Robin Milner

A Calculus of Communicating Systems

Springer-Verlag
Berlin Heidelberg New York 1980
CONTENTS

0. Introduction
   Purpose - Character - Related Work - Evolution - Outline. 1

1. Experimenting on Nondeterministic Machines
   Traditional equivalence of finite state acceptors - Experimenting
   upon acceptors - Behaviour as a tree - Algebra of RSTs -
   Unobservable actions. 9

2. Synchronization
   Mutual experimentation - Composition, restriction and relabelling -
   Extending the Algebra of STs - A simple example: binary semaphores -
   The ST Expansion Theorem. 19

3. A case study in synchronization and proof techniques
   A scheduling problem - Building the scheduler as a Petri Net -
   Observation equivalence - Proving the scheduler. 33

4. Case studies in value-communication
   Review - Passing values - An example: Data Flow - Derivations -
   An example: Zero searching. 47

5. Syntax and Semantics of CCS
   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. 65

6. Communication Trees (CTs) as a model of CCS
   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. 84

7. Observation equivalence and its properties
   Review - Observation equivalence in CCS - Observation congruence -
   Laws of observation congruence - Proof techniques - Proof of Theorem 7.7 - Further exercises. 98
8. **Some proofs about Data Structures**  
Introduction - Registers and memories - Chaining operations - Pushdowns and queues.

9. **Translation into CCS**  
Discussion - The language P - Sorts and auxiliary definitions - Translation of P - Adding procedures to P - Protection of resources.

10. **Determinacy and Confluence**  
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. **Conclusion**  
What has been achieved? - Is CCS a programming language? - The question of fairness - The notion of behaviour - Directions for further work.

**Appendix**: Properties of congruence and equivalence.

**References**