Foreword

Computer science aims to explain the way computational systems behave for us. The notion of calculational process, or algorithm, is a lot older than computing technology; so, oddly enough, a lot of computer science existed before modern computers. But the invention of real stored-program computers presented enormous challenges; these tools can do a lot for us if we describe properly what we want done. So computer science has made immense strides in ways of *presenting* data and algorithms, in ways of manipulating these presentations themselves as data, in matching algorithm description to task description, and so on. Technology has been the catalyst in the growth of modern computer science.

The first large phase of this growth was in free-standing computer systems. Such a system might have been a single computer program, or a multi-computer serving a community by executing several single programs successively or simultaneously. Computing theorists have built many mathematical models of these systems, in relation to their purposes. One very basic such model – the λ-calculus – is remarkably useful in this role, even if it was designed by Alonzo Church around 1940.

The second phase of the growth of computer science is in response to the advent of computer networks. No longer are systems freestanding; they interact, collaborate and interrupt each other. This has an enormous effect on the way we think about our systems. We can no longer get away with considering each system as sequential, goal-directed, deterministic or hierarchical; networks are none of these. So if we confine ourselves to such concepts then we remain dumb if asked to predict whether a network will behave in a proper – or an improper – way; for example, whether someone logging in to his bank may (as happened recently) find himself scanning someone else's account instead of his own.

The present book is a rigorous account of a basic calculus which aims to underpin our theories of interactive systems, in the same way that the λ-calculus did for freestanding computation. The authors are two of the original researchers on the π-calculus, which is now over ten years old and has served as a focus for much theoretical and practical experiment. It cannot claim to be definitive; in fact, since it was designed it has become common to express ideas about interaction and mobility in variants of the calculus. So it has become a kind of workshop of ideas.

That's the spirit in which the book is written. Half the book analyses the constructions of the calculus, searching out its meaning and exploring its expressivity by looking at weaker variants, or by looking at various type disciplines. Enthusiasts about types in programming will be struck to find that π-calculus types don't just classify *values*; they classify *patterns of behaviour*. This reflects the fact that what matters most in mobile interactive systems is not values, but connectivity and mobility of processes. With or without types, the unifying feature is *behaviour*, and what it means to say that two different processes behave the same.

The later part of the book deals with two generic applications. One of these is classical; how the π-calculus can actually do the old job which the λ-calculus does in underpinning conventional programming. The other is modern; how the calculus informs one of the most important models of interaction, the *object-oriented* model. These applications bring together much of the theory developed earlier; together, they show that a small set of constructs, provided that they emphasize *interaction* rather than calculation, can still bring some conceptual unity to the greatly extended scope of modern computing.

This book has been a labour of love for the authors over several years. Their scholarship is immense, and their organisation of ideas meticulous. As one privileged to have worked closely with them both, it's a great pleasure to be able to recommend the result as a storehouse of ideas and techniques which is unlikely to be equalled in the next decade or two.

Robin Milner
Cambridge
February 2001