

W. R. Cleaveland (Ed.)

Cc 01-630

CONCUR '92

Third International Conference
on Concurrency Theory
Stony Brook, NY, USA, August 24-27, 1992
Proceedings

Springer-Verlag

Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

BIBLIOTHEQUE DU CERIST

Series Editors

Gerhard Goos
Universität Karlsruhe
Postfach 69 80
Vincenz-Priessnitz-Straße 1
W-7500 Karlsruhe, FRG

Juris Harmanis
Department of Computer Science
Cornell University
5149 Upson Hall
Ithaca, NY 14853, USA

Volume Editor

W. Rance Cleaveland
Department of Computer Science, North Carolina State University
Raleigh, NC 27695-8206, USA

6254

CR Subject Classification (1991): F.3.2, F.3.1, F.3.3, D.3.2, F.1.2

ISBN 3-540-55822-5 Springer-Verlag Berlin Heidelberg New York

ISBN 0-387-55822-5 Springer-Verlag New York Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1992
Printed in Germany

Typesetting: Camera ready by author/editor
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

Preface

CONCUR '92 is the third in what is becoming an annual series of conferences devoted to the study of theories of concurrency. The first two CONCURRENCES were held in Amsterdam, the Netherlands, in 1990 and 1991; 1992 marks the first year that the conference has been held outside of Europe. The goal of CONCUR is to provide a forum for reporting on advances in theories of concurrency and their applications and to facilitate exchanges of ideas and information among the several different schools of concurrency theory that have arisen over the past 20 years.

This proceedings contains 34 papers that were culled from 112 submissions, 41 more than last year. Five invited papers are also included. The number of submissions substantially exceeded expectations, and I would like especially to thank the members of the program committee, and their subreferees, for their efforts in assembling this collection of papers.

Support for CONCUR '92 has generously been provided by the National Science Foundation (NSF grant CCR-9201450) and ESPRIT. The organizers of the conference would like to thank Nat Macon of NSF and Michel Bosco of ESPRIT for serving as the cognizant officers of these awards.

I would also like to thank the State University of New York at Stony Brook for hosting CONCUR '92 and the organizing committee for their tireless efforts in arranging the conference.

Raleigh, June 1992

Rance Cleaveland

Program Committee

R. Cleaveland (N.C. State University), *Chair*

F. de Boer (TU Eindhoven)
B. Bloom (Cornell)
S. Brookes (CMU)
N. Francez (Technion)
U. Goltz (GMD)
J. Gunawardena (Hewlett-Packard)
L. Logrippo (University of Ottawa)

M. Merritt (AT&T Bell Labs)
J. Sifakis (Université de Grenoble)
P. Sistla (University of Illinois-Chicago)
E. Stark (SUNY Stony Brook)
B. Steffen (RWTH-Aachen)
P. Wolper (Université de Liège)

Organizing Committee

S. Smolka, *Chair*
R. Cleaveland, *Program Chair*
K. Germana
F. Moller, *Tutorials Chair*
A. Zwarico, *Publicity Chair*

Steering Committee

J.C.M. Baeten, *Chair*
E. Best
K.G. Larsen
U. Montanari
P. Wolper

BIBLIOTHEQUE DU CERIST

Table of Contents

Invited Lecture: The Polyadic π -Calculus

<i>R. Milner (U. Edinburgh)</i>	1
---------------------------------------	---

Session: Semantics I

Testing Equivalences for Mobile Processes

<i>M. Boreale, R. De Nicola (U. Roma)</i>	2
---	---

Testing Equivalence for Petri Nets with Action Refinement

<i>L. Jategaonkar, A. Meyer (MIT, Cambridge, MA)</i>	17
--	----

Session: Proof Techniques

The Problem of "Weak Bisimulation up to"

<i>D. Sangiorgi, R. Milner (U. Edinburgh)</i>	32
---	----

On the Uniqueness of Fixpoints Modulo Observation Congruence

<i>E. Brinksma (U. Twente)</i>	47
--------------------------------------	----

Session: Modular Analysis

Verification of Parallel Systems via Decomposition

<i>J.F. Groote (CWI, Amsterdam), F. Moller (U. Edinburgh)</i>	62
---	----

Interface Refinement in Reactive Systems

<i>R. Gerth, R. Kuiper, J. Segers (T.U. Eindhoven)</i>	77
--	----

Session: True Concurrency I

Concurrent Testing of Processes

<i>M. Hennessy (U. Sussex)</i>	94
--------------------------------------	----

A Theory of Processes with Localities

<i>G. Boudol, I. Castellani (INRIA, Sophia-Antipolis), M. Hennessy (U. Sussex), A. Kiehn (T.U. Munich)</i>	108
--	-----

Session: Decision Problems**Model Checking for Context-Free Processes***O. Burkart, B. Steffen (RWTH Aachen)*..... 123**Bisimulation Equivalence Is Decidable for All Context-Free Processes***S. Christensen (U. Edinburgh), H. Hüttel (U. Aalborg),
Colin Stirling (U. Edinburgh)*..... 138**Distributed Bisimilarity Is Decidable for a Class of
Infinite State-Space Processes***S. Christensen (U. Edinburgh)*..... 148**Invited Talk: How Vital Is Liveness?****Verifying Timing Properties of Reactive and Hybrid Systems***A. Pnueli (Weizmann Institute, Rehovot)*..... 162**Session: Temporal Logic I****Preserving Specific Properties in Program Development***F.A. Stomp (U. Kiel)*..... 176**Sometimes "Some" Is as Good as "All"***D. Peled (U. Warwick)*..... 192**Session: Temporal Logic II****The Weakest Compositional Semantic Equivalence Preserving****Nexttime-Less Linear Temporal Logic***R. Kaivola (U. Helsinki), A. Valmari (Technical Research Center Finland)*... 207**Propositional Temporal Logics and Equivalences***U. Goltz (GMD, Bonn), R. Kuiper (T.U. Eindhoven),
W. Penczek (Polish Academy of Science, Warsaw)*..... 222**Invited Talk: The Duality of Time and Information***V.R. Pratt (Stanford U.)*..... 237**Session: True Concurrency II****Homology of Higher Dimensional Automata***E. Goubault (Ecole Normale Supérieure, Paris),
T.P. Jensen (Imperial College, London)*..... 254**Posets for Configurations!***A. Rensink (U. Twente)*..... 269

Session: True Concurrency III

On the Semantics of Petri Nets

J. Meseguer (SRI, Menlo Park), U. Montanari, V. Sassone (U. Pisa)..... 286

Structural Operational Specifications and Trace Automata

E. Badouel, P. Darondeau (IRISA, Rennes)..... 302

Invited Talk: At-Most-Once Message Delivery: A Case Study in Algorithm Verification

N. Lynch (MIT, Cambridge, MA) together with

B. Lampson (DEC, Palo Alto),

J. Sogaard-Andersen (MIT, Cambridge, MA)..... 317

Session: Automata

Games I/O Automata Play

N. Reingold (Yale U., New Haven), D.-W. Wang (U. Delaware, Newark),

L.D. Zuck (Yale U., New Haven)..... 325

Minimization of Timed Transition Systems

R. Alur (AT&T, Murray Hill), C. Courcoubetis (U. Crete),

N. Halbwachs (U. Grenoble), D. Dill, H. Wong-Toi (Stanford U.)..... 340

Session: Applications

Using CSP to Verify a Timed Protocol over a Fair Medium

J. Davies, S. Schneider (Oxford U.)..... 355

Timed Ethernet: Real-Time Formal Specification of Ethernet

H.B. Weinberg, L.D. Zuck (Yale U., New Haven)..... 370

Implementing LOTOS Specifications by Communicating State Machines

G. Karjoth (IBM Zurich)..... 386

Invited Talk: Discrete Time Process Algebra

J.C.M. Baeten (T.U. Eindhoven),

J.A. Bergstra (U. Amsterdam and Utrecht U.)..... 401

Session: Real-Time

The Silent Step in Time

S. Klusener (CWI, Amsterdam)..... 421

Action Transducers and Timed Automata

F. Vaandrager (CWI, Amsterdam), N. Lynch (MIT, Cambridge, MA)..... 436

Session: Probabilistic Processes

Compositional Verification of Probabilistic Processes

K.G. Larsen, A. Skou (U. Aalborg)..... 456

Axiomatizing Probabilistic Processes: ACP with Generative Probabilities

*J.C.M. Baeten (T.U. Eindhoven), J.A. Bergstra (U. Amsterdam),
S.A. Smolka (SUNY Stony Brook)*..... 472**Session: Trace Semantics**

Embeddings Among Concurrent Programming Languages

E. Shapiro (Weizmann Institute, Rehovot)..... 486

Logic of Trace Languages

A. Rabinovich (IBM Yorktown Heights)..... 504**Session: Synchronization**

Multiway Synchronization Verified with Coupled Simulation

J. Parrow (SICS, Kista), P. Sjödin (Uppsala U.)..... 518

Programming in a General Model of Synchronization

S.M. German (GTE, Waltham)..... 534**Session: Semantics II**

Operational and Compositional Semantics of Synchronous Automaton

Compositions

F. Maraninchi (U. Grenoble)..... 550

Towards a Theory of Actor Computation

*G. Agha (U. Illinois, Urbana), I.A. Mason (Stanford U.),
S. Smith (Johns Hopkins U., Baltimore), C. Talcott (Stanford U.)*..... 565

Authors Index..... 580

The Polyadic π -calculus

Robin Milner

Laboratory for the Foundations of Computer Science
Department of Computer Science
The King's Buildings
University of Edinburgh
Edinburgh EH9 3JZ
U.K.

The π -calculus is a calculus of concurrent processes based upon the idea of *naming*. It models dynamically changing concurrent systems, has a rich algebraic theory, and contains in a precise way both functions (the λ -calculus) and data structures, represented as processes. The calculus is a generalization of CCS, and was introduced by Robin Milner, Joachim Parrow and David Walker, based on important ideas of Mogens Nielsen and Uffe Engberg.

The way in which everything is built upon naming is this: When two processes interact, they *use* a name (which can be thought of as a channel). This name is called the *subject* of the interaction. The *object* of an interaction – its information content – is also a name; this is the *mention* of a name, not the *use* of it. To receive a name is to acquire the ability to use it, perhaps to interact with a process which was previously inaccessible. This process may indeed represent a datum, as explained above; then one can think of the datum itself having been received.

In this lecture I shall discuss the polyadic version of the π -calculus, which supports a very fruitful notion of *sort* and *sorting*, akin to simple typing in the λ -calculus. It will be seen how different sortings are appropriate for different applications. The encoding of the λ -calculus into π -calculus, and the uniform representation of data structures, are best seen in the polyadic setting with suitable sortings.

Testing Equivalence for Mobile Processes

(Extended Abstract)

Michele Boreale and Rocco De Nicola

Università di Roma "La Sapienza"
Dipartimento di Scienze dell'Informazione
Via Salaria 113, I-00198 Roma (Italy)

Abstract. The impact of applying the testing approach to a calculus of processes with a dynamically changing structure is investigated. A proof system for the finite fragment of the calculus is introduced which consists of two groups of laws: those for strong observational equivalence and those needed to deal with τ actions. Soundness and completeness w.r.t. a testing preorder are shown. A fully abstract denotational model for the language is presented which relies on the existence of normal forms for processes.

1. Introduction

Process Algebras ([Mil89], [Hoa85], [BK89]) are generally recognized as a good formalism for describing and studying properties of distributed concurrent systems. A process algebra is often defined by specifying its syntax and the transitional semantics of its terms by means of Structured Operational Semantics [Plo81]. By now, this approach has become a standard tool for specifying basic semantics of process algebras, but it was early recognized that it does not yield extensional accounts of processes. Thus, techniques have been developed to abstract from unwanted details in systems descriptions. Many of these techniques are based on behavioural equivalences; two terms are identified if and only if no observer can notice any difference between their external behaviours.

Process description languages, such as CCS, have been (and are) thoroughly studied using equivalence notions based on bisimulations (see e.g. [Mil89]) or on testing (see e.g. [Hen88]). Complete axiomatizations have been put forward which are of fundamental importance for manipulating process expressions by means of simple axioms and inference rules and constitute the theoretical basis for a class of verification tools (see e.g. [DIN90, Hui92]).

Almost all process algebras which have been considered permit describing only systems whose subparts can interact by performing *pure synchronizations*. Lately, a language with value-passing has been investigated and a complete axiomatization of a testing-based equivalence has been performed by provided for it [HI91]. A further step toward improving the descriptive power of process algebras has been performed by adding