

# Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

92

---

Robin Milner

A Calculus of  
Communicating Systems

---



Springer-Verlag  
Berlin Heidelberg New York 1980

## **Editorial Board**

W. Brauer P. Brinch Hansen D. Gries C. Moler G. Seegmüller  
J. Stoer N. Wirth

## **Author**

Robin Milner  
University of Edinburgh  
Dept. of Computer Science  
James Clerk Maxwell Building  
The King's Buildings  
Mayfield Road  
Edinburgh EH9 3JZ  
Great Britain

AMS Subject Classifications (1979): 68-02

CR Subject Classifications (1974): 4.30, 5.20, 5.22, 5.24

ISBN 3-540-10235-3 Springer-Verlag Berlin Heidelberg New York

ISBN 0-387-10235-3 Springer-Verlag New York Heidelberg Berlin

Library of Congress Cataloging in Publication Data. Milner, Robin. A calculus of communicating systems. (Lecture notes in computer science; 92) Bibliography: p. Includes index. 1. Machine theory. 2. Formal languages. I. Title. II. Series. QA267.M53. 511.3 80-21068

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to the publisher, the amount of the fee to be determined by agreement with the publisher.

© by Springer-Verlag Berlin Heidelberg 1980

Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.  
2145/3140-543210

### ACKNOWLEDGEMENTS

This work was mainly done during my six-month appointment, from August 1979 to January 1980, at the Computer Science department in Aarhus University, Denmark. Although much of the ground work had been done previously it was mainly in response to their encouragement (to make the theory more accessible and related to practice), and to their informed criticism, that the material reached a somewhat coherent form. I am deeply grateful to them and their students for allowing me to lecture once a week on what was, at first, a loosely connected set of ideas, and for the welcoming environment in which I was able to put the ideas in order. I also thank Edinburgh University for awarding me five months sabbatical leave subsequently, which helped me to complete the task in a reasonable time.

The calculus presented here grew out of work which was inspired by Dana Scott's theory of computation, though it has since diverged in some respects. At every stage I have been influenced by Gordon Plotkin; even where I cannot trace particular ideas to him I have been greatly illuminated by our discussions and by his chance remarks, and without them the outcome would certainly be less than it is. I would also like to thank others with whom I have worked: George Milne, with whom I worked out the Laws of Flow Algebra; Matthew Hennessy, with whom the notion of observation equivalence developed; and Tony Hoare, whose parallel work on different but strongly related ideas, expressed in his "Communicating Sequential Processes", has been a strong stimulus.

Many people have given detailed and helpful criticisms of the manuscript, and thus improved its final form. In particular I thank Michael Gordon and David MacQueen, who went through it all in detail in a Seminar at the Information Sciences Institute, University of California; this not only exposed some mistakes and obscurities but gave me more confidence in the parts they didn't criticise.

Finally, I am very thankful to Dorothy McKie and Gina Temple for their patience and skill in the long and involved task of typing.

## CONTENTS

0.	<u>Introduction</u>	1
	Purpose - Character - Related Work - Evolution - Outline.	
1.	<u>Experimenting on Nondeterministic Machines</u>	9
	Traditional equivalence of finite state acceptors - Experimenting upon acceptors - Behaviour as a tree - Algebra of RSTs - Unobservable actions.	
2.	<u>Synchronization</u>	19
	Mutual experimentation - Composition, restriction and relabelling - Extending the Algebra of STs - A simple example: binary semaphores - The ST Expansion Theorem.	
3.	<u>A case study in synchronization and proof techniques</u>	33
	A scheduling problem - Building the scheduler as a Petri Net - Observation equivalence - Proving the scheduler.	
4.	<u>Case studies in value-communication</u>	47
	Review - Passing values - An example: Data Flow - Derivations - An example: Zero searching.	
5.	<u>Syntax and Semantics of CCS</u>	65
	Introduction - Syntax - Semantics by derivations - Defining behaviour identifiers - Sorts and programs - Direct equivalence of behaviour programs - Congruence of behaviour programs - Congruence of behaviour expressions and the Expansion Theorem.	
6.	<u>Communication Trees (CTs) as a model of CCS</u>	84
	CTs and the dynamic operations - CTs and the static operations - CTs defined by recursion - Atomic actions and derivations of CTs - Strong equivalence of CTs - Equality in the CT model - Summary.	
7.	<u>Observation equivalence and its properties</u>	98
	Review - Observation equivalence in CCS - Observation congruence - Laws of observation congruence - Proof techniques - Proof of Theorem 7.7 - Further exercises.	

8.	<u>Some proofs about Data Structures</u>	111
	Introduction - Registers and memories - Chaining operations - Pushdowns and queues.	
9.	<u>Translation into CCS</u>	126
	Discussion - The language P - Sorts and auxiliary definitions - Translation of P - Adding procedures to P - Protection of resources.	
10.	<u>Determinacy and Confluence</u>	138
	Discussion - Strong confluence - Composite guards and the use of confluence - Strong determinacy: Confluent determinate CCS - Proof in DCCS: the scheduler again - Observation confluence and determinacy.	
11.	<u>Conclusion</u>	158
	What has been achieved? - Is CCS a programming language? - The question of fairness - The notion of behaviour - Directions for further work.	
	<u>Appendix:</u> Properties of congruence and equivalence.	166
	<u>References</u>	169