BRICS Research Series, Abstracts, 1995

July 9, 2003

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

References

[RS-95-60] *PostScript*.
Jrgen H. Andersen, Carsten H. Kristensen, and Arne Skou.
Specification and Automated Verification of Real-Time Behaviour — A Case Study.
December 1995.
24 pp. Appears in *3rd IFAC/IFIP workshop on Algoritms and Architectures for Real-Time Control*, AARTC '95 Proceedings, 1995, pages 613–628 and in *Annual Reviews of Control*, 20:55–70, 1996.
Abstract: In this paper we sketch a method for specification and automatic verification of real-time software properties. The method combines the IEC 848 norm and the recent specification techniques TCCS (Timed Calculus of Communicating Systems) and TML (Timed Modal Logic) — supported by an automatic verification tool, EPSILON. The method is illustrated by modelling a small real-life steam generator example and subsequent automated analysis of its properties.

[RS-95-59] *PostScript*, *DVI*.
Luca Aceto and Anna Inglfsdttir.
On the Finitary Bisimulation.
November 1995.
29 pp.
Abstract: The finitely observable, or *finitary*, part of bisimulation is a key tool in establishing full abstraction results for denotational semantics for process algebras with respect to bisimulation-based preorders. A bisimulation-like characterization of this relation for arbitrary transition systems is given, relying on Abramsky's characterization in terms of the finitary domain logic. More informative behavioural, observation-independent characterizations of the finitary bisimulation are also offered for several interesting classes of transition systems. These include transition systems with countable action sets, those that have bounded convergent sort and the sort-finite ones. The result for sort-finite transition systems sharpens a previous

behavioural characterization of the finitary bisimulation for this class of structures given by Abramsky.

[RS-95-58] *PostScript*, *DVI*.
Nils Klarlund, Madhavan Mukund, and Milind Sohoni.
Determinizing Asynchronous Automata on Infinite Inputs.
November 1995.
32 pp. Appears in Thiagarajan, editor, *Foundations of Software Technology and Theoretical Computer Science: 15th Conference*, FST&TCS '95 Proceedings, LNCS 1026, 1995, pages 456–471.

Abstract: Asynchrono us automata are a natural distributed machine model for recognizing *trace languages*—languages defined over an alphabet equipped with an independence relation.

To handle *infinite* traces, Gastin and Petit introduced Bchi asynchronous automata, which accept precisely the class of ω-regular trace languages. Like their sequential counterparts, these automata need to be non-deterministic in order to capture all ω-regular languages. Thus complementation of these automata is non-trivial. Complementation is an important operation because it is fundamental for treating the logical connective "not" in decision procedures for monadic secondorder logics.

Subsequently, Diekert and Muscholl solved the complementation problem by showing that with a Muller acceptance condition, deterministic automata suffice for recognizing ω-regular trace languages. However, a direct determinization procedure, extending the classical subset construction, has proved elusive.

In this paper, we present a direct determinization procedure for Bchi asynchronous automata, which generalizes Safra's construction for sequential Bchi automata. As in the sequential case, the blow-up in the state space is essentially that of the underlying subset construction.

[RS-95-57] *PostScript*, *DVI*.
Jaap van Oosten.
Topological Aspects of Traces.
November 1995.
16 pp. Appears in Billington and Reisig, editors, *17th International Conference on Application and Theory of Petri Nets*, ICATPN '96 Proceedings, LNCS 1091, 1996, pages 480–496.

Abstract: Traces play a major role in several models of concurrency. They arise out of "independence structures" which are sets with a symmetric, irreflexive relation. In this paper, independence structures are characterized as certain topological spaces. We show that these spaces are a universal construction known as "soberification", a topological generalization of the ideal completion construction in domain theory. We also show that there is an interesting group action connected to this construction. Finally, generalizing the constructions in the first part of the paper, we define a new category of "labelled systems of posets". This category includes labelled event structures as a full reflec-

tive subcategory, and has moreover a very straightforward notion of bisimulation which restricts on event structures to strong history-preserving bisimulation.

[RS-95-56] *PostScript*, *DVI*.

Luca Aceto, Willem Jan Fokkink, Rob J. van Glabbeek, and Anna Inglfsdttir.

Axiomatizing Prefix Iteration with Silent Steps.

November 1995.

25 pp. Appears in Bjerner, Larsson and Nordstrm, editors, *7th Nordic Workshop on Programming Theory*, NWPT '7 Proceedings, Programming Methodology Group's Report Series Report 86, January 1996, 1995, and in *Information and Computation*, 127(1):26–40, May 1996.

Abstract: Prefix iteration is a variation on the original binary version of the Kleene star operation P^*Q, obtained by restricting the first argument to be an atomic action. The interaction of prefix iteration with silent steps is studied in the setting of Milner's basic CCS. Complete equational axiomatizations are given for four notions of behavioural congruence overbasic CCS with prefix iteration, viz. branching congruence, η-congruence, delay congruence and weak congruence. The completeness proofs for η-, delay, and weak congruence are obtained by reduction to the completeness theorem for branching congruence. It is also argued that the use of the completeness result for branching congruence in obtaining the completeness result for weak congruence leads to a considerable simplification with respect to the only direct proof presented in the literature. The preliminaries and the completeness proofs focus on open terms, i.e., terms that may contain process variables. As a byproduct, the ω-completeness of the axiomatizations is obtained as well as their completeness for closed terms.

[RS-95-55] *PostScript*.

Mogens Nielsen and Kim Sunesen.

Behavioural Equivalence for Infinite Systems — Partially Decidable!

November 1995.

38 pp. Full version of paper appearing in Billington and Reisig, editors, *17th International Conference on Application and Theory of Petri Nets*, ICATPN '96 Proceedings, LNCS 1091, 1996, pages 460–479.

Abstract: For finite-state systems non-interleaving equivalences are computationally at least as hard as interleaving equivalences. In this paper we show that when moving to infinite-state systems, this situation may change dramatically.

We compare standard language equivalence for process description languages with two generalizations based on traditional approaches capturing non-interleaving behaviour, *pomsets* representing global causal dependency, and *locality* representing spatial distribution of events.

We first study equivalences on Basic Parallel Processes, BPP, a process calculus equivalent to communication-free Petri nets. For this simple process language our two notions of non-interleaving equivalences agree. More interestingly, we show that they are decidable, contrasting a result of Hirshfeld that standard interleaving language equivalence is undecidable. Our result is inspired by a recent result of Esparza and Kiehn, showing the same phenomenon in the setting of model checking.

We follow up investigating to which extent the result extends to larger subsets of CCS and TCSP. We discover a significant difference between our non-interleaving equivalences. We show that for a certain non-trivial subclass of processes between BPP and TCSP, not only are the two equivalences different, but one (*locality*) is decidable whereas the other (*pomsets*) is not. The decidability result for *locality* is proved by a reduction to the reachability problem for Petri nets.

Comments: Previously announced under the title "Trace Equivalence — Partially Decidable!".

[RS-95-54] *PostScript*, *DVI*.

Nils Klarlund, Mogens Nielsen, and Kim Sunesen.

A Case Study in Automated Verification Based on Trace Abstractions.

November 1995.

35 pp. Full version appears in Broy, Merz and Spies, editors, *Formal Systems Specification: The RPC-Memory Specification Case Study*, FSS '96 Selected Solutions, LNCS 1169, 1996, pages 341–374, under the title *Using Monadic Second-Order Logic over Finite Domains for Specification and Verification.*

Abstract: In a previous paper [PODC'96], we proposed a framework for the automatic verification of reactive systems. Our main tool is a decision procedure, MONA, for Monadic Second-order Logic (M2L) on finite strings. MONA translates a formula in M2L into a finite-state automaton. We show in [PODC'96] how *traces*, i.e. finite executions, and their abstractions can be described behaviorally. These state-less descriptions can be formulated in terms of customized temporal logic operators or idioms.

In the present paper, we give a self-contained, introductory account of our method applied to the RPC-memory specification problem of the 1994 Dagstuhl Seminar on Specification and Refinement of Reactive Systems. The purely behavioral descriptions that we formulate from the informal specifications are formulas that may span 10 pages or more.

Such descriptions are a couple of magnitudes larger than usual temporal logic formulas found in the literature on verification. To securely write these formulas, we introduce FIDO as a reactive system description language. FIDO is designed as a high-level symbolic language for expressing regular properties about recursive data structures.

All of our descriptions have been verified automatically by MONA from M2L formulas generated by FIDO.

Our work shows that complex behaviors of reactive systems can be formulated and reasoned about without explicit state-based programming. With FIDO, we can state temporal properties succinctly while enjoying automated analysis and verification.

Comments: Previously announced under the title "Using Monadic Second-Order Logic with Finite Domains for Specification and Verification".

[RS-95-53] *PostScript*, *DVI.*
Nils Klarlund, Mogens Nielsen, and Kim Sunesen.
Automated Logical Verification based on Trace Abstractions.
November 1995.
19 pp. Appears in *The Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '96 Proceedings, 1996, pages 101–110.

Abstract: We propose a new and practical framework for integrating the behavioral reasoning about distributed systems with model-checking methods.

Our proof methods are based on *trace abstractions*, which relate the behaviors of the program and the specification. We show that for finite-state systems such symbolic abstractions can be specified conveniently in Monadic Second-Order Logic (*M2L*). Model-checking is then made possible by the reduction of non-determinism implied by the trace abstraction.

Our method has been applied to a recent verification problem by Broy and Lamport. We have transcribed their behavioral description of a distributed program into temporal logic and verified it against another distributed system without constructing the global program state space. The reasoning is expressed entirely within *M2L* and is carried out by a decision procedure. Thus *M2L* is a practical vehicle for handling complex temporal logic specifications, where formulas decided by a push of a button are as long as 10–15 pages.

[RS-95-52] *PostScript*, *DVI.*
Antonn Kučera.
Deciding Regularity in Process Algebras.
October 1995.
42 pp.

Abstract: We consider the problem of deciding regularity of normed BPP and normed BPA processes. A process is regular if it is bisimilar to a process with finitely many states. We show, that regularity of normed BPP processes is decidable and we provide a constructive regularity test. We also show, that the same result can be obtained for the class of normed BPA processes.

Regularity can be defined also w.r.t. other behavioural equivalences. We define notions of strong regularity and finite characterisation and we examine their relationship with notions of regularity and finite representation. The introduced notion of the finite characterisation is especially interesting from the point of view of possible verification of concurrent systems.

In the last section we present some negative results. If we extend the BPP algebra with the operator of restriction, regularity becomes undecidable and similar results can be obtained also for other process algebras.

[RS-95-51] *PostScript*, *DVI*.
Rowan Davies.
A Temporal-Logic Approach to Binding-Time Analysis.
October 1995.
15 pp. Appears in *Eleventh Annual IEEE Symposium on Logic in Computer Science*, LICS '96 Proceedings, 1996, pages 184–195.

Abstract: The Curry-Howard isomorphism identifies proofs with typed λ-calculus terms, and correspondingly identifies propositions with types. We show how this isomorphism can be extended to relate constructive temporal logic with binding-time analysis. In particular, we show how to extend the Curry-Howard isomorphism to include the $\bigcirc$ ("next") operator from linear-time temporal logic. This yields the simply typed $\lambda^{\bigcirc}$-calculus which we prove to be equivalent to a multi-level binding-time analysis like those used in partial evaluation.

[RS-95-50] *PostScript*, *DVI*.
Dany Breslauer.
On Competitive On-Line Paging with Lookahead.
September 1995.
12 pp. Appears in Puech and Reischuk, editors, *13th Annual Symposium on Theoretical Aspects of Computer Science*, STACS '96 Proceedings, LNCS 1046, 1996, pages 593–603.

Abstract: This paper studies two methods for improving the competitive efficiency of on-line paging algorithms: in the first, the on-line algorithm can use more pages; in the second, it is allowed to have a lookahead, or in other words, some partial knowledge of the future. The paper considers a new measure for the lookahead size as well as Young's resource-bounded lookahead and proves that both measures have the attractive property that the competitive efficiency of an on-line algorithm with k extra pages and lookahead l depends on $k + l$. Hence, under these measures, an on-line algorithm has the same benefit from using an extra page or knowing an extra bit of the future.

[RS-95-49] *PostScript*, *DVI*.
Mayer Goldberg.
Solving Equations in the λ-Calculus using Syntactic Encapsulation.
September 1995.
13 pp.

Abstract: Syntactic encapsulation is a relation between an expression and one of its sub-expressions, that constraints how the given sub-expression can be used throughout the reduction of the expression. In this paper, we present a class of systems of equations, in which the right-hand side of each equation is syntactically encapsulated in the left-hand side. This class is general

enough to allow equations to contain self-application, and to allow unknowns to appear on both sides of the equation. Yet such a system is simple enough to be solvable, and for a solution (though of course not its normal form) to be obtainable in constant time.

Comments: Keywords: λ-calculus, programming calculi.

[RS-95-48] *PostScript*, *DVI*.
Devdatt P. Dubhashi.
Simple Proofs of Occupancy Tail Bounds.
September 1995.
7 pp. Appears in *Random Structures and Algorithms*, 11, 1997.

Abstract: We give short proofs of some occupancy tail bounds using the method of bounded differences in expected form and the notion of negative association.

[RS-95-47] *PostScript*, *DVI*.
Dany Breslauer.
The Suffix Tree of a Tree and Minimizing Sequential Transducers.
September 1995.
15 pp. Appears in Hirschberg and Myers, editors, *Combinatorial Pattern Matching: 7th Annual Symposium*, CPM '96 Proceedings, LNCS 1075, 1996, pages 116–129 and in *Theoretical Computer Science*, 191(1–2):131–144, January 1998.

Abstract: This paper gives a linear-time algorithm for the construction of the suffix tree of a tree. The suffix tree of a tree is used to obtain an efficient algorithm for the minimization of sequential transducers.

[RS-95-46] *PostScript*, *DVI*.
Dany Breslauer, Livio Colussi, and Laura Toniolo.
On the Comparison Complexity of the String Prefix-Matching Problem.
August 1995.
39 pp. Appears in Leeuwen, editor, *Second Annual European Symposium on Algorithms*, ESA '94 Proceedings, LNCS 855, 1994, pages 483–494.

Abstract: In this paper we study the exact comparison complexity of the string prefix-matching problem in the deterministic sequential comparison model with equality tests. We derive almost tight lower and upper bounds on the number of symbol comparisons required in the worst case by on-line prefix-matching algorithms for any fixed pattern and variable text. Unlike previous results on the comparison complexity of string-matching and prefix-matching algorithms, our bounds are almost tight for any particular pattern.

We also consider the special case where the pattern and the text are the same string. This problem, which we call the *string self-prefix* problem, is similar to the pattern preprocessing step of the Knuth-Morris-Pratt string-matching algorithm that is used in several comparison efficient string-matching and prefix-matching algorithms, including in our new algorithm. We obtain roughly tight lower and upper bounds on the number of symbol comparisons required in the worst case by on-line self-prefix algorithms.

Our algorithms can be implemented in linear time and space in the standard uniform-cost random-access-machine model.

[RS-95-45] *PostScript*.
Gudmund Skovbjerg Frandsen and Sven Skyum.
Dynamic Maintenance of Majority Information in Constant Time per Update.
August 1995.
9 pp. Revised version appears in *Information Processing Letters* vol. 63 (1997), pages 75–78.

Abstract: We show how to maintain information about the existence of a majority colour in a set of elements under insertion and deletion of single elements using $O(1)$ time and at most 4 equality tests on colours per update. No ordering information is used.

[RS-95-44] *PostScript*.
Bruno Courcelle and Igor Walukiewicz.
Monadic Second-Order Logic, Graphs and Unfoldings of Transition Systems.
August 1995.
39 pp. Presented at the *9th Annual Conference of the European Association for Computer Science Logic*, CSL '95. Journal version appears in *Annals of Pure and Applied Logic*, 92(1):35–62, 1998.

Abstract: We prove that every monadic second-order property of the unfolding of a transition system is a monadic second-order property of the system itself. We prove a similar result for certain graph coverings.

[RS-95-43] *PostScript*, *DVI*.
Noam Nisan and Avi Wigderson.
Lower Bounds on Arithmetic Circuits via Partial Derivatives (Preliminary Version).
August 1995.
17 pp. Appears in *36th Annual Symposium on Foundations of Computer Science*, FOCS '95 Proceedings, 1995, pages 16–25. Journal version in *Computational Complexity*, 6(3):217–234, 1997.

Abstract: In this paper we describe a new technique for obtaining lower bounds on restriced classes of nonmonotone arithmetic circuits. The heart of this technique is a complexity measure for multivariate polynomials, based on the linear span of their partial derivatives. We use the technique to obtain new lower bounds for computing symmetric polynomials and iterated matrix products.

[RS-95-42] *PostScript*, *DVI*.
Mayer Goldberg.
An Adequate Left-Associated Binary Numeral System in the λ-Calculus.
August 1995.
16 pp. Revised version Also available as BRICS Report R̈S-96-6. The revised version appears in *Journal of Functional Programming*

10(6):607–623, 2000.

Abstract: This paper introduces a sequence of λ-expressions, modelling the binary expansion of integers. We derive expressions computing the test for zero, the successor function, and the predecessor function, thereby showing the sequence to be an adequate numeral system. These functions can be computed efficiently. Their complexity is independent of the order of evaluation.

[RS-95-41] Olivier Danvy, Karoline Malmkjr, and Jens Palsberg.
Eta-Expansion Does The Trick.
August 1995.
23 pp. Please refer to the revised version BRICS-RS-96-17. Now appears in *ACM Transactions on Programming Languages and Systems*, 8(6):730–751, 1996.

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

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

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

[RS-95-40] *PostScript, DVI.*
Anna Inglfsdttir and Andrea Schalk.
A Fully Abstract Denotational Model for Observational Congruence.
August 1995.
29 pp. Appears with the title *A Fully Abstract Denotational Model for Observational Precongruence* in Bning, editor, *European Association for Computer Science Logic: 9th Workshop*, CSL '95 Selected Papers, LNCS 1092, 1996, pages 335–361. Appears also in *Theoretical Computer Science* 254(1–2):35–61, 2001.

Abstract: A domain theoretical denotational model is given for a simple sublanguage of *CCS* extended with divergence operator. The model is derived as an abstraction on a suitable notion of normal forms for labelled transition systems. It is shown to be fully abstract with respect to observational precongruence.

[RS-95-39] *PostScript.*
Allan Cheng.
Petri Nets, Traces, and Local Model Checking.
July 1995.
32 pp. Full version of paper appearing in Alagar and Nivat, editors, *Algebraic Methodology and Software Technology: 4th International Conference*, AMAST '95 Proceedings, LNCS 936, 1995, pages 322-337. Appears also in *Theoretical Computer Science*, 183(2):229–251, (1997).

Abstract: It has been observed that the behavioural view of concurrent systems that all possible sequences of actions are relevant is too generous; not all sequences should be considered as likely behaviours. Taking progress fairness assumptions into account one obtains a more realistic behavioural view of the systems. In this paper we consider the problem of performing model checking relative to this behavioural view. We present a CTL-like logic which is interpreted over the model of concurrent systems labelled 1-safe nets. It turns out that Mazurkiewicz trace theory provides a natural setting in which the progress fairness assumptions can be formalized. We provide the first, to our knowledge, set of sound and complete tableau rules for a CTL-like logic interpreted under progress fairness assumptions.

[RS-95-38] Mayer Goldberg.
Gdelisation in the λ-Calculus.
July 1995.
7 pp. Appears in *Information Processing Letters* 75(1–2):13–16(2000). Please refer to the revised version BRICS-RS-96-5.

Abstract: Gdelisation is a meta-linguistic encoding of terms in a language. While it is impossible to define an operator in the λ-calculus which encodes all closed λ-expressions, it is possible to construct restricted versions of such an encoding operator. In this paper, we propose such an encoding operator for proper combinators.

[RS-95-37] *PostScript*, *DVI.*
Sten Agerholm and Mike Gordon.
Experiments with ZF Set Theory in HOL and Isabelle.
July 1995.
14 pp. Appears in Schubert, Windley and Alves-Foss, editors, *Higher Order Logic Theorem Proving and its Applications: 8th International Workshop*, HOLTPA '95 Proceedings, LNCS 971, 1995, pages 32-45.

Abstract: Most general purpose proof assistants support versions of typed higher order logic. Experience has shown that these logics are capable of representing most of the mathematical models needed in Computer Science. However, perhaps there exist applications where ZF-style set theory is more natural, or even necessary. Examples may include Scott's classical inverse-limit

construction of a model of the untyped λ-calculus (D_∞) and the semantics of parts of the Z specification notation. This paper compares the representation and use of ZF set theory within both HOL and Isabelle. The main case study is the construction of D_∞. The advantages and disadvantages of higher-order set theory versus first-order set theory are explored experimentally. This study also provides a comparison of the proof infrastructure of HOL and Isabelle.

[RS-95-36] *PostScript*, *DVI*.
Sten Agerholm.
Non-Primitive Recursive Function Definitions.
July 1995.
15 pp. Appears in Schubert, Windley and Alves-Foss, editors, *Higher Order Logic Theorem Proving and its Applications: 8th International Workshop*, HOLTPA '95 Proceedings, LNCS 971, 1995, pages 17-31.

Abstract: This paper presents an approach to the problem of introducing non-primitive recursive function definitions in higher order logic. A recursive specification is translated into a domain theory version, where the recursive calls are treated as potentially non-terminating. Once we have proved termination, the original specification can be derived easily. A collection of algorithms are presented which hide the domain theory from a user. Hence, the derivation of a domain theory specification has been automated completely, and for well-founded recursive function specifications the process of deriving the original specification from the domain theory one has been automated as well, though a user must supply a well-founded relation and prove certain termination properties of the specification. There are constructions for building well-founded relations easily.

[RS-95-35] *PostScript*, *DVI*.
Mayer Goldberg.
Constructing Fixed-Point Combinators Using Application Survival.
June 1995.
14 pp.

Abstract: The theory of *application survival* was developed in our Ph.D. thesis as an approach for reasoning about application in general and self-application in particular. In this paper, we show how application survival provides a uniform framework not only for for reasoning about fixed-points, fixed-point combinators, but also for deriving and comparing known and new fixed-point combinators.

[RS-95-34] *PostScript*, *DVI*.
Jens Palsberg.
Type Inference with Selftype.
June 1995.
22 pp.

Abstract: The metavariable *self* is fundamental in object-oriented languages. Typing self in the presence of inheritance has been studied by Abadi and Cardelli, Bruce, and others. A key concept in these developments is the notion of *selftype*, which enables flexible type annotations that are impossible with recursive types and subtyping. Bruce et al. demonstrated that, for the language TOOPLE, type checking is decidable. Open until now is the problem of type inference with selftype.

In this paper we present a type inference algorithm for a type system with selftype, recursive types, and subtyping. The example language is the object calculus of Abadi and Cardelli, and the type inference algorithm runs in nondeterministic polynomial time.

[RS-95-33] *PostScript*, *DVI*.
Jens Palsberg, Mitchell Wand, and Patrick O'Keefe.
Type Inference with Non-structural Subtyping.
June 1995.
22 pp. Appears in *Formal Aspects of Computing*, 9(1):49–67, 1997.

Abstract: We present an $O(n^3)$ time type inference algorithm for a type system with a largest type $\top$, a smallest type $\perp$, and the usual ordering between function types. The algorithm infers type annotations of minimal size, and it works equally well for recursive types. For the problem of typability, our algorithm is simpler than the one of Kozen, Palsberg, and Schwartzbach for type inference *without* $\perp$. This may be surprising, especially because the system with $\perp$ is strictly more powerful.

[RS-95-32] *PostScript*, *DVI*.
Jens Palsberg.
Efficient Inference of Object Types.
June 1995.
32 pp. Appears in *Information and Computation*, 123(2):198–209, 1995. Preliminary version appears in *Ninth Annual IEEE Symposium on Logic in Computer Science*, LICS '94 Proceedings, 1994, pages 186–195.

Abstract: Abadi and Cardelli have recently investigated a calculus of objects. The calculus supports a key feature of object-oriented languages: an object can be emulated by another object that has more refined methods. Abadi and Cardelli presented four first-order type systems for the calculus. The simplest one is based on finite types and no subtyping, and the most powerful one has both recursive types and subtyping. Open until now is the question of type inference, and in the presence of subtyping "the absence of minimum typings poses practical problems for type inference".

In this paper we give an $O(n^3)$ algorithm for each of the four type inference problems and we prove that all the problems are P-complete. We also indicate how to modify the algorithms to handle functions and records.

[RS-95-31] *PostScript*, *DVI*.
Jens Palsberg and Peter rbk.

Trust in the λ-calculus.
June 1995.
32 pp. Appears in Mycroft, editor, *2nd International Static Analysis Symposium*, SAS '95 Proceedings, LNCS 983, 1995, pages 314–330.

Abstract: This paper introduces trust analysis for higher-order languages. Trust analysis encourages the programmer to make explicit the trustworthiness of data, and in return it can guarantee that no mistakes with respect to trust will be made at run-time. We present a confluent λ-calculus with explicit trust operations, and we equip it with a trust-type system which has the subject reduction property. Trust information in presented as two annotations of each function type constructor, and type inference is computable in $O(n^3)$ time.

[RS-95-30] *PostScript*, *DVI.*
Franck van Breugel.
From Branching to Linear Metric Domains (and back).
June 1995.
30 pp. Abstract appeared in Engberg, Larsen and Mosses, editors, *6th Nordic Workshop on Programming Theory*, NWPT '6 Proceedings, BRICS Notes Series NS-94-6, December 1994, 1994, pages 444–447.

Abstract: A branching and a linear metric domain—both turned into a category—are related by means of a reflection and a coreflection.

[RS-95-29] *PostScript.*
Nils Klarlund.
An $n \log n$ Algorithm for Online BDD Refinement.
May 1995.
20 pp. Conference version in Grumberg, editor, *Computer-Aided Verification: 9th International Conference*, CAV '97 Proceedings, LNCS 1254, 1997, pages 107–118. Journal version in *Journal of Algorithms*, 32(2):133-154, 1999.

Abstract: Binary Decision Diagrams are in widespread use in verification systems for the canonical representation of Boolean functions. A BDD representing a function $\varphi : I\!B^\nu \rightarrow I\!N$ can easily be reduced to its canonical form in linear time.

In this paper, we consider a natural online BDD refinement problem and show that it can be solved in $O(n \log n)$ if n bounds the size of the BDD and the total size of update operations.

We argue that BDDs in an algebraic framework should be understood as minimal fixed points superimposed on maximal fixed points. We propose a technique of controlled growth of equivalence classes to make the minimal fixed point calculations be carried out efficiently. Our algorithm is based on a new understanding of the interplay between the splitting and growing of classes of nodes.

We apply our algorithm to show that automata with exponen-

tially large, but implicitly represented alphabets, can be minimized in time $O(n \log n)$, where n is the total number of BDD nodes representing the automaton.

[RS-95-28] *PostScript*, *DVI*.
Luca Aceto and Jan Friso Groote.
A Complete Equational Axiomatization for MPA with String Iteration.
May 1995.
39 pp. Appears in *Theoretical Computer Science*, 211(1–2):339–374, January 1999.

Abstract: We study equational axiomatizations of bisimulation equivalence for the language obtained by extending Milner's basic CCS with string iteration. String iteration is a variation on the original binary version of the Kleene star operation p^*q obtained by restricting the first argument to be a non-empty sequence of atomic actions. We show that, for every positive integer k, bisimulation equivalence over the set of processes in this language with loops of length at most k is finitely axiomatizable. We also offer a countably infinite equational theory that completely axiomatizes bisimulation equivalence over the whole language. We prove that this result cannot be improved upon by showing that no finite equational axiomatization of bisimulation equivalence over basic CCS with string iteration can exist, unless the set of actions is empty.

[RS-95-27] *PostScript*, *DVI*.
David Janin and Igor Walukiewicz.
Automata for the μ-calculus and Related Results.
May 1995.
11 pp. Appears in Wiedermann and Hjek, editors, *Mathematical Foundations of Computer Science: 20th International Symposium*, MFCS '95 Proceedings, LNCS 969, 1995, pages 552-562.

Abstract: The propositional μ-calculus as introduced by Kozen in [TCS 27] is considered. The notion of disjunctive formula is defined and it is shown that every formula is semantically equivalent to a disjunctive formula. For these formulas many difficulties encountered in the general case may be avoided. For instance, satisfiability checking is linear for disjunctive formulas. This kind of formula gives rise to a new notion of finite automaton which characterizes the expressive power of the μ-calculus over all transition systems.

[RS-95-26] *PostScript*, *DVI*.
Faith Fich and Peter Bro Miltersen.
Tables Should Be Sorted (on Random Access Machines).
May 1995.
11 pp. Appears in Akl, Dehne, Sack and Santoro, editors, *Algorithms and Data Structures: 4th Workshop*, WADS '95 Proceedings, LNCS 955, 1995, pages 482–493.

Abstract: We consider the problem of storing an n element subset S of a universe of size m, so that membership queries (is $x \in S$?)

can be answered efficiently. The model of computation is a random access machine with the standard instruction set (direct and indirect adressing, conditional branching, addition, subtraction, and multiplication). We show that if s memory registers are used to store S, where $n \leq s \leq m/n^{\epsilon}$, then query time $\Omega(\log n)$ is necessary in the worst case. That is, under these conditions, the solution consisting of storing S as a sorted table and doing binary search is optimal. The condition $s \leq m/n^{\epsilon}$ is essentially optimal; we show that if $n+m/n^{o(1)}$ registers may be used, query time $o(\log n)$ is possible.

[RS-95-25] *PostScript*, *DVI*.
Sren B. Lassen.
Basic Action Theory.
May 1995.
47 pp.

Abstract: Action semantics is a semantic description framework with very good pragmatic properties but until now a rather weak theory for reasoning about programs. A strong action theory would have a great practical potential, as it would facilitate reasoning about the large class of programming languages that can be described in action semantics.

This report develops the foundations for a richer action theory, by bringing together concepts and techniques from process theory and from work on operational reasoning about functional programs. Semantic preorders and equivalences in the action semantics setting are studied and useful operational techniques for establishing contextual equivalences are presented. These techniques are applied to establish equational and inequational action laws and an induction rule.

[RS-95-24] *PostScript*, *DVI*.
Peter rbk.
Can you Trust your Data?
April 1995.
15 pp. Appears in Mosses, Nielsen and Schwartzbach, editors, *Theory and Practice of Software Development: 6th International Conference Joint Conference CAAP/FASE*, TAPSOFT '95 Proceedings, LNCS 915, 1995, pages 575–590.

Abstract: A new program analysis is presented, and two compile time methods for this analysis are given. The analysis attempts to answer the question: "Given some trustworthy and some untrustworthy input, can we trust the value of a given variable after execution of some code". The analyses are based on an abstract interpretation framework and a constraint generation framework respectively. The analyses are proved safe with respect to an instrumented semantics. We explicitly deal with a language with pointers and possible aliasing problems. The constraint based analysis is related *directly* to the abstract interpretation and therefore indirectly to the instrumented semantics.

[RS-95-23] *PostScript*, *DVI*.
Allan Cheng and Mogens Nielsen.
Open Maps (at) Work.
April 1995.
33 pp. Appears with the title *Observing Behaviour Categorically* in Thiagarajan, editor, *Foundations of Software Technology and Theoretical Computer Science: 15th Conference*, FST&TCS '95 Proceedings, LNCS 1026, 1995, pages 263–278.

Abstract: The notion of bisimilarity, as defined by Park and Milner, has turned out to be one of the most fundamental notions of operational equivalences in the field of process algebras. Not only does it induce a congruence (largest bisimulation) in CCS which have nice equational properties, it has also proven itself applicable for numerous models of parallel computation and settings such as Petri Nets and semantics of functional languages. In an attempt to understand the relationships and differences between the extensive amount of research within the field, Joyal, Nielsen, and Winskel recently presented an abstract category-theoretic definition of bisimulation. They identify spans of morphisms satisfying certain "path lifting" properties, so-called open maps, as a possible abstract definition of bisimilarity. In their LICS '93 paper they show, that they can capture Park and Milner's bisimulation. The aim of this paper is to show that the abstract definition of bisimilarity is applicable "in practice" by showing how a representative selection of *well-known* bisimulations and equivalences, such as e.g. Hennessy's testing equivalence, Milner and Sangiorgi's barbed bisimulation, and Larsen and Skou's probabilistic bisimulation, are captured in the setting of open maps and hence, that the proposed notion of open maps seems successful. Hence, we confirm that the treatment of strong bisimulation in the LICS '93 paper is not a one-off application of open maps.

[RS-95-22] *PostScript*, *DVI*.
Anna Inglfsdttir.
A Semantic Theory for Value–Passing Processes, Late Approach, Part II: A Behavioural Semantics and Full Abstractness.
April 1995.
33 pp. To appear in *Information and Computation* (together with part I).

Abstract: A bisimulation based behavioural semantics, which reflects the late semantic approach, is given for CCS-like language. The denotational semantics given in the companion paper, part I, is shown to be fully abstract to a value-finitary version of the bisimulation preorder.

[RS-95-21] *PostScript*.
Jesper G. Henriksen, Ole J. L. Jensen, Michael E. Jrgensen, Nils Klarlund, Robert Paige, Theis Rauhe, and Anders B. Sandholm.
MONA: Monadic Second-Order Logic in Practice.
May 1995.
17 pp. Appears in Brinksma, Cleaveland, Larsen, Margaria and Stef-

fen, editors, *Tools and Algorithms for The Construction and Analysis of Systems: International Workshop*, TACAS '95 Selected Papers, LNCS 1019, 1995, pages 89–110.

Abstract: The purpose of this article is to introduce Monadic Second-order Logic as a practical means of specifying regularity. The logic is a highly succinct alternative to the use of regular expressions. We have built a tool Mona, which acts as a decision procedure and as a translator to finite-state machines. The tool is based on new algorithms for minimizing finite-state machines that use binary decision diagrams (BDD's) to represent transition functions in compressed form. A byproduct of this work is a new bottom-up algorithm to reduce BDD's in linear time without hashing.

The potential applications are numerous. We discuss text processing, Boolean circuits, and distributed systems.

Our main example is an automatic proof of properties for the "Dining Philosophers with Encyclopedia" example by Kurshan and MacMillan. We establish these properties for the parameterized case *without* the use of induction.

[RS-95-20] *PostScript*, *DVI*.
Anders Kock.
The Constructive Lift Monad.
March 1995.
18 pp.

Abstract: We study the lift monad on posets in the setting of intuitionistic set theory (meaning inside a topos). Unlike in the boolean situation, the lift monad here consists in more than just adding a bottom element freely. — We also study some related monads in the category of locales.

[RS-95-19] *PostScript*, *DVI*.
François Laroussinie and Kim G. Larsen.
Compositional Model Checking of Real Time Systems.
March 1995.
20 pp. Appears in Lee and Smolka, editors, *Concurrency Theory: 6th International Conference*, CONCUR '95 Proceedings, LNCS 962, 1995, pages 27–41.

Abstract: A major problem in applying model checking to finite–state systems is the potential combinatorial explosion of the state space arising from parallel composition. Solutions of this problem have been attempted for practical applications using a variety of techniques. Recent work by Andersen (Partial Model Checking. To appear in Proceedings of LICS '95) proposes a very promising compositional model checking technique, which has experimentally been shown to improve results obtained using Binary Decision Diagrams.

In this paper we make Andersen's technique applicable to systems described by networks of *timed automata*. We present a quotient construction, which allows timed automata components

to be gradually moved from the network expression into the specification. The intermediate specifications are kept small using minimization heuristics suggested by Andersen. The potential of the combined technique is demonstrated using a prototype implemented in CAML.

[RS-95-18] *PostScript*, *DVI*.
Allan Cheng.
Complexity Results for Model Checking.
February 1995.
18pp.

Abstract: The complexity of model checking branching and linear time temporal logics over Kripke structures has been addressed by Clarke, Emerson, and Sistla, among others. In terms of the size of the Kripke model and the length of the formula, they show that the model checking problem is solvable in polynomial time for CTL and NP-complete for $\mathsf{L}(F)$. The model checking problem can be generalised by allowing more succinct descriptions of systems than Kripke structures. We investigate the complexity of the model checking problem when the instances of the problem consist of a formula and a description of a system whose state space is at most exponentially larger than the description. Based on Turing machines, we define *compact systems* as a general formalisation of such system descriptions. Examples of such compact systems are K-bounded Petri nets and synchronised automata, and in these cases the well-known algorithms presented by Clarke, Emerson, and Sistla (1985,1986) would require exponential space in term of the sizes of the system descriptions and the formulas; we present polynomial space upper bounds for the model checking problem over compact systems and the logics CTL and $\mathsf{L}(X, U, S)$. As an example of an application of our general results we show that the model checking problems of both the branching time temporal logic CTL and the linear time temporal logics $\mathsf{L}(F)$ and $\mathsf{L}(X, U, S)$ over K-bounded Petri nets are PSPACE-complete.

[RS-95-17] *PostScript*, *DVI*.
Jari Koistinen, Nils Klarlund, and Michael I. Schwartzbach.
Design Architectures through Category Constraints.
February 1995.
19 pp.

Abstract: We provide a rigorous and concise formalism for specifying design architectures exterior to the design language. This allows several evolving architectural styles to be supported independently. Such architectural styles are specified in a tailored parse tree logic, which permits automatic support for conformance and consistency. We exemplify these ideas with a small design architecture inspired by real world constraints found in the Ericsson ATM Broadband System.

[RS-95-16] *PostScript*, *DVI*.
Dany Breslauer and Ramesh Hariharan.

Optimal Parallel Construction of Minimal Suffix and Factor Automata.
February 1995.
9 pp. Appears in *Parallel Processing Letters*, 6(1):35–44, 1996.

Abstract: This paper gives optimal parallel algorithms for the construction of the smallest deterministic finite automata recognizing all the suffixes and the factors of a string. The algorithms use recently discovered optimal parallel suffix tree construction algorithms together with data structures for the efficient manipulation of trees, exploiting the well known relation between suffix and factor automata and suffix trees.

[RS-95-15] *PostScript*, *DVI.*
Devdatt P. Dubhashi, Grammati E. Pantziou, Paul G. Spirakis, and Christos D. Zaroliagis.
The Fourth Moment in Luby's Distribution.
February 1995.
10 pp. Appears in *Theoretical Computer Science*, 148(1):133–140, 1995.

Abstract: Luby proposed a way to derandomize randomized computations which is based on the construction of a small probability space whose elements are 3-wise independent. In this paper we prove some new properties of Luby's space. More precisely, we analyze the fourth moment and prove an interesting technical property which helps to understand better Luby's distribution. As an application, we study the behavior of random edge cuts in a weighted graph.

[RS-95-14] *PostScript*, *DVI.*
Devdatt P. Dubhashi.
Inclusion–Exclusion(3) *Implies Inclusion–Exclusion*(n).
February 1995.
6 pp.

Abstract: We consider a natural generalisation of the familiar inclusion–exclusion formula for sets in the setting of ranked lattices. We show that the generalised inclusion–exclusion formula holds in a lattice if and only if the lattice is distributive and the rank function is modular. As a consequence it turns out (perhaps surprisingly) that the inclusion–exclusion formula for three elements implies the inclusion–exclusion formula for an arbitrary number of elements.

[RS-95-13] *PostScript*, *DVI.*
Torben Braner.
The Girard Translation Extended with Recursion.
February 1995.
79 pp. Full version of paper appearing in Pacholski and Tiuryn, editors, *European Association for Computer Science Logic: 8th Workshop*, CSL '94 Selected Papers, LNCS 933, 1995, pages 31–45.

Abstract: This paper extends Curry-Howard interpretations of Intuitionistic Logic (IL) and Intuitionistic Linear Logic (ILL)

with rules for recursion. The resulting term languages, the λ^{rec}-calculus and the linear λ^{rec}-calculus respectively, are given sound categorical interpretations. The embedding of proofs of IL into proofs of ILL given by the Girard Translation is extended with the rules for recursion, such that an embedding of terms of the λ^{rec}-calculus into terms of the linear λ^{rec}-calculus is induced via the extended Curry-Howard isomorphisms. This embedding is shown to be sound with respect to the categorical interpretations.

[RS-95-12] *PostScript*.
Gerth Stlting Brodal.
Fast Meldable Priority Queues.
February 1995.
12 pp. Appears in Akl, Dehne, Sack and Santoro, editors, *Algorithms and Data Structures: 4th Workshop*, WADS '95 Proceedings, LNCS 955, 1995, pages 282–290.

Abstract: We present priority queues that support the operations MakeQueue, FindMin, Insert and Meld in worst case time O(1) and Delete and DeleteMin in worst case time $O(\log n)$. They can be implemented on the pointer machine and require linear space. The time bounds are optimal for all implementations where Meld takes worst case time $o(n)$.

To our knowledge this is the first priority queue implementation that supports Meld in worst case constant time and DeleteMin in logarithmic time.

[RS-95-11] *PostScript, DVI*.
Alberto Apostolico and Dany Breslauer.
An Optimal $O(\log\log n)$ Time Parallel Algorithm for Detecting all Squares in a String.
February 1995.
18 pp. Appears in *SIAM Journal on Computing*, 25(6):1318–1331, December, 1996.

Abstract: An optimal $O(\log\log n)$ time concurrent-read concurrent-write parallel algorithm for detecting all squares in a string is presented. A tight lower bound shows that over general alphabets this is the fastest possible optimal algorithm. When p processors are available the bounds become $\Theta(\lceil\frac{n\log n}{p}\rceil + \log\log_{\lceil 1+p/n\rceil} 2p)$. The algorithm uses an optimal parallel string-matching algorithm together with periodicity properties to locate the squares within the input string.

[RS-95-10] *PostScript, DVI*.
Dany Breslauer and Devdatt P. Dubhashi.
Transforming Comparison Model Lower Bounds to the Parallel-Random-Access-Machine.
February 1995.
11 pp. Appears in *Fifth Italian Conference on Theoretical Computer Science*, November 1995 and in *Information Processing Letters*, 62(2):103–110, April 1997.

Abstract: This note provides general transformations of lower bounds in Valiant's parallel comparison decision tree model to lower bounds in the priority concurrent-read concurrent-write parallel-random-access-machine model. The proofs rely on standard Ramsey–theoretic arguments that simplify the structure of the computation by restricting the input domain. The transformation of comparison model lower bounds, which are usually easier to obtain, to the parallel-random-access-machine, unifies some known lower bounds and gives new lower bounds for several problems.

[RS-95-9] *PostScript*, *DVI*.
Lars Ramkilde Knudsen.
Partial and Higher Order Differentials and Applications to the DES.
February 1995.
24 pp.

Abstract: In 1994 Lai considered higher order derivatives of discrete functions and introduced the concept of higher order differentials. We introduce the concept of partial differentials and present attacks on ciphers presumably secure against differential attacks, but vulnerable to attacks using higher order and partial differentials. Also we examine the DES for partial and higher order differentials and give a differential attack using partial differentials on DES reduced to 6 rounds using only 46 chosen plaintexts with an expected running time of about the time of 3,500 encryptions. Finally it is shown how to find a minimum nonlinear order of a block cipher using higher order differentials.

[RS-95-8] *PostScript*, *DVI*.
Ole I. Hougaard, Michael I. Schwartzbach, and Hosein Askari.
Type Inference of Turbo Pascal.
February 1995.
19 pp. Appeas in *Software—Consepts & Tools*, 16:160–169, Springer-Verlag, 1995.

Abstract: Type inference is generally thought of as being an exclusive property of the functional programming paradigm. We argue that such a feature may be of significant benefit for also standard imperative languages. We present a working tool (available by WWW) providing these benefits for a full version of Turbo Pascal. It has the form of a preprocessor that analyzes programs in which the type annotations are only partial or even absent. The resulting program has full type annotations, will be accepted by the standard Turbo Pascal compiler, and has polymorphic use of procedures resolved by means of code expansion.

[RS-95-7] *PostScript*.
David A. Basin and Nils Klarlund.
Hardware Verification using Monadic Second-Order Logic.
January 1995.
13 pp. Appears in Wolper, editor, *Computer-Aided Verification: 7th International Conference*, CAV '95 Proceedings, LNCS 939, 1995, pages 31–41.

Abstract: We show how the second-order monadic theory of strings can be used to specify hardware components and their behavior. This logic admits a decision procedure and counter-model generator based on canonical automata for formulas. We have used a system implementing these concepts to verify, or find errors in, a number of circuits proposed in the literature. The techniques we use make it easier to identify regularity in circuits, including those that are parameterized or have parameterized behavioral specifications. Our proofs are semantic and do not require lemmas or induction as would be needed when employing a conventional theory of strings as a recursive data type.

[RS-95-6] *PostScript*, *DVI*.
Igor Walukiewicz.
A Complete Deductive System for the μ-Calculus.
January 1995.
39 pp . Appears in *Information and Computation*, 157:142–182, 2000. Appeared earlier in Kozen, editor, *Tenth Annual IEEE Symposium on Logic in Computer Science*, LICS '95 Proceedings, 1995, pages 14–24.

Abstract: The propositional μ-calculus as introduced by Kozen is considered. In that paper a finitary axiomatisation of the logic was presented but its completeness remained an open question. Here a different finitary axiomatisation of the logic is proposed and proved to be complete. The two axiomatisations are compared.

[RS-95-5] *PostScript*, *DVI*.
Luca Aceto and Anna Inglfsdttir.
A Complete Equational Axiomatization for Prefix Iteration with Silent Steps.
January 1995.
27 pp. Appears in Wirsing and Nivat, editors, *Algebraic Methodology and Software Technology: 5th International Conference*, AMAST '96 Proceedings, LNCS 1101, 1996, pages 195–209 under the title *An Equational Axiomatization of Observation Congruence for Prefix Iteration.*

Abstract: Fokkink ((1994) *Inf. Process. Lett.* **52**: 333–337) has recently proposed a complete equational axiomatization of strong bisimulation equivalence for the language obtained by extending Milner's basic CCS with prefix iteration. Prefix iteration is a variation on the original binary version of the Kleene star operation p^*q obtained by restricting the first argument to be an atomic action. In this paper, we extend Fokkink's results to a setting with the unobservable action τ by giving a complete equational axiomatization of Milner's observation congruence over Fokkink's language. The axiomatization is obtained by extending Fokkink's axiom system with two of Milner's standard τ-laws and the following three equations that describe the interplay between the silent nature of τ and prefix iteration:

$$\tau.x \quad = \quad \tau^* x$$

$$
\begin{aligned}
a^*(x + \tau.y) &= a^*(x + \tau.y + a.y) \\
\tau.(a^*x) &= a^*(\tau.a^*x) \ .
\end{aligned}
$$

Using a technique due to Groote, we also show that the resulting axiomatization is ω-complete, i.e., complete for equality of open terms.

[RS-95-4] *PostScript*, *DVI*.
Mogens Nielsen and Glynn Winskel.
Petri Nets and Bisimulations.
January 1995.
36 pp. Appears in *Theoretical Computer Science* 153(1–2):211–244, January 1996.
Abstract: Several categorical relationships (adjunctions) between models for concurrency have been established, allowing the translation of concepts and properties from one model to another. A central example is a coreflection between Petri nets and asynchronous transition systems. The purpose of the present paper is to illustrate the use of such relationships by transferring to Petri nets a general concept of bisimulation.

[RS-95-3] *PostScript*, *DVI*.
Anna Inglfsdttir.
A Semantic Theory for Value–Passing Processes, Late Approach, Part I: A Denotational Model and Its Complete Axiomatization.
January 1995.
37 pp. To appear in *Information and Comutation* (together with part II).
Abstract: A general class of languages and denotational models for value-passing calculi based on the late semantic approach is defined. A concrete instantiation of the general syntax is given. This is a modification of the standard *CCS* according to the late approach. A denotational model for the concrete language is given, an instantiation of the general class. An equationally based proof system is defined and shown to be sound and complete with respect to the model.

[RS-95-2] *PostScript*, *DVI*.
François Laroussinie, Kim G. Larsen, and Carsten Weise.
From Timed Automata to Logic - and Back.
January 1995.
21 pp. Appears in Wiedermann and Hjek, editors, *Mathematical Foundations of Computer Science: 20th International Symposium*, MFCS '95 Proceedings, LNCS 969, 1995, pages 529–539.
Abstract: One of the most successful techniques for automatic verification is that of *model checking*. For finite automata there exist since long extremely efficient model–checking algorithms, and in the last few years these algorithms have been made applicable to the verification of real–time automata using the region–techniques of Alur and Dill. In this paper, we continue this transfer of existing techniques from the setting of finite (untimed) automata to that of timed automata. In particular, a timed logic

L_ν is put forward, which is sufficiently expressive that we for any timed automaton may construct a single *characteristic* L_ν formula uniquely characterizing the automaton up to timed bisimilarity. Also, we prove decidability of the *satisfiability* problem for L_ν with respect to given bounds on the number of clocks and constants of the timed automata to be constructed. None of these results have as yet been succesfully accounted for in the presence of time (An exception occurs in Alur's thesis in which a decidability result is presented for a *linear* timed logic called MITL).

[RS-95-1] *PostScript*, *DVI*.
Gudmund Skovbjerg Frandsen, Thore Husfeldt, Peter Bro Miltersen, Theis Rauhe, and Sren Skyum.
Dynamic Algorithms for the Dyck Languages.
January 1995.
21 pp. Appears in Akl, Dehne, Sack and Santoro, editors, *Algorithms and Data Structures: 4th Workshop*, WADS '95 Proceedings, LNCS 955, 1995, pages 98–108.

Abstract: We study dynamic membership problems for the Dyck languages, the class of strings of properly balanced parentheses. We also study the Dynamic Word problem for the free group. We present deterministic algorithms and data structures which maintain a string under replacements of symbols, insertions, and deletions of symbols, and language membership queries. Updates and queries are handled in polylogarithmic time. We also give both Las Vegas- and Monte Carlo-type randomised algorithms to achieve better running times, and present lower bounds on the complexity for variants of the problems.