W. R. Cleaveland (Ed.)

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

Series Editors

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

Juris Hartmanis
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

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

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 CONCURs 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 collectio 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), *Chai*

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

Table of Contents

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.

This article was processed using the LaTeX macro package with LLNCS style

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

---

Work partially supported by "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo" of CNR, contract n. 91.00894.69 and by Istituto di Elaborazione dell'Informazione of CNR, Pisa.