literature.

The third article revisits one of the success stories of partial evaluation, the generation of efficient string matchers from intuitive but inefficient implementations. The basic idea is that specializing a naive string matcher with respect to a pattern string should result in a matcher that searches a text for this pattern with running time independent of the pattern and linear in the length of the text. In order to succeed with basic partial-evaluation techniques, the naive matcher has to be modified in a non-trivial way, carrying out so-called *binding-time improvements*. We present a step-by-step derivation of a binding-time improved matcher consisting of one problem-dependent step followed by standard binding-time improvements. We also consider several variants of matchers that specialize well, amongst them the first such matcher presented in the literature, and we demonstrate how variants can be derived from each other systematically.

The fourth article is concerned with program analysis rather than program transformation. A challenging goal for program analysis is to extract information about time or space complexity from a program. In complexity analysis, one often establishes *cost recurrences* as an intermediate step, and this step requires an abstraction from data to data size. We use information contained in dependent types to achieve such an abstraction: Dependent ML (DML), a conservative extension of ML, provides dependent types that can be used to associate data with size information, thus describing a possible abstraction. We automatically extract cost recurrences from first-order DML programs, guiding the abstraction from data to data size with information contained in DML type derivations.

[DS-01-5] *PostScript*, *PDF*.
Daniel Damian.
On Static and Dynamic Control-Flow Information in Program Analysis and Transformation.
August 2001.
PhD thesis. xii+111 pp.

Abstract: This thesis addresses several aspects of static and dynamic control-flow information in programming languages, by investigating its interaction with program transformation and program sanalysis.

Control-flow information indicates for each point in a program the possible program points to be executed next. Control-flow information in a program may be static, as when the syntax of the program directly determines which parts of the program may be executed next. Control-flow information may be dynamic, as when run-time values and inputs of the program are required to determine which parts of the program may be executed next. A control-flow analysis approximates the dynamic control-flow information with conservative static control-flow information.

We explore the impact of a continuation-passing-style (CPS) transformation on the result of a constraint-based control-flow analysis over Moggi's computational metalanguage. A CPS transformation makes control-flow explicitly available to the program by abstracting the remainder of the computation into a continuation. Moggi's computational metalanguage supports reasoning about higher-order programs in the presence of computational effects. We show that a non-duplicating CPS transformation does not alter the result of a monovariant constraint-based control-flow analysis.

Building on control-flow analysis, we show that traditional constraint-based binding-time analysis and traditional partial evaluation benefit from the effects of a CPS transformation, while the same CPS transformation does not affect continuation-based partial evaluation and its corresponding binding-time analysis. As an intermediate result, we show that reducing a program in the computational metalanguage to monadic normal form also improves binding times for traditional partial evaluation while it does not affect continuation-based partial evaluation.

In addition, we show that linear β-reductions have no effect on control-flow analysis. As a corollary, we solve a problem left open in Palsberg and Wand's CPS transformation of flow information. Furthermore, using Danvy and Nielsen's first-order, one-pass CPS transformation, we present a simpler CPS transformation of flow information with a simpler correctness proof.

We continue by exploring Shivers's time-stamps-based technique for approximating program analyses over programs with dynamic control flow. We formalize a time-stamps-based algorithm for approximating the least fixed point of a generic program analysis over higher-order programs, and we prove its correctness.

We conclude by investigating the translation of first-order structured programs into first-order unstructured programs. We present a one-pass translation that integrates static control-flow information and that produces programs containing no chains of jumps, no unused labels, and no redundant labels.

[DS-01-4] *PostScript*, *PDF*.
Morten Rhiger.
Higher-Order Program Generation.
August 2001.
PhD thesis. xiv+144 pp.

Abstract: This dissertation addresses the challenges of embedding programming languages, specializing generic programs to specific parameters, and generating specialized instances of programs directly as executable code. Our main tools are higher-order programming techniques and automatic program generation. It is our thesis that they synergize well in the development of customizable software.

Recent research on domain-specific languages propose to embed them into existing general-purpose languages. Typed higher-order languages have proven especially useful as meta languages because they provide a rich infrastructure of higher-order functions, types, and modules. Furthermore, it has been observed that embedded programs can be restricted to those having simple types using a technique called "phantom types". We prove, using an idealized higher-order language, that such an embedding is sound (i.e., when all object-language terms that can be embedded into the meta language are simply typed) and that it is complete (i.e., when all simply typed object-language terms can be embedded into the meta language). The soundness proof is shown by induction over meta-language terms using a Kripke logical relation. The completeness proof is shown by induction over object-language terms. Furthermore, we address the use of Haskell and Standard ML as meta-languages.

Normalization functions, as embodied in type-directed partial evaluation, map a simply-typed higher-order value into a representation of its long beta-eta normal form. However, being dynamically typed, the original Scheme implementation of type-directed partial evaluation does not restrict input values to be typed at all. Furthermore, neither the original Scheme implementation nor the original Haskell implementation guarantee that type-directed partial evaluation preserves types. We present three implementations of type-directed partial evaluation in Haskell culminating with a version that restricts the input to typed values and for which the proofs of type-preservation and normalization are automated.

Partial evaluation provides a solution to the disproportion between general programs that can be executed in several contexts and their specialized counterparts that can be executed efficiently. However, stand-alone partial evaluation is usually too costly when a program must be specialized at run time. We introduce a collection of byte-code combinators for OCaml, a dialect of ML, that provides run-time code generation for OCaml programs. We apply these byte-code combinators in semantics-directed compilation for an imperative language and in run-time specialization using type-directed partial evaluation.

Finally, we present an approach to compiling goal-directed programs, i.e., programs that backtrack and generate successive results: We first specify the semantics of a goal-directed language using a monadic semantics and a spectrum of monads. We then compile goal-directed programs by specializing their interpreter (i.e., by using the first Futamura projection), using type-directed partial evaluation. Through various back ends, including a run-time code generator, we generate ML code, C code, and OCaml byte code.

[DS-01-3] *PostScript*, *PDF*.
Thomas S. Hune.
Analyzing Real-Time Systems: Theory and Tools.
March 2001.
PhD thesis. xii+265 pp.

Abstract: The main topic of this dissertation is the development and use of methods for formal reasoning about the correctness of real-time systems, in particular methods and tools to handle new classes of problems. In real-time systems the correctness of the system does not only depend on the order in which actions take place, but also the timing of the actions. The formal reasoning presented here is based on (extensions of) the model of timed automata and tools supporting this model, mainly UPPAAL. Real-time systems are often part of safety critical systems e.g. control systems for planes, trains, or factories, though also everyday electronics as audio/video equipment and (mobile) phones are considered real-time systems. Often these systems are concurrent systems with a number of components interacting, and reasoning about such systems is notoriously difficult. However, since most of the systems are either safety critical or errors in the systems are costly, ensuring correctness is very important, and hence formal reasoning can play a role in increasing the reliability of real-time systems.

[DS-01-2] Jakob Pagter.
Time-Space Trade-Offs.
March 2001.
PhD thesis. xii+83 pp.

Abstract: In this dissertation we investigate the complexity of the time-space trade-off for sorting, element distinctness, and similar problems. We contribute new results, techniques, tools, and definitions pertaining to time-space trade-offs for the RAM (Random Access Machine), including new tight upper bounds on the time-space trade-off for Sorting in a variety of RAM models and a strong time-space trade-off lower bound for a decision problem in the RAM (in fact the branching program) model. We thus contribute to the general understanding of a number of fundamental combinatorial problems.

Time and space are the two most fundamental resources studied in the areas of algorithms and computational complexity theory and it is clearly important to study how they are related. In 1966 Cobham founded the area of time-space trade-offs by formally proving the first time-space trade-off—for the language of palindromes, i.e., a formulae that relate time and space in the form of either an upper or a lower bound—in this case $T \cdot S \in \Theta(n^2)$. Over the last thirty five years the area of time-space trade-offs has developed significantly and now constitutes its own branch of algorithms and computational complexity.

The area of time-space trade-offs deals with both upper and lower bounds and both are interesting, theoretically as well as practically. The viewpoint of this dissertation is theoretical, but we believe that some of our results can find applications in practice as well.

The last four years has witnessed a number of significant breakthroughs on proving lower bounds for decision problems, including the first non-trivial lower bound on the unit-cost RAM. We generalize work of Ajtai and prove the quantitatively best known lower bound in this model, namely a $\Omega(n \lg n / \lg\lg n)$ bound on time for any RAM deciding the so-called Hamming distance problem in space $n^{1-\epsilon}$.

For non-decision problems, i.e., problems with multiple outputs,

much stronger results are known. For Sorting, a fundamental problem in computer science, Beame proved that for any RAM the time-space product satisfies $T \cdot S \in \Omega(n^2)$. We prove that this bound is in fact tight (for most time functions), i.e., we show a $T \cdot S \in O(n^2)$ upper bound for time between $O(n(\lg\lg n)^2)$ and (roughly) $O(n^2)$. If we allow randomization in the Las Vegas fashion the lower bound still holds, and in this case we can meet it down to $O(n \lg\lg n)$.

From a practical perspective hierarchical memory layout models are the most interesting. Such models are called external memory models, in contrast to the internal memory models discussed above. Despite the fact that space might be of great relevance when solving practical problems on real computers, no theoretical model capturing space (and time simultaneously) has been defined. We introduce such a model and use it to prove so-called IO-space trade-offs for Sorting. Building on the abovementioned techniques for time-space efficient internal memory Sorting, we develop the first IO-space efficient external memory Sorting algorithms. Further, we prove that these algorithms are optimal using a transformation result which allows us to deduce external memory lower bounds (on IO-space trade-offs) from already known internal memory lower bounds (such as that of Beame).

[DS-01-1] *PostScript*, *PDF*.
Stefan Dziembowski.
Multiparty Computations — Information-Theoretically Secure Against an Adaptive Adversary.
January 2001.
PhD thesis. 109 pp.

Abstract: In this thesis we study a problem of doing Verifiable Secret Sharing (VSS) and Multiparty Computations in a model where private channels between the players and a broadcast channel is available. The adversary is active, adaptive and has an unbounded computing power. The thesis is based on two papers [1,2].

In [1] we assume that the adversary can corrupt any set from a given *adversary* structure. In this setting we study a problem of doing efficient VSS and MPC given an access to a secret sharing scheme (SS). For all adversary structures where VSS is possible at all, we show that, up to a polynomial time black-box reduction, the complexity of adaptively secure VSS is the same as that of ordinary secret sharing (SS), where security is only required against a passive, static adversary. Previously, such a connection was only known for linear secret sharing and VSS schemes.

We then show an impossibility result indicating that a similar equivalence does not hold for Multiparty Computation (MPC): we show that even if protocols are given black-box access for free to an idealized secret sharing scheme secure for the access structure in question, it is not possible to handle all relevant access structures efficiently, not even if the adversary is passive and static. In other words, general MPC can only be black-box reduced efficiently to secret sharing if extra properties of the secret sharing scheme used (such as linearity) are assumed.

The protocols of [2] assume that we work against a *threshold* adversary structure. We propose new VSS and MPC protocols that are substantially more efficient than the ones previously known.

Another contribution of [2] is an attack against a Weak Secret Sharing Protocol (WSS) of [3]. The attack exploits the fact that the adversary is adaptive. We present this attack here and discuss other problems caused by the adaptiveness.

All protocols in the thesis are formally specified and the proofs of their security are given.

[1] Ronald Cramer, Ivan Damgård, Stefan Dziembowski, Martin Hirt, and Tal Rabin. Efficient multiparty computations with dishonest minority. In Advances in Cryptology — Eurocrypt '99, volume 1592 of Lecture Notes in Computer Science, pages 311–326, 1999.

[2] Ronald Cramer, Ivan Damgård, and Stefan Dziembowski. On the complexity of verifiable secret sharing and multiparty computation. In Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, pages 25–334, May 2000.

[3] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing, pages 73–85, Seattle, Washington, 15–17 May 1989.

.

[DS-00-7] *PostScript*, *PDF*.
Marcin Jurdziński.
Games for Verification: Algorithmic Issues.
December 2000.
PhD thesis. ii+112 pp.

Abstract: This dissertation deals with a number of algorithmic problems motivated by computer aided formal verification of finite state systems. The goal of formal verification is to enhance the design and development of complex systems by providing methods and tools for specifying and verifying correctness of designs. The success of formal methods in practice depends heavily on the degree of automation of development and verification process. This motivates development of efficient algorithms for problems underlying many verification tasks.

Two paradigmatic algorithmic problems motivated by formal verification that are in the focus of this thesis are model checking and bisimilarity checking. In the thesis game theoretic formulations of the problems are used to abstract away from syntactic and semantic peculiarities of formal models and specification formalisms studied. This facilitates a detailed algorithmic analysis,

leading to two novel model checking algorithms with better theoretical or practical performance, and to an undecidability result for a notion of bisimilarity.

The original technical contributions of this thesis are collected in three research articles whose revised and extended versions are included in the dissertation.

In the first two papers the computational complexity of deciding the winner in parity games is studied. The problem of solving parity games is polynomial time equivalent to the modal mu-calculus model checking. The modal mu-calculus plays a central role in the study of logics for specification and verification of programs. The model checking problem is extensively studied in literature on computer aided verification. The question whether there is a polynomial time algorithm for the modal mu-calculus model checking is one of the most challenging and fascinating open questions in the area.

In the first paper a new algorithm is developed for solving parity games, and hence for the modal mu-calculus model checking. The design and analysis of the algorithm are based on a semantic notion of a progress measure. The worst-case running time of the resulting algorithm matches the best worst-case running time bounds known so far for the problem, achieved by the algorithms due to Browne at al., and Seidl. Our algorithm has better space complexity: it works in small polynomial space; the other two algorithms have exponential worst-case space complexity.

In the second paper a novel approach to model checking is pursued, based on a link between parity games and discounted payoff and stochastic games, established and advocated by Puri. A discrete strategy improvement algorithm is given for solving parity games, thereby proving a new procedure for the modal mu-calculus model checking. Known strategy improvement algorithms, as proposed for stochastic games by Hoffman and Karp, and for discounted payoff games and parity games by Puri, work with real numbers and require solving linear programming instances involving high precision arithmetic. The present algorithm for parity games avoids these difficulties by efficient manipulation of carefully designed discrete valuations. A fast implementation is given for a strategy improvement step. Another advantage of the approach is that it provides a better conceptual understanding of the underlying discrete structure and gives hope for easier analysis of strategy improvement algorithms for parity games. However, so far it is not known whether the algorithm works in polynomial time. The long standing problem whether parity games can be solved in polynomial time remains open.

In the study of concurrent systems it is common to model concurrency by non-determinism. There are, however, some models of computation in which concurrency is represented explicitly; elementary net systems and asynchronous transition systems are well-known examples. History preserving and hereditary history preserving bisimilarities are behavioural equivalence notions taking into account causal relationships between events of concurrent systems. Checking history preserving bisimilarity is known to be decidable for finite labelled elementary nets systems and asynchronous transition systems. Its hereditary version appears to be only a slight strengthening and it was conjectured to be decidable too. In the third paper it is proved that checking hereditary history preserving bisimilarity is undecidable for finite labelled asynchronous transition systems and elementary net systems. This solves a problem open for several years. The proof is done in two steps. First an intermediate problem of deciding the winner in domino bisimulation games for origin constrained tiling systems is introduced and its undecidability is shown by a reduction from the halting problem for 2-counter machines. Then undecidability of hereditary history preserving bisimilarity is shown by a reduction from the problem of domino bisimulation games.

[DS-00-6] *PostScript*, *PDF*.
Jesper G. Henriksen.
Logics and Automata for Verification: Expressiveness and Decidability Issues.
May 2000.
PhD thesis. xiv+229 pp.

Abstract: This dissertation investigates and extends the mathematical foundations of logics and automata for the interleaving and synchronous noninterleaving view of system computations with an emphasis on decision procedures and relative expressive powers, and introduces extensions of these foundations to the emerging domain of noninterleaving asynchronous computations. System computations are described as occurrences of system actions, and tractable collections of such computations can be naturally represented by finite automata upon which one can do formal analysis. Specifications of system properties are usually described in formal logics, and the question whether the system at hand satisfies its specification is then solved by means of automata-theoretic constructions.

Our focus here is on the linear time behaviour of systems, where executions are modeled as sequence-like objects, neither reflecting nondeterminism nor branching choices. We consider a variety of linear time paradigms, such as the classical interleaving view, the synchronous noninterleaving view, and conclude by considering an emerging paradigm of asynchronous noninterleaving computation. Our contributions are mainly theoretical though there is one piece of practical implementation work involving a verification tool. The theoretical work is concerned with a range of logics and automata and the results involve the various associated decision procedures motivated by verification problems, as well as the relative expressive powers of many of the logics that we consider.

Our research contributions, as presented in this dissertation, are as follows. We describe the practical implementation of the verification tool Mona. This tool is basically driven by an engine which translates formulas of monadic second-order logic for finite strings to deterministic finite automata. This translation is known to have a daunting complexity-theoretic lower bound, but surprisingly enough, it turns out to be possible to implement a translation algorithm which often works efficiently in practice; one of the major reasons being that the internal representation of the constituent automata can be maintained symbolically in terms of binary decision diagrams. In effect, our implementation can be used to verify the so-called safety properties because collections of finite strings suffice to capture such properties.

For reactive systems, one must resort to infinite computations to capture the so-called liveness properties. In this setting, the predominant specification mechanism is Pnueli's LTL which turns out to be computationally tractable and, moreover, equal in expressive power to the first-order fragment of monadic second-order logic. We define an extension of LTL based on the regular programs of PDL to obtain a temporal logic, DLTL, which remains computationally feasible and is yet expressively equivalent to the full monadic second-order logic.

An important class of distributed systems consists of networks of sequential agents that synchronize by performing common actions together. We exhibit a distributed version of DLTL and show that it captures exactly all linear time properties of such systems, while the verification problem, once again, remains tractable.

These systems constitute a subclass of a more general class of systems with a static notion of independence. For such systems the set of computations constitute interleavings of occurrences of causally independent actions. These can be grouped in a natural manner into equivalence classes corresponding to the same partially-ordered behaviour. The equivalence classes of computations of such systems can be canonically represented by restricted labelled partial orders known as (Mazurkiewicz) traces. It has been noted that many properties expressed as LTL-formulas have the "all-or-none" flavour, i.e. either all computations of an equivalence class satisfy the formula or none do. For such properties (e.g. "reaching a deadlocked state") it is possible to take advantage of the noninterleaving nature of computations and apply the so-called partial-order methods for verification to substantially reduce the computational resources needed for the verification task. This leads to the study of linear time temporal logics interpreted directly over traces, as specifications in such logics are guaranteed to have the "all-or-none" property. We provide an elaborate survey of the various distributed linear time temporal logics interpreted over traces.

One such logic is TLC, through which one can directly formulate causality properties of concurrent systems. We strengthen TLC to obtain a natural extended logic TLC* and show that the extension adds nontrivially to the expressive power. In fact, very little is known about the relative expressive power of the various logics for traces. The game-theoretic proof technique that we introduce may lead to new separation results concerning such logics.

In application domains such as telecommunication software, the synchronous communication mechanism that traces are based on is not appropriate. Rather, one would like message-passing to be the basic underlying communication mechanism. Message Sequence Charts (MSCs) are ideally suited for the description of such scenarios, where system executions constitute partially ordered exchanges of messages. This raises the question of what constitutes reasonable collections of MSCs upon which one can hope to do formal analysis. We propose a notion of regularity for collections of MSCs and tie it up with a class of finite-state devices characterizing such regular collections. Furthermore, we identify a monadic second-order logic which also defines exactly these regular collections.

The standard method for the description of multiple scenarios in this setting has been to employ MSC-labelled finite graphs, which might or might not describe collections of MSCs regular in our sense. One common feature is that these collections are finitely generated, and we conclude by exhibiting a subclass of such graphs which describes precisely the collections of MSCs that are both finitely generated and regular.

[DS-00-5] *PostScript*, *PDF*.
Rune B. Lyngsø.
Computational Biology.
March 2000.
PhD thesis. xii+173 pp.

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

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

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

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

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

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

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

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

[DS-00-4] *PostScript*, *PDF*.
Christian N. S. Pedersen.
Algorithms in Computational Biology.
March 2000.
PhD thesis. xii+210 pp.

Abstract: In this thesis we are concerned with constructing algorithms that address problems with biological relevance. This activity is part of a broader interdisciplinary area called computational biology, or bioinformatics, that focus on utilizing the capacities of computers to gain knowledge from biological data. The majority of problems in computational biology relate to molecular or evolutionary biology, and focus on analyzing and comparing the genetic material of organisms. A deciding factor in shaping the area of computational biology is that biomolecules that DNA, RNA and proteins that are responsible for storing and utilizing the genetic material in an organism, can be described as strings over finite alphabets. The string representation of biomolecules allows for a wide range of algorithmic techniques concerned with strings to be applied for analyzing and comparing biological data. We contribute to the field of computational biology by constructing and analyzing algorithms that

address problems with relevance to biological sequence analysis and structure prediction.

The genetic material of organisms evolve by discrete mutations, most prominently substitutions, insertions and deletions of nucleotides. Since the genetic material is stored in DNA sequences and reflected in RNA and protein sequences, it makes sense to compare two or more biological sequences in order to look for similarities and differences that can be used to infer the relatedness of the sequences. In the thesis we consider the problem of comparing two sequences of coding DNA when the relationship between DNA and proteins is taken into account. We do this by using a model that penalizes an event on the DNA by the change it induces on the encoded protein. We analyze the model in details and construct an alignment algorithm that improves on the existing best alignment algorithm in the model by reducing its running time with a quadratic factor. This makes the running time of our alignment algorithm equal to the running time of alignment algorithms based on much simpler models.

If a family of related biological sequences are available it is natural to derive a compact characterization of the sequence family. Among other things, such a characterization can be used to search for unknown members of the sequence family. A widely used way to describe the characteristics of a sequence family is to construct a hidden Markov model that generates members of the sequence family with high probability and non-members with low probability. In this thesis we consider the general problem of comparing hidden Markov models. We define novel measures between hidden Markov models and show how to compute them efficiently using dynamic programming. Since hidden Markov models are widely used to characterize biological sequence families, our measures and methods for comparing hidden Markov models immediate apply to comparison of entire biological sequence families.

Besides comparing sequences and sequence families, we also consider problems of finding regularities in a single sequence. Looking for regularities in a single biological sequence can be used to reconstruct part of the evolutionary history of the sequence or to identify the sequence among other sequences. In this thesis we focus on general string problems motivated by biological applications because biological sequences are strings. We construct an algorithm that finds all maximal pairs of equal substrings in a string, where each pair of equal substrings adheres to restrictions on the number of characters between the occurrences of the two substrings in the string. This is a generalization of finding tandem repeats and the running time of the algorithm is comparable to the running time of existing algorithms for finding tandem repeats. The algorithm is based on a general technique that combines a traversal of a suffix tree with efficient merging of search trees. We use the same general technique to construct an algo-

rithm that finds all maximal quasiperiodic substrings in a string. A quasiperiodic substring is a substring that can be described as concatenations and superpositions of a shorter substring. Our algorithm for finding maximal quasiperiodic substrings has a running time that is a logarithmic factor better than the running time of the existing best algorithm for the problem.

Analyzing and comparing the string representations of biomolecules can reveal a lot of useful information about the biomolecules, but knowing the three-dimensional structures of the biomolecules often reveal additional information that is not immediately visible from their string representations. Unfortunately it is difficult and time consuming to determine the three-dimensional structure of a biomolecule experimentally, so computational methods for structure prediction are in demand. Constructing such methods is also difficult and often results in the formulation of intractable computational problems. In this thesis we construct an algorithm that improves on the widely used mfold algorithm for RNA secondary structure prediction by allowing a less restrictive model of structure formation without an increase in the running time. We also analyze the protein folding problem in the two-dimensional hydrophobic-hydrophilic lattice model. Our analysis shows that several complicated folding algorithms do not produce better foldings in the worst case, in terms of free energy, than an existing much simpler folding algorithm.

[DS-00-3] Theis Rauhe.
Complexity of Data Structures (Unrevised).
March 2000.
PhD thesis. xii+115 pp.

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

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

Next we introduce new models for dynamic computation based on the cell probe model. We give these models access to non-

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

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

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

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

[DS-00-2] *PostScript*, *PDF*.
Anders B. Sandholm.
Programming Languages: Design, Analysis, and Semantics.
February 2000.
PhD thesis. xiv+233 pp.

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

Domain-specific language design is an emerging trend in software engineering. The idea is to express the analysis of a particular problem domain through the design of a programming language tailored to solving problems within that domain. There are many established examples of domain-specific languages, such as LaTeX, flex, and yacc. The novelty is to use such techniques for solving traditional software engineering task, where one would previously construct some collection of libraries within a general-purpose language. In Part I, we start by presenting the design of a domain-specific language for programming Web services. The various domain-specific aspects of Web service programming, such as, Web server runtime system design, document handling, and concurrency control, are treated.

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

We then turn to the use of dynamic Web documents, which in our language relies heavily on the use of static program analysis. Static program analysis allows one to offer static compile-time techniques for predicting safe and computable approximations to the set of values or behavior arising dynamically at runtime when executing a program on a computer. Some of the tradi-

tional applications of program analysis are avoidance of redundant and superfluous computations. Another aspect is that of program validation, where one provides guarantees that the analyzed code can safely be executed. We present an example of such an analysis as explained in the following. Many interactive Web services use the CGI interface for communication with clients. They will dynamically create HTML documents that are presented to the client who then resumes the interaction by submitting data through incorporated form fields. This protocol is difficult to statically type-check if the dynamic document are created by arbitrary script code using "printf" statements. Previous proposals have suggested using static document templates which trades flexibility for safety. We propose a notion of typed, higher-order templates that simultaneously achieve flexibility and safety. Our type system is based on a flow analysis of which we prove soundness. We also present an efficient runtime implementation that respects the semantics of only well-typed programs.

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

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

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

[DS-00-1] *PostScript*, *PDF*.
Thomas Troels Hildebrandt.
Categorical Models for Concurrency: Independence, Fairness and Dataflow.
February 2000.
PhD thesis. x+141 pp.

Abstract: This thesis is concerned with formal semantics and models for *concurrent* computational systems, that is, systems consisting of a number of parallel computing sequential systems, interacting with each other and the environment. A formal semantics gives meaning to computational systems by describing their *behaviour* in a *mathematical model*. For concurrent systems the interesting aspect of their computation is often how they interact with the environment *during* a computation and not in which state they terminate, indeed they may not be intended to terminate at all. For this reason they are often referred to as *reactive systems*, to distinguish them from traditional *calculational* systems, as e.g. a program calculating your income tax, for which the interesting behaviour is the answer it gives when (or if) it terminates, in other words the (possibly partial) function it computes between input and output. *Church's thesis* tells us that regardless of whether we choose the *lambda calculus*, *Turing machines*, or almost any modern programming language such as *C* or *Java* to describe calculational systems, we are able to describe exactly the same class of functions. However, there is no agreement on observable behaviour for concurrent reactive systems, and consequently there is no correspondent to Church's thesis. A result of this fact is that an overwhelming number of different and often competing notions of observable behaviours, primitive operations, languages and mathematical models for describing their semantics, have been proposed in the litterature on concurrency.

The work presented in this thesis contributes to a categorical approach to semantics for concurrency which have been developed through the last 15 years, aiming at a more coherent theory. The initial stage of this approach is reported on in the chapter on models for concurrency by Winskel and Nielsen in the Handbook of Logic in Computer Science, and focuses on relating the already existing models and techniques for reasoning about them. This work was followed by a uniform characterisation of behavioural equivalences from the notion of *bisimulation from*

open maps proposed by Joyal, Winskel and Nielsen, which was applicable to any of the categorical models and shown to capture a large number of existing variations on bisimulation. At the same time, a class of abstract models for concurrency was proposed, the *presheaf models for concurrency*, which have been developed over the last 5 years, culminating in the recent thesis of Cattani.

This thesis treats three main topics in concurrency theory: *independence*, *fairness* and *non-deterministic dataflow*.

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

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

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

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

[DS-99-1] *PostScript*, *PDF*.
Gian Luca Cattani.
Presheaf Models for Concurrency (Unrevised).
April 1999.
PhD thesis. xiv+255 pp.

Abstract: In this dissertation we investigate presheaf models for concurrent computation. Our aim is to provide a systematic treatment of bisimulation for a wide range of concurrent process calculi. Bisimilarity is defined abstractly in terms of open maps as in the work of Joyal, Nielsen and Winskel. Their work inspired this thesis by suggesting that presheaf categories could provide abstract models for concurrency with a built-in notion of bisimulation.

We show how presheaf categories, in which traditional models of concurrency are embedded, can be used to deduce congruence properties of bisimulation for the traditional models. A key result is given here; it is shown that the homomorphisms between presheaf categories, i.e., colimit preserving functors, preserve open map bisimulation. We follow up by observing that presheaf categories and colimit preserving functors organise in what can be considered as a category of non-deterministic domains. Presheaf models can be obtained as solutions to recursive domain equations. We investigate properties of models given for a range of concurrent process calculi, including **CCS**, **CCS** with value-passing, π-calculus and a form of **CCS** with linear process passing. Open map bisimilarity is shown to be a congruence for each calculus. These are consequences of general mathematical results like the preservation of open map bisimulation by colimit preserving functors. In all but the case of the higher order calculus, open map bisimulation is proved to coincide with traditional notions of bisimulation for the process terms. In the case of higher order processes, we obtain a ner equivalence than the one one would normally expect, but this helps reveal interesting aspects of the relationship between the presheaf and the operational semantics. For a fragment of the language, corresponding to a form of λ-calculus, open map bisimulation coincides with applicative bisimulation.

In developing a suitable general theory of domains, we extend results and notions, such as the limit-colimit coincidence theorem of Smyth and Plotkin, from the order-enriched case to a "fully" 2-categorical situation. Moreover we provide a domain theoretical analysis of (open map) bisimulation in presheaf categories. We present, in fact, induction and coinduction principles for recursive domains as in the works of Pitts and of Hermida and Jacobs and use them to derive a coinduction property based on bisimulation.

[DS-98-3] *PostScript*, *PDF*.
Kim Sunesen.
Reasoning about Reactive Systems.
December 1998.
PhD thesis. xvi+204 pp.

Abstract: The main concern of this thesis is the formal reasoning about reactive systems.

We first consider the automated verification of safety properties of finite systems. We propose a practical framework for integrating the behavioural reasoning about distributed reactive systems with model-checking methods. We devise a small self-contained theory of distributed reactive systems including standard concepts like implementation, abstraction, and proof methods for compositional reasoning. The proof methods are based on trace abstractions that relate the behaviours of the program with the specification. We show that trace abstractions and the proof methods can be expressed in a decidable Monadic Second-Order Logic (M2L) on words. To demonstrate the practical pertinence of the approach, we give a self-contained, introductory account of the method applied to an RPC-memory specification problem proposed by Broy and Lamport. The purely behavioural descriptions which we formulate from the informal specifications are written in the high-level symbolic language FIDO, a syntactic extension of M2L. Our solution involves FIDO-formulas more than 10 pages long. They are translated into M2L-formulas of length more than 100 pages which are decided automatically within minutes. Hence, our work shows that complex behaviours of reactive systems can be formulated and reasoned about without explicit state-based programming, and moreover that temporal properties can be stated succinctly while enjoying automated analysis and verification.

Next, we consider the theoretical borderline of decidability of behavioural equivalences for infinite-state systems. We provide a systematic study of the decidability of non-interleaving linear-

time behavioural equivalences for infinite-state systems defined by CCS and TCSP style process description languages. We compare standard language equivalence with two generalisations based on the predominant approaches for capturing non-interleaving behaviour: pomsets representing global causal dependency, and locality representing spatial distribution of events. Beginning with the process calculus of Basic Parallel Processes (BPP) obtained as a minimal concurrent extension of finite processes, we systematically investigate extensions towards full CCS and TCSP. Our results show that whether a non-interleaving equivalence is based on global causal dependency between events or whether it is based on spatial distribution of events can have an impact on decidability. Furthermore, we investigate tau-forgetting versions of the two non-interleaving equivalences, and show that for BPP they are decidable.

Finally, we address the issue of synthesising distributed systems – modelled as elementary net systems – from purely sequential behaviours represented by synchronisation trees. Based on the notion of regions, Ehrenfeucht and Rozenberg have characterised the transition systems that correspond to the behaviour of elementary net systems. Building upon their results, we characterise the synchronisation trees that correspond to the behaviour of active elementary net systems, that is, those in which each condition can always cease to hold. The identified class of synchronisation trees is definable in a monadic second order logic over infinite trees. Hence, our work provides a theoretical foundation for smoothly combining techniques for the synthesis of nets from transition systems with the synthesis of synchronisation trees from logical specifications.

[DS-98-2] *PostScript*, *PDF*.
Søren B. Lassen.
Relational Reasoning about Functions and Nondeterminism.
December 1998.
PhD thesis. x+126 pp.

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

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

Part II extends the results from the first part to nondeterministic functional programs. May and must operational semantics

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

[DS-98-1] *PostScript*, *PDF*.
Ole I. Hougaard.
The CLP(OIH) Language.
February 1998.
PhD thesis. xii+187 pp.

Abstract: Many type inference problems are different instances of the same constraint satisfaction problem. That is, there is a class of constraints so that these type inference problems can be reduced to the problem of finding a solution to a set of constraints. Furthermore, there is an efficient constraint solving algorithm which can find this solution in polynomial time.

We have shown this by defining the appropriate constraint domain, devising an efficient constraint satisfaction algorithm, and designing a constraint logic programming language over the constraint domain. We have implemented an interpreter for the language and have thus produced a tool which is well-suited for type inference problems of different kinds.

Among the problems that can be reduced to our constraint domain are the following:

- The simply typed λ-calculus.
- The λ-calculus with subtyping.
- Arbadi and Cardelli's Object calculus.
- Effect systems for control flow analysis.
- Turbo Pascal.

With the added power of the constraint logic programming language, certain type systems with no known efficient algorithm can also be implemented — e.g. the object calculus with self-type.

The programming language thus provides a very easy way of implementing a vast array of different type inference problems.

[DS-97-3] *PostScript*, *PDF*.
Thore Husfeldt.
Dynamic Computation.
December 1997.
PhD thesis. 90 pp.

Abstract: Chapter 1 contains a brief introduction and motivation of dynamic computations, and illustrates the main computational models used throughout the thesis, the random access

machine and the *cell probe model* introduced by Fredman.

Chapter 2 paves the road to proving lower bounds for several dynamic problems. In particular, the chapter identifies a number of key problems which are hard for dynamic computations, and to which many other dynamic problems can be reduced. The main contribution of this chapter can be summarised in two results. The first shows that the signed prefix sum problem, which has already been heavily exploited for proving lower bounds on dynamic algorithms and data structures, remains hard even when we provide some amount of nondeterminism to the query algorithms. The second result studies the amount of extra information that can be provided to the query algorithm without affecting the lower bound. Some applications of these results are contained in this chapter; in addition, they are heavily developed for the lower bound proofs in the remainder of the thesis.

Chapter 3 investigates the dynamic complexity of the symmetric Boolean functions, and provides upper and lower bounds. These results establish links between parallel complexity (namely, Boolean circuit complexity) and dynamic complexity. In particular, it is shown that the circuit depth of any symmetric function and the dynamic prefix problem for the same function depend on the same combinatorial properties. The connection between these two different modes and models of computation is shown to be very strong in that the trade-off between circuit size and circuit depth is similar to the trade-off between update and query time.

Chapter 4 considers dynamic graph problems. In particular, it presents the fastest known algorithm for dynamic reachability on planar acyclic digraphs with one source and one sink (known as planar *st-graphs*). Previous partial solutions to this problem were known. In the second part of the chapter, the techniques for lower bound from chapter 2 are further exploited to yield new hardness results for a number of graph problems, including reachability problems in planar graphs and grid graphs, dynamic upward planarity testing and monotone point location.

Chapter 5 turns to strings, and focuses on the problem of maintaining a string of parentheses, known as the dynamic membership problem for the Dyck languages. Parentheses are inserted and removed dynamically, while the algorithm has to check whether the string is properly balanced. It is shown that this problem can be solved in polylogarithmic time per operation. The lower bound techniques from the thesis are again used to prove the hardness of this problem.

[DS-97-2] *PostScript*, *PDF*.
Peter Ørbæk.
Trust and Dependence Analysis.
July 1997.
PhD thesis. x+175 pp.
Abstract: The two pillars of *trust analysis* and *dependence algebra*

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

We have developed trust analyses for imperative languages with pointers as well as for higher order functional languages, utilizing both abstract interpretation, constraint generation and type inference.

The mathematical theory of dependence algebra has been developed and investigated. Two applications of dependence algebra have been given: a practical implementation of a trust analysis for the C programming language, and a soft type inference system for action semantic specifications. Soundness results for the two analyses have been established.

[DS-97-1] *PostScript*, *PDF*.
Gerth Stølting Brodal.
Worst Case Efficient Data Structures.
January 1997.
PhD thesis. x+121 pp.

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

The *node copying* technique of Driscoll *et al.* to make bounded indegree and out-degree data structures partially persistent, supports update steps in amortized constant time and access steps in worst case constant time. We show how to extend the technique such that update steps can be performed in worst case constant time on the pointer machine model.

We present two comparison based priority queue implementations. The first implementation supports FindMin, Insert and Meld in worst case constant time and Delete and DeleteMin in worst case time $O(\log n)$. The second implementation achieves the same performance, but furthermore supports DecreaseKey in worst case constant time. We show that these time bounds are optimal for all implementations supporting Meld in worst case time $o(n)$. We show that any randomized implementation with expected amortized cost t comparisons per Insert and Delete operation has expected cost at least $n/2^{O(t)}$ comparisons for FindMin.

For the CREW PRAM we present time and work optimal priority queues, supporting FindMin, Insert, Meld, DeleteMin, Delete and DecreaseKey in constant time with $O(\log n)$ processors. To be able to speed up Dijkstra's algorithm for the single-source shortest path problem we present a different parallel priority data structure yielding an implementation of Dijkstra's algorithm which runs in $O(n)$ time and performs $O(m \log n)$ work on a CREW PRAM.

On a unit cost RAM model with word size w bits we consider the maintenance of a set of n integers from $0..2^w - 1$ under INSERT, DELETE, FINDMIN, FINDMAX and PRED (predecessor query). The RAM operations used are addition, left and right bit shifts, and bit-wise boolean operations. For any function $f(n)$ satisfying $\log\log n \leq f(n) \leq \sqrt{\log n}$, we support FINDMIN and FINDMAX in constant time, INSERT and DELETE in $O(f(n))$ time, and PRED in $O((\log n)/f(n))$ time.

The last problem we consider is given a set of n binary strings of length m each, to answer d–queries, *i.e.*, given a binary string of length m to report if there exists a string in the set within Hamming distance d of the string. We present a data structure of size $O(nm)$ supporting 1–queries in time $O(m)$ and the reporting of all strings within Hamming distance 1 of the query string in time $O(m)$. The data structure can be constructed in time $O(nm)$ and supports the insertion of new strings in amortized time $O(m)$.

[DS-96-4] *PostScript*, *PDF*.
Torben Braüner.
An Axiomatic Approach to Adequacy.
November 1996.
Ph.D. thesis. 168 pp.

Abstract: This thesis studies adequacy for PCF-like languages in a general categorical framework. An adequacy result relates the denotational semantics of a program to its operational semantics; it typically states that a program terminates whenever the interpretation is non-bottom. The main concern is to generalise to a linear version of PCF the adequacy results known for standard PCF, keeping in mind that there exists a Girard translation from PCF to linear PCF with respect to which the new constructs should be sound. General adequacy results have usually been obtained in order-enriched categories, that is, categories where all hom-sets are typically cpos, maps are assumed to be continuous and fixpoints are interpreted using least upper bounds. One of the main contributions of the thesis is to propose a completely different approach to the problem of axiomatising those categories which yield adequate semantics for PCF and linear PCF. The starting point is that the only unavoidable assumption for dealing with adequacy is the existence, in any object, of a particular "undefined" point denoting the non-terminating computation of the corresponding type; hom-sets are pointed sets, and this is the only required structure. In such a category of pointed objects, there is a way of axiomatising fixpoint operators: They are maps satisfying certain equational axioms, and furthermore, satisfying a very natural non-equational axiom called rational openness. It is shown that this axiom is sufficient, and in a precise sense necessary, for adequacy. The idea is developed in the intuitionistic case (standard PCF) as well as in the linear case (linear PCF), which is obtained by augmenting a Curry-Howard interpretation of Intuitionistic Linear Logic with numerals and

fixpoint constants, appropriate for the linear context. Using instantiations to concrete models of the general adequacy results, various purely syntactic properties of linear PCF are proved to hold.

[DS-96-3] *PostScript*, *PDF*.
Lars Arge.
Efficient External-Memory Data Structures and Applications.
August 1996.
Ph.D. thesis. xii+169 pp.

Abstract: In this thesis we study the Input/Output (I/O) complexity of large-scale problems arising e.g. in the areas of database systems, geographic information systems, VLSI design systems and computer graphics, and design I/O-efficient algorithms for them.

A general theme in our work is to design I/O-efficient algorithms through the design of I/O-efficient data structures. One of our philosophies is to try to design I/O algorithms from internal memory algorithms by exchanging the data structures used in internal memory with their external memory counterparts. The results in the thesis include a technique for transforming an internal memory tree data structure into an external data structure which can be used in a *batched* dynamic setting, that is, a setting where we for example do not require that the result of a search operation is returned immediately. Using this technique we develop batched dynamic external versions of the (one-dimensional) range-tree and the segment-tree and we develop an external priority queue. These structures can be used in standard internal memory sorting algorithms and algorithms for problems involving geometric objects. The latter has applications to VLSI design. Using the priority queue we improve upon known I/O algorithms for fundamental graph problems, and develop new efficient algorithms for the graph-related problem of ordered binary-decision diagram manipulation. Ordered binary-decision diagrams are the state-of-the-art data structure for boolean function manipulation and they are extensively used in large-scale applications like logic circuit verification.

Combining our batched dynamic segment tree with the novel technique of external-memory fractional cascading we develop I/O-efficient algorithms for a large number of geometric problems involving line segments in the plane, with applications to geographic informations systems. Such systems frequently handle huge amounts of spatial data and thus they require good use of external-memory techniques.

We also manage to use the ideas in the batched dynamic segment tree to develop "on-line" external data structures for a special case of two-dimensional range searching with applications to databases and constraint logic programming. We develop an on-line external version of the segment tree, which improves upon the previously best known such structure, and an optimal on-

line external version of the interval tree. The last result settles an important open problem in databases and I/O algorithms. In order to develop these structure we use a novel balancing technique for search trees which can be regarded as weight-balancing of B-trees.

Finally, we develop a technique for transforming internal memory lower bounds to lower bounds in the I/O model, and we prove that our new ordered binary-decision diagram manipulation algorithms are asymptotically optimal among a class of algorithms that include all know manipulation algorithms.

[DS-96-2] *PostScript*, *PDF*.
Allan Cheng.
Reasoning About Concurrent Computational Systems.
August 1996.
Ph.D. thesis. xiv+229 pp.

Abstract: This thesis consists of three parts. The first presents contributions in the field of verification of finite state concurrent systems, the second presents contributions in the field of behavioural reasoning about concurrent systems based on the notions of behavioural preorders and behavioural equivalences, and, finally, the third presents contributions in the field of set constraints.

In the first part, we study the computational complexity of several standard verification problems for 1-safe Petri nets and some of its subclasses. Our results provide the first systematic study of the computational complexity of these problems.

We then investigate the computational complexity of a more general verification problem, model-checking, when an instance of the problem consists of a formula and a description of a system whose state space is at most exponentially larger than the description.

We continue by considering the problem of performing model-checking relative to a partial order semantics of concurrent systems, in which not all possible sequences of actions are considered relevant. We provide the first, to the best of our knowledge, set of sound and complete tableau rules for a CTL-like logic interpreted under progress fairness assumptions.

In the second part, we investigate Joyal, Nielsen, and Winskel's proposal of spans of open maps as an abstract category-theoretic way of adjoining a bisimulation equivalence, $\mathcal{P}$-bisimilarity, to a category of models of computation $\mathcal{M}$. We show that a representative selection of well-known bisimulations and behavioural equivalences can be captured in the setting of spans of open maps.

We then define the notion of functors being $\mathcal{P}$-factorisable and show how this ensures that $\mathcal{P}$-bisimilarity is a congruence with respect to such functors.

In the last part we investigate set constraints. Set constraints are

inclusion relations between expressions denoting sets of ground terms over a ranked alphabet. They are typically derived from the syntax of a program and solutions to them can yield useful information for, e.g., type inference, implementations, and optimisations.

We provide a complete Gentzen-style axiomatisation for sequents $\Phi \vdash \Psi$, where Φ and Ψ are finite sets of set constraints, based on the axioms of termset algebra. We show that the deductive system is complete for the restricted sequents $\Phi \vdash \perp$ over standard models, incomplete for general sequents $\Phi \vdash \Psi$ over standard models, but complete for general sequents over set-theoretic termset algebras.

We then continue by investigating Kozen's rational spaces. We give several interesting results, among others a Myhill-Nerode like characterisation of rational points.

[DS-96-1] *PostScript*, *PDF*.

Urban Engberg.

Reasoning in the Temporal Logic of Actions — The design and implementation of an interactive computer system.

August 1996.

Ph.D. thesis. xvi+222 pp.

Abstract: Reasoning about algorithms stands out as an essential challenge of computer science. Much work has been put into the development of formal methods, within recent years focusing especially on concurrent algorithms. The Temporal Logic of Actions (TLA) is one of the formal frameworks that has been the result of such research. In TLA a system and its properties may be specified as logical formulas, allowing the application of reasoning without any intermediate translation. Being a linear-time temporal logic, it easily expresses liveness as well as safety properties.

TLP, the TLA Prover, is a system developed by the author for doing mechanical reasoning in TLA. TLP is a complex system that combines different reasoning techniques to be able to handle verification of real systems. It uses the Larch Prover as its main verification back-end, but as well makes it possible to utilize results provided by other back-ends, such as an automatic linear-time temporal logic checker for finite-state systems. Specifications to be verified within the TLP system are written in a dedicated language, an extension of pure TLA. The language as well contains syntactical constructs for representing natural deduction proofs, and a separate language for defining methods to be applied during verification. With the TLP translators and interactive front-end, it is possible to write specifications and incrementally develop mechanically verifiable proofs in a fashion that has not been seen before.