

**Milner, Robin**

**Communicating and mobile systems: The  $\pi$ -calculus.** (English) [Zbl 0942.68002](#)  
Cambridge: Cambridge University Press. xii, 161 p. (1999).

This book forms the long-awaited successor to the textbook and standard reference on Milner's Calculus of Communicating Systems (CCS) [*R. Milner*, Communication and concurrency (Prentice Hall, New York) (1989; [Zbl 0683.68008](#))]. Apart from describing the  $\pi$ -calculus (an extension of CCS with mobility) it contains a new, more streamlined development of CCS proper. While described by the author as a textbook it may well become the standard work on the  $\pi$ -calculus which since its inception a decade ago enjoys great popularity among researchers in concurrency and has an increasing number of applications. So, what are CCS and the  $\pi$ -calculus? CCS provides a formal model for the kind of concurrent computation where concurrency arises from the nature of the system rather than from aiming for higher speed through parallelism. Typical examples are communication protocols, networks, scheduling, etc. CCS provides a mathematisation of such systems thus making them amenable to formal analysis such as verification of logical properties (e.g. equivalence, temporal properties), simulation, and programming language design. Apart from sequential and parallel composition as well as (often guarded) nondeterministic choice the main feature of CCS is a communication construct allowing two given processes to exchange information and to alter their subsequent behaviour according to the information transmitted. The  $\pi$ -calculus goes beyond CCS by allowing such communication to take place along named channels which themselves may have been communicated previously so that the topology of the modelled system is allowed to change dynamically (this is called "mobility"). The running example in the book under review of a real-life system where such generality is needed, is a mobile phone network where the connections between the user and transmitter stations change all the time. The world wide web, of course, provides for a plethora of other examples. Quite notably, however, at the time the  $\pi$ -calculus has been invented, instances of genuine mobility were rather limited. The elegance of the  $\pi$ -calculus lies in its isolating the essence of such mobility with hardly any commitment to a particular application or implementation. From a logician's point of view the probably most interesting aspect of this is the *new* construct which generates a fresh name that may – quite unlike bound variables in, say, the  $\lambda$ -calculus – be extruded from its scope by communication. In this way, in a construct such as  $new\ x.P(x)|Q(x)$  the processes  $P$  and  $Q$  share a common channel name  $x$  not available to any other process. They may, however, at any time communicate  $x$  on some other channel to the outside. Having developed appropriate evaluation rules and semantics for this construct constitutes in the reviewer's opinion the most important intellectual achievement of the  $\pi$ -calculus. The first part of the book gives a thorough account of CCS. Among the highlights are a much improved syntax and a more abstract definition of weak bisimulation. In the new syntax a process is viewed as abstracted over the set of free names contained in it and hiding of names is hence treated as a binding construct (called *new*). This syntax allows for the  $\pi$ -calculus to be presented as an extension of CCS rather than an altogether new system as has been the case historically. This is done in the second part where after a general motivation of mobility the  $\pi$ -calculus with its evaluation semantics and theory of congruence and observational equivalence is introduced. Another aspect of possible interest to logicians is the fact that process equivalence is stable under injective renaming of names, but not under arbitrary renaming. This provides an elementary indication as to why sheaves over sets with injective functions or equivalently sets equipped with a continuous  $S_{\mathbb{N}}$  action can serve as a natural semantic model of the  $\pi$ -calculus. Each chapter ends with a brief summary. Solutions to exercises as well as an indication as to their degree of difficulty would have been appreciated. At least one is apparently an open problem!

Reviewer: [Martin Hofmann](#) (Edinburgh)

**MSC:**

- 68-02 Research exposition (monographs, survey articles) pertaining to computer science
- 68Q85 Models and methods for concurrent and distributed computing (process algebras, bisimulation, transition nets, etc.)
- 03-02 Research exposition (monographs, survey articles) pertaining to mathematical logic and foundations
- 68Q10 Modes of computation (nondeterministic, parallel, interactive, probabilistic, etc.)
- 68Q60 Specification and verification (program logics, model checking, etc.)
- 68Q55 Semantics in the theory of computing
- 68Q42 Grammars and rewriting systems
- 68N30 Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.)
- 68N15 Theory of programming languages
- 03B70 Logic in computer science

Cited in <b>1</b> Review Cited in <b>272</b> Documents
---

**Keywords:**

concurrency; parallelism; programming languages; modelling and verification; calculus of communicating systems;  $\pi$ -calculus; extension of CCS with mobility

**Software:**

Pict