BRICS Research Series, Abstracts, 2007

June 27, 2008

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

References

[RS-07-18] *PostScript*, *PDF*, *DVI*.
Jan Midtgaard.
Control-Flow Analysis of Functional Programs.
December 2007.
iii+38 pp.
Abstract: We present a survey of control-flow analysis of functional programs, which has been the subject of extensive investigation throughout the past 25 years. Analyses of the control flow of functional programs have been formulated in multiple settings and have led to many different approximations, starting with the seminal works of Jones, Shivers, and Sestoft. In this paper we survey control-flow analysis of functional programs by structuring the multitude of formulations and approximations and comparing them.

[RS-07-17] Luca Aceto, Willem Jan Fokkink, and Anna Ingólfsdóttir.
A Cancellation Theorem for 7BCCSP.
December 2007.
30 pp.
Abstract: This paper presents a cancellation theorem for the preorders in van Glabbeek's linear time-branching time spectrum over BCCSP. Apart from having some intrinsic interest, the proven cancellation result plays a crucial role in the study of the cover equations, in the sense of Fokkink and Nain, that characterize the studied semantics. The techniques used in the proof of the cancellation theorem may also have some independent interest.

[RS-07-16] *PostScript*, *PDF*, *DVI*.
Olivier Danvy and Kevin Millikin.
On the Equivalence between Small-Step and Big-Step Abstract Machines: A Simple Application of Lightweight Fusion.
November 2007.
ii+11 pp. To appear in *Information Processing Letters* (extended version). Supersedes BRICS RS-07-8.

Abstract: We show how Ohori and Sasano's recent lightweight fusion by fixed-point promotion provides a simple way to prove the equivalence of the two standard styles of specification of abstract machines: (1) in small-step form, as a state-transition function together with a 'driver loop,' i.e., a function implementing the iteration of this transition function; and (2) in big-step form, as a tail-recursive function that directly maps a given configuration to a final state, if any. The equivalence hinges on our observation that for abstract machines, fusing a small-step specification yields a big-step specification. We illustrate this observation here with a recognizer for Dyck words, the CEK machine, and Krivine's machine with call/cc.

The need for such a simple proof is motivated by our current work on small-step abstract machines as obtained by refocusing a function implementing a reduction semantics (a syntactic correspondence), and big-step abstract machines as obtained by CPS-transforming and then defunctionalizing a function implementing a big-step semantics (a functional correspondence).

[RS-07-15] Jooyong Lee.
A Case for Dynamic Reverse-code Generation.
August 2007.
ii+10 pp.

Abstract: Backtracking (i.e. reverse execution) helps the user of a debugger to naturally think backwards along the execution path of a program, and thinking backwards makes it easy to locate the origin of a bug. So far backtracking has been implemented mostly by state saving or by checkpointing. These implementations, however, inherently do not scale. As has often been said, the ultimate solution for backtracking is to use reverse code: executing the reverse code restores the previous states of a program. In our earlier work, we presented a method to generate reverse code on the fly while running a debugger. This article presents a case study of dynamic reverse-code generation. We compare the memory usage of various backtracking methods in a simple but nontrivial example, a bounded-buffer program. In the case of non-deterministic programs such as this bounded-buffer program, our dynamic reverse-code generation can outperform the existing backtracking methods in terms of memory efficiency.

[RS-07-14] *PostScript*, *PDF*, *DVI*.
Olivier Danvy and Michael Spivey.
On Barron and Strachey's Cartesian Product Function.
July 2007.
ii+14 pp.

Abstract: Over forty years ago, David Barron and Christopher Strachey published a startlingly elegant program for the Cartesian product of a list of lists, expressing it with a three nested occurrences of the function we now call *foldr*. This program is remarkable for its time because of its masterful display of higher-order functions and lexical scope, and we put it forward as possi-

bly the first ever functional pearl. We first characterize it as the result of a sequence of program transformations, and then apply similar transformations to a program for the classical power set example. We also show that using a higher-order representation of lists allows a definition of the Cartesian product function where *foldr* is nested only twice.

[RS-07-13] Martin Lange.
Temporal Logics Beyond Regularity.
July 2007.
82 pp.
Abstract: Non-regular program correctness properties play an important role in the specification of unbounded buffers, recursive procedures, etc. This thesis surveys results about the relative expressive power and complexity of temporal logics which are capable of defining non-regular properties. In particular, it features Propositional Dynamic Logic of Context-Free Programs, Fixpoint Logic with Chop, the Modal Iteration Calculus, and Higher-Order Fixpoint Logic.

Regarding expressive power we consider two classes of structures: arbitrary transition systems as well as finite words as a subclass of the former. The latter is meant to give an intuitive account of the logics' expressive powers by relating them to known language classes defined in terms of grammars or Turing Machines.

Regarding the computational complexity of temporal logics beyond regularity we focus on their model checking problems since their satisfiability problems are all highly undecidable. Their model checking complexities range between polynomial time and non-elementary.

[RS-07-12] Gerth Stølting Brodal, Rolf Fagerberg, Allan Grønlund Jørgensen, Gabriel Moruz, and Thomas Mølhave.
Optimal Resilient Dynamic Dictionaries.
July 2007.

[RS-07-11] *PostScript*, *PDF*, *DVI*.
Luca Aceto and Anna Ingólfsdóttir.
The Saga of the Axiomatization of Parallel Composition.
June 2007.
15 pp. To appear in the Proceedings of CONCUR 2007, the 18th International Conference on Concurrency Theory (Lisbon, Portugal, September 4–7, 2007), Lecture Notes in Computer Science, Springer-Verlag, 2007.
Abstract: This paper surveys some classic and recent results on the finite axiomatizability of bisimilarity over CCS-like languages. It focuses, in particular, on non-finite axiomatizability results stemming from the semantic interplay between parallel composition and nondeterministic choice. The paper also highlights the role that auxiliary operators, such as Bergstra and Klop's left and communication merge and Hennessy's merge operator, play in the search for a finite, equational axiomatization of parallel com-

position both for classic process algebras and for their real-time extensions.

[RS-07-10] *PostScript*, *PDF*.
Claus Brabrand, Robert Giegerich, and Anders Møller.
Analyzing Ambiguity of Context-Free Grammars.
May 2007.
17 pp. Full version of paper presented at CIAA '07.
Abstract: It has been known since 1962 that the ambiguity problem for context-free grammars is undecidable. Ambiguity in context-free grammars is a recurring problem in language design and parser generation, as well as in applications where grammars are used as models of real-world physical structures.

We observe that there is a simple linguistic characterization of the grammar ambiguity problem, and we show how to exploit this to conservatively approximate the problem based on local regular approximations and grammar unfoldings. As an application, we consider grammars that occur in RNA analysis in bioinformatics, and we demonstrate that our static analysis of context-free grammars is sufficiently precise and efficient to be practically useful.

[RS-07-9] Janus Dam Nielsen and Michael I. Schwartzbach.
The SMCL Language Specification.
March 2007.

[RS-07-8] *PostScript*, *PDF*, *DVI*.
Olivier Danvy and Kevin Millikin.
A Simple Application of Lightweight Fusion to Proving the Equivalence of Abstract Machines.
March 2007.
ii+6 pp.
Abstract: We show how Ohori and Sasano's recent lightweight fusion by fixed-point promotion provides a simple way to prove the equivalence of the two standard styles of specification of abstract machines: (1) as a transition function together with a 'driver loop' implementing the iteration of this transition function; and (2) as a function directly iterating upon a configuration until reaching a final state, if ever. The equivalence hinges on the fact that the latter style of specification is a fused version of the former one. The need for such a simple proof is motivated by our recent work on syntactic correspondences between reduction semantics and abstract machines, using refocusing.

[RS-07-7] *PostScript*, *PDF*, *DVI*.
Olivier Danvy and Kevin Millikin.
Refunctionalization at Work.
March 2007.
ii+16 pp. Invited talk at the 8th International Conference on Mathematics of Program Construction, MPC '06.
Abstract: We present the left inverse of Reynolds's defunctionalization and we show its relevance to programming and to pro-

gramming languages. We present two methods to put a program that is almost in defunctionalized form into one that is actually in defunctionalized form, and we illustrate them with a recognizer for Dyck words and with Dijkstra's shunting-yard algorithm.

[RS-07-6] *PostScript*, *PDF*, *DVI*.
Olivier Danvy, Kevin Millikin, and Lasse R. Nielsen.
On One-Pass CPS Transformations.
March 2007.
ii+19 pp. Theoretical Pearl to appear in the *Journal of Functional Programming*. Revised version of BRICS RS-02-3.

Abstract: We bridge two distinct approaches to one-pass CPS transformations, i.e., CPS transformations that reduce administrative redexes at transformation time instead of in a post-processing phase. One approach is compositional and higher-order, and is independently due to Appel, Danvy and Filinski, and Wand, building on Plotkin's seminal work. The other is non-compositional and based on a reduction semantics for the lambda-calculus, and is due to Sabry and Felleisen. To relate the two approaches, we use three tools: Reynolds's defunctionalization and its left inverse, refunctionalization; a special case of fold-unfold fusion due to Ohori and Sasano, fixed-point promotion; and an implementation technique for reduction semantics due to Danvy and Nielsen, refocusing.

This work is directly applicable to transforming programs into monadic normal form.

[RS-07-5] *PostScript*, *PDF*, *DVI*.
Luca Aceto, Silvio Capobianco, and Anna Ingólfsdóttir.
On the Existence of a Finite Base for Complete Trace Equivalence over BPA with Interrupt.
February 2007.
26 pp.

Abstract: We study Basic Process Algebra with interrupt modulo complete trace equivalence. We show that, unlike in the setting of the more demanding bisimilarity, a ground complete finite axiomatization exists. We explicitly give such an axiomatization, and extend it to a finite complete one in the special case when a single action is present.

[RS-07-4] *PostScript*, *PDF*, *DVI*.
Kristian Støvring and Søren B. Lassen.
A Complete, Co-Inductive Syntactic Theory of Sequential Control and State.
February 2007.
36 pp. Appears in the proceedings of POPL 2007, p. 161–172.

Abstract: We present a new co-inductive syntactic theory, eager normal form bisimilarity, for the untyped call-by-value lambda calculus extended with continuations and mutable references.

We demonstrate that the associated bisimulation proof principle is easy to use and that it is a powerful tool for proving equiva-

lences between recursive imperative higher-order programs.

The theory is modular in the sense that eager normal form bisimilarity for each of the calculi extended with continuations and/or mutable references is a fully abstract extension of eager normal form bisimilarity for its sub-calculi. For each calculus, we prove that eager normal form bisimilarity is a congruence and is sound with respect to contextual equivalence. Furthermore, for the calculus with both continuations and mutable references, we show that eager normal form bisimilarity is complete: it coincides with contextual equivalence.

[RS-07-3] *PostScript*, *PDF*.
Luca Aceto, Willem Jan Fokkink, and Anna Ingólfsdóttir.
Ready To Preorder: Get Your BCCSP Axiomatization for Free!
February 2007.
37 pp.

Abstract: This paper contributes to the study of the equational theory of the semantics in van Glabbeek's linear time - branching time spectrum over the language BCCSP, a basic process algebra for the description of finite synchronization trees. It offers an algorithm for producing a complete (respectively, ground-complete) equational axiomatization of a behavioral congruence lying between ready simulation equivalence and partial traces equivalence from a complete (respectively, ground-complete) inequational axiomatization of its underlying precongruence—that is, of the precongruence whose kernel is the equivalence. The algorithm preserves finiteness of the axiomatization when the set of actions is finite. It follows that each equivalence in the spectrum whose discriminating power lies in between that of ready simulation and partial traces equivalence is finitely axiomatizable over the language BCCSP if so is its defining preorder.

[RS-07-2] *PostScript*, *PDF*, *DVI*.
Luca Aceto and Anna Ingólfsdóttir.
Characteristic Formulae: From Automata to Logic.
January 2007.
18 pp.

Abstract: This paper discusses the classic notion of characteristic formulae for processes using variations on Hennessy-Milner logic as the underlying logical specification language. It is shown how to characterize logically (states of) finite labelled transition systems modulo bisimilarity using a single formula in Hennessy-Milner logic with recursion. Moreover, characteristic formulae for timed automata with respect to timed bisimilarity and the faster-than preorder of Moller and Tofts are offered in terms of the logic L_ν of Laroussinie, Larsen and Weise.

[RS-07-1] *PostScript*, *PDF*.
Daniel Andersson.
HIROIMONO is NP-complete.
January 2007.
8 pp.

Abstract: In a Hiroimono puzzle, one must collect a set of stones from a square grid, moving along grid lines, picking up stones as one encounters them, and changing direction only when one picks up a stone. We show that deciding the solvability of such puzzles is NP-complete.