Preface

Computers that "program themselves" has long been an aim of computer scientists. Recently genetic programming (GP) has started to show its promise by automatically evolving programs. Indeed in a small number of problems GP has evolved programs whose performance is similar to or even slightly better than that of programs written by people. The main thrust of GP has been to automatically create functions. While these can be of great use they contain no memory and relatively little work has addressed automatic creation of program code including stored data. It is this issue which this book addresses.

Motivated by the observation from software engineering that data abstraction (e.g. via abstract data types) is essential in programs created by human programmers we will show that abstract data types can be similarly beneficial to the automatic production of programs using GP.

We will show how abstract data types (stacks, queues and lists) can be evolved using genetic programming, demonstrate GP can evolve general programs which solve the nested brackets problem, recognise a Dyck context free language and implement a simple four function calculator. In these cases an appropriate data structure is beneficial compared to simple indexed memory. This book also includes a survey of GP, including a critical review of experiments with evolving memory and reports investigations of real world electrical network maintenance scheduling problems that demonstrate that Genetic Algorithms can find low cost viable solutions to such problems.

The subtitle of this book is initially derived from the famous expression by Niklaus Wirth *Algorithms + Data Structures = Programs* used for the title of his book [Wirth, 1975] and subsequently rephrased by Zbigniew Michalewicz to *Genetic Algorithms + Data Structures = Evolution Programs* as the title of his book [Michalewicz, 1994]. With *Genetic Programming + Data Structures = Automatic Programming!* we continue the common thread. All three books share a common idea. "To build [or evolve] a successful program, appropriate data structures should be used together with appropriate algorithms (these correspond to genetic operators used for transforming individual chromosomes [or programs])", [Michalewicz, 1994, page xi]. It would unfair to the reader to mislead them into thinking it is possible at present to automatically evolve every program. The subtitle reflects the hope that this book will be a step towards automatic programming.

This book is an updated version of my thesis which was submitted as part of the requirements of the degree of Doctor of Philosophy in the University of London in September 1996 and obtained in December 1996. The thesis was written at the Computer Science Department of University College, London, which is part of the University of London.

W. B. LANGDON