Preface

This book has its roots in lecture notes which we prepared for a course on constraint logic programming given at both Monash University and the University of Melbourne for fourth year level students. The rapidly growing interest in constraint programming and, in particular, constraint logic programming, suggested that a course, which treated the area in some depth, should be offered. However, no adequate text was available. There are several good texts covering some aspects of such a course such as Tsang's *Foundations of Constraint Satisfaction*, Van Hentenryck's *Constraint Satisfaction in Logic Programming* (for constraint logic programming techniques for finite domains), and Sterling and Shapiro's *The Art of Prolog* (for logic programming techniques). However, a comprehensive introductory text covering constraint programming, one of the most interesting programming language paradigms ever invented, did not exist.

We hope that the present book provides such an introduction to the discipline of constraint programming and, in particular, constraint logic programming languages. It is self-contained, providing the necessary background material from artificial intelligence, logic programming, operations research and mathematical programming.

The book covers a wide spectrum of topics, from constraint solving techniques to programming methodologies for constraint programming languages. The chapters are organized into three parts:

1. **Constraints**. In this part, we discuss particular constraint domains and constraint solving techniques as well as other operations on constraints such as simplification, optimization and implication. Arithmetic, tree and finite domain constraint solving are discussed in some detail. Constraint satisfaction problems and generic constraint solving techniques are also introduced.

2. **Constraint Logic Programming**. Here, we introduce constraint logic programming languages and the way in which they can be used to build user-defined constraints. We then continue by examining techniques for modelling constraint problems using constraint logic programs. Example programs are taken from real-world applications of constraint programming and include financial modelling, circuit analysis, engineering design and job scheduling. Then we explore efficiency issues and how to write more efficient programs. Next we look at several advanced programming techniques that are used in constraint logic programming for extending or modifying the underlying solver and execution mechanism. Finally we discuss in some detail how practical CLP systems are implemented.

3. **Other Constraint Programming Languages**. In this part, we introduce other approaches to constraint programming. These include: constraint databases, which are similar to constraint logic programs, except that a bottom-up execution mechanism is used; concurrent constraint programming languages, which extend constraint logic programming languages by allowing parallel execution of procedures and asynchronous communication between procedures by way of constraint entailment; constraint toolkits, which provide constraint-solving within an object-oriented language; mathematical constraints languages, which are languages used by mathematicians to reason about constraints; constraint functional programming languages which combine constraint programming with functional programming; and constraint imperative languages which combine constraint programming techniques with object oriented languages.

How to Use This Book

The book has been designed to allow its use in a variety of courses at the undergraduate or postgraduate level. The first and second parts contain both core and supplementary chapters, the latter covering more esoteric or advanced material. The core chapters in the first two parts can be used to introduce constraint programming as part of an undergraduate course in modern programming language paradigms. Covering all the material in the first two parts provides a short postgraduate course in constraint programming. A longer course can also cover some of the material in the third part.

Some sections of the book are annotated with a (*). This indicates that the material is not core to the book and can be omitted if desired. Exercises and practical exercises that are annotated (*) are either more challenging or go beyond the material within the chapter.

An outline of the organization of the book is illustrated in Figure A. Core chapters are aligned in a column under the introduction. An arrow between two chapters indicates a dependency.

Programs are presented in a manner that is independent of any actual constraint logic programming language. This is because there is, as yet, no agreed syntax for these. When teaching from this book we have made use of the $CLP(\mathcal{R})$ and ECLiPSe constraint logic programming systems, which are both freely available for research and teaching purposes, as well as SICStus Prolog. Details of how to obtain these constraint logic programming systems is contained in the practical exercises sections of Chapters 1 and 3. Information about how to use each of these systems is given in the practical exercises sections of Chapters 1, 2, 3, 4, 8 and 9. The book is also designed for easy use with other constraint logic programming systems such as Prolog IV, `clp(fd)`, and others.

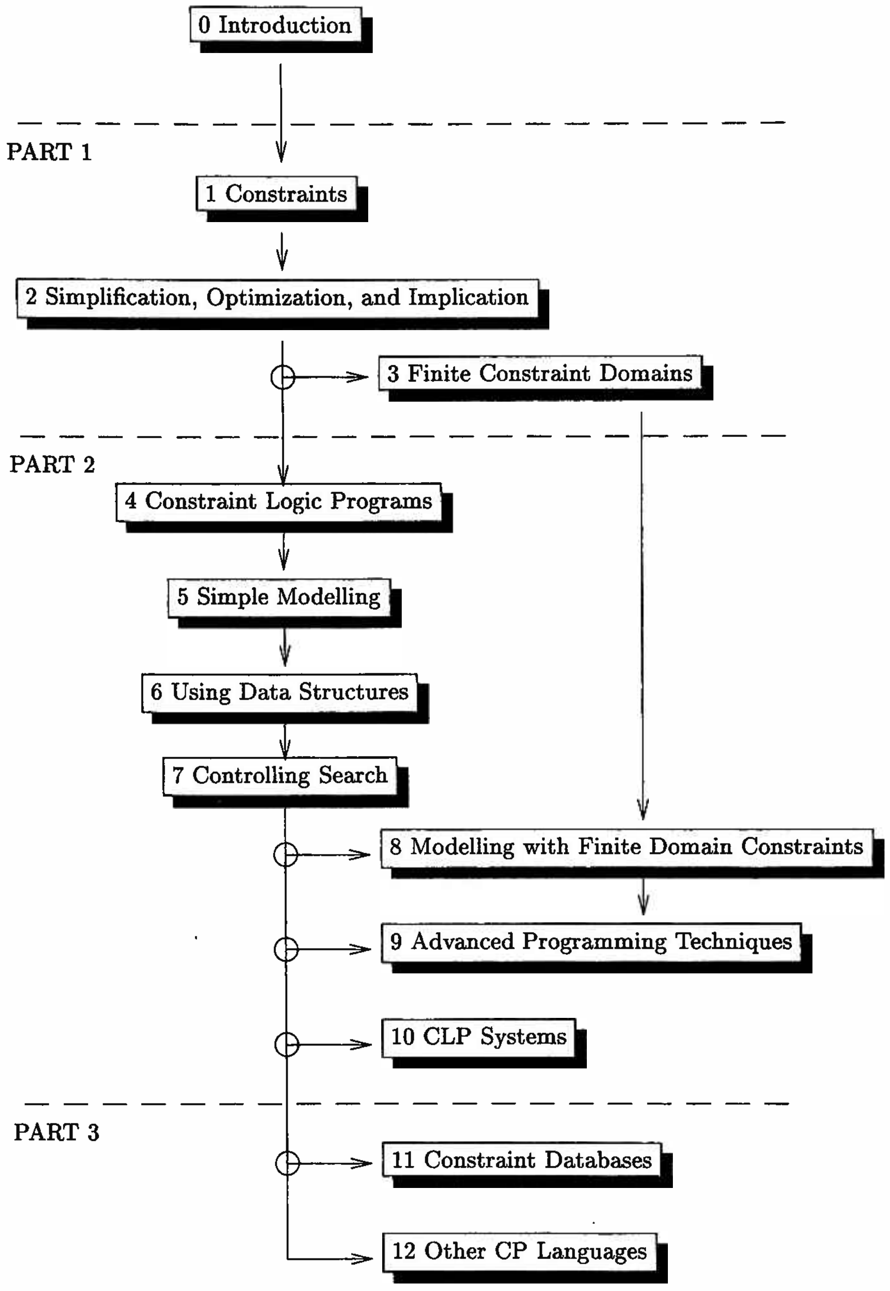
0 Introduction
PART 1
1 Constraints
2 Simplification, Optimization, and Implication
3 Finite Constraint Domains
PART 2
4 Constraint Logic Programs
5 Simple Modelling
6 Using Data Structures
7 Controlling Search
8 Modelling with Finite Domain Constraints
9 Advanced Programming Techniques
10 CLP Systems
PART 3
11 Constraint Databases
12 Other CP Languages

Figure A: Chapter dependencies.

Acknowledgements

Numerous people helped us by reviewing drafts of the book: John Crossley, Michael Maher and Rosemary Marriott provided invaluable help by reading and rereading successive drafts of the book, while Natashia Boland, Alan Borning, Graham Farr, Volker Gaede, Joxan Jaffar, Jimmy Lee, Steve Murphy, Harald Søndergaard, Mark Wallace and Roland Yap provided advice about specific topics and chapters. Of course, we are responsible for all remaining errors.

Harald Søndergaard provided many of the notes and references for Chapter 1 and we are grateful to Gert Smolka for permission to include the Oz program on page 435 and to Jean-François Puget for permission to include the ILOG SOLVER program on page 441. We would also like to thank Bob Prior of MIT Press for his encouragement and support and Deborah Cantor-Adams for her careful proofreading of the final manuscript.

Our partners, to whom this book is dedicated, have also contributed in many ways to this book, both directly and indirectly. Nicole Beyer provided valuable advice about graphic design and layout, while María García de la Banda carefully proofread several drafts. Most importantly, however, they helped each of us to keep the writing of this book in perspective.

Kim Marriott
Peter Stuckey

Programming with Constraints