

# CONTENTS



UNIVERSIDAD NACIONAL DE ENTRE RÍOS  
FACULTAD DE INGENIERIA  
CENTRO DE MEDIOS  
BIBLIOTECA

22 96

## Preface xi

### 1 Introduction 1

- 1.1 Separating Levels of Concern 2
- 1.2 A Programming Example of Data Abstraction 4
- Exercises 8

### 2 Mathematical Background 10

- 2.1 Propositions 10
- 2.2 Sets 17
  - 2.2.1 Operators on sets 18
  - 2.2.2 Ordered pairs 19
  - 2.2.3 Sets,  $n$ -tuples, and Pascal 20
- 2.3 Functions 22
  - 2.3.1 Revisited: levels of concern 23
  - 2.3.2 The four simple classes of functions 24
  - 2.3.3 Selectors and field identifiers 27
  - 2.3.4 Predicates and quantifiers 27
- 2.4 Natural Numbers and Mathematical Induction 29
  - 2.4.1 Nature and purpose of formal definitions 29
  - 2.4.2 An axiomatic definition of the natural numbers 31
- 2.5 Complexity measures 38
- Exercises 40

### 3 Recursive Programs and Recursive Definitions 43

- 3.1 The Tower of Hanoi 43
- 3.2 Sorting 50
- 3.3 Recursive Definitions 52
- Exercises 53

**4 Abstract Data types 55**

- 4.1 Specification 55
    - 4.1.1 *Specification of data types* 56
    - 4.1.2 *Completeness of specifications* 59
  - 4.2 Implementation 62
    - 4.2.1 *Size-restricted (bounded) objects* 63
    - 4.2.2 *Implementation based on a scalar type* 63
    - 4.2.3 *Implementation based on integer numbers* 66
    - 4.2.4 *Implementation based on arrays* 68
    - 4.2.5 *The substance of an implementation: the functions ABSTR and VALID* 72
    - 4.2.6 *Direct implementation of the axioms* 80
    - 4.2.7 *Array-based versus direct implementation* 84
  - 4.3 Values and Mutable Objects 84
    - 4.3.1 *Specification of mutable objects* 87
    - 4.3.2 *Implementation of mutable objects* 91
    - 4.3.3 *Dynamic creation of mutable objects* 95
    - 4.3.4 *Objects or values, which should we choose?* 97
- Exercises 98

**5 Cells 101**

- 5.1 Specification of Cells as Values 102
- 5.2 Implementation of Cell Values 103
  - 5.2.1 *Implementation by integer numbers* 103
  - 5.2.2 *Direct implementation of the cell axioms* 104
  - 5.2.3 *Cells built with Pascal's variant record types* 106
- 5.3 Cells as Mutable Objects 109
- 5.4 Implementation of Cell Objects 113
- 5.5 Common Language Support for Cells 116

**6 Linear Data Types 119**

- 6.1 Specification 120
  - 6.1.1 *Functions on sequences with only one parameter* 120
  - 6.1.2 *Some functions with two parameters* 121
  - 6.1.3 *How to construct specifications* 125
  - 6.1.4 *Sequences with a "point of interest"* 129
  - 6.1.5 *The occasional need for "hidden functions"* 131
  - 6.1.6 *The specification of the Pascal type "file"* 134
- 6.2 Implementation of Sequences as Values 135
  - 6.2.1 *Implementation based on enumerated types* 136
  - 6.2.2 *Implementation based on integer numbers* 136

- 6.2.3 *Implementation based on arrays* 141
- 6.2.4 *Direct implementation of the axioms* 141
- 6.2.5 *Linked implementation* 142
- 6.3 Implementation of Sequences as Objects 152
  - 6.3.1 *A simple implementation of queue objects* 153
  - 6.3.2 *A queue as a circular linked list* 154
  - 6.3.3 *Sequences with PREV as singly linked lists* 155
  - 6.3.4 *Sequence objects as doubly linked lists* 156
- 6.4 The File of Tokens: a Case Study 158
  - Exercises 166
  
- 7 Binary Trees 169**
  - 7.1 Binary Trees and Linear Types 170
  - 7.2 More Properties of Binary Trees 172
  - 7.3 A Simple Method of Implementing Trees 176
  - 7.4 Applications of Binary Trees 177
    - 7.4.1 *Trees and arithmetic expressions: a case study* 177
    - 7.4.2 *Printing a tournament schedule* 188
    - 7.4.3 *Printing a tree root-first* 189
    - 7.4.4 *Binary search trees* 191
    - 7.4.5 *Traversing binary trees* 197
  - 7.5 Other Primitive Operations on Trees 197
  - 7.6 Trees as Mutable Objects 198
  - 7.7 More on Implementing Trees 207
    - 7.7.1 *Implementation based on arrays* 207
    - 7.7.2 *Implementation based on sequences* 209
    - 7.7.3 *Cell implementation of tree objects with sites* 210
  - Exercises 212
  
- 8 N-ary Trees and General Trees 216**
  - 8.1 Vectors and Sequences 216
  - 8.2 Definition of *N*-ary Trees and General Trees 218
  - 8.3 Specification of General Trees 219
  - 8.4 Typical Applications of General Trees 221
  - 8.5 Operations and Algorithms for General Trees 222
  - 8.6 Implementation of General Trees 225
  - Exercises 226
  
- 9 Sets, Dictionaries, and Graphs 227**
  - 9.1 Specification of Sets 227

---

9.2	Implementation of Sets	233
9.2.1	<i>Sets implemented by bit vectors</i>	234
9.2.2	<i>Sets implemented by sequences</i>	235
9.2.3	<i>Hashing</i>	236
9.2.4	<i>Implementation of sets by search trees</i>	240
9.2.5	<i>Tree balancing</i>	242
9.3	Functions, Dictionaries, Symbol Tables	246
9.3.1	<i>Implementation of the type "Funktion"</i>	249
9.4	Graphs and Relations	250
9.4.1	<i>A programming example</i>	252
9.4.2	<i>More operations on graphs</i>	254
9.4.3	<i>Implementation of graphs</i>	255
	Exercises	257
<b>10</b>	<b>Storage Management</b>	<b>259</b>
10.1	Reference Counts	260
10.2	Garbage Collection	263
10.3	Garbage Compaction	270
	Exercises	271
	<b>Bibliographical Notes</b>	<b>273</b>
	<b>Index</b>	<b>275</b>