Contents

Preface

1 Why Z?

1 Formal methods

1.1 What are formal methods? 3
1.2 What formal methods are not 5
1.3 When are formal methods useful? 5
1.4 How can we use formal methods? 6
1.5 Are formal methods too difficult? 11
1.6 Formal methods at work 12
1.7 Formal methods can be practical 12

2 Why Use Formal Methods?

2.1 True stories 15
2.2 Some popular fallacies 16
2.3 Some hopeful alternatives 18
## 3 Formal methods and project management 21

3.1 Work in stages 21

3.2 Gathering requirements 23

3.3 From informal requirements to formal specifications 25

3.4 Validating formal specifications 26

### Further reading 27

## 4 Introducing Z 31

## 5 A first example in Z 33

## 6 From prose to Z: control console 39

6.1 Informal requirements 40

6.2 Data flow diagram 44

6.3 State transition diagram 45

6.4 State transition table 46

6.5 Z 47

## 7 Introducing schemas: text editor 49

7.1 Basic types and abbreviation definitions 49

7.2 Axiomatic descriptions 50

7.3 State schemas 50

7.4 Initialization schemas 51

7.5 Operation schemas 52

7.6 Implicit preconditions 53
7.7 Schema calculus 55
7.8 The Way of Z 56

Further reading 59

III Elements of Z 63

8 Elements 63
8.1 Sets and types, declarations, and variables 63
8.2 Expressions and operators 71
8.3 Predicates, equations, and laws 74

9 Structure 78
9.1 Tuples and records 78
9.2 Relations, tables, and databases 80
9.3 Pairs and binary relations 81
9.4 Functions 88
9.5 Sequences 93
9.6 Operators 94

10 Logic 96
10.1 Basic predicates 96
10.2 Using predicates in Z 96
10.3 Relations as predicates 98
10.4 Logical connectives 100
10.5 Logic and natural language 105
10.6 Quantifiers 106
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.7 Z and Boolean types</td>
<td>108</td>
</tr>
<tr>
<td>10.8 Predicates and undefined expressions</td>
<td>110</td>
</tr>
<tr>
<td>11 Synthesis</td>
<td>112</td>
</tr>
<tr>
<td>11.1 Set comprehensions</td>
<td>112</td>
</tr>
<tr>
<td>11.2 Lambda expressions</td>
<td>114</td>
</tr>
<tr>
<td>11.3 Some simple formal specifications</td>
<td>115</td>
</tr>
<tr>
<td>11.4 Conveniences and shortcuts</td>
<td>116</td>
</tr>
<tr>
<td>11.5 Modelling systems and change</td>
<td>119</td>
</tr>
<tr>
<td>12 Schemas and schema calculus</td>
<td>122</td>
</tr>
<tr>
<td>12.1 Inside the schema boxes</td>
<td>122</td>
</tr>
<tr>
<td>12.2 Schema calculus: conjunction and disjunction</td>
<td>127</td>
</tr>
<tr>
<td>12.3 Schemas everywhere</td>
<td>132</td>
</tr>
<tr>
<td>12.4 Other schema calculus operators</td>
<td>134</td>
</tr>
<tr>
<td>13 Schema types and bindings</td>
<td>138</td>
</tr>
<tr>
<td>13.1 Schema types</td>
<td>138</td>
</tr>
<tr>
<td>13.2 Using schema types and bindings</td>
<td>142</td>
</tr>
<tr>
<td>14 Generic definitions and free types</td>
<td>146</td>
</tr>
<tr>
<td>14.1 Generics</td>
<td>146</td>
</tr>
<tr>
<td>14.2 Free types</td>
<td>148</td>
</tr>
<tr>
<td>15 Formal reasoning</td>
<td>149</td>
</tr>
<tr>
<td>15.1 Calculation and proof</td>
<td>150</td>
</tr>
<tr>
<td>15.2 Laws</td>
<td>152</td>
</tr>
<tr>
<td>15.3 Checking specifications</td>
<td>154</td>
</tr>
<tr>
<td>15.4 Precondition calculation</td>
<td>155</td>
</tr>
</tbody>
</table>
15.5 Formal reasoning and intuition 159
15.6 Machine-checked proof 159

Further reading 161

IV Studies in Z
16 Document control system 165
17 Text processing 169
  17.1 Breaking a text into words 169
  17.2 A word counting utility 171
  17.3 Filling paragraphs 171
18 Eight queens 174
19 Computer graphics and computational geometry 180
20 Rule-based programming 189
  20.1 Elements of rule-based programming 190
  20.2 Facts and rules 190
  20.3 Deducing new facts 191
  20.4 Checking the rules 193
  20.5 Specifying rule-based programs 194
  20.6 Conclusion 198
21 Graphical user interface 199
  21.1 Events 199
  21.2 Displays and dialogs 200
  21.3 Selecting a display 202
  21.4 Changing a setting value 203
21.5 Z and state transition systems 206
21.6 Changing the machine state 208
21.7 Conclusions 209

22 Safety-critical protection system 211
22.1 Partition 211
22.2 Refinement 214
22.3 Enforcing the safety requirements 217

23 Modelling large systems 218
23.1 A single subsystem 218
23.2 Many subsystems 221
23.3 Some useful idioms 225
23.4 Subsystems, conditions, and modes 227
23.5 Conclusion 230

24 Object-oriented programming 231
24.1 The object-oriented model and Z 231
24.2 Inheritance and schema inclusion 231
24.3 Object-oriented Z dialects 233

25 Concurrency and real time 234
25.1 Concurrency 234
25.2 Events 237
25.3 Real time 239

Further reading 241
V  Programming with Z

26  Refinement  247
    26.1  What is refinement?  247
    26.2  A refinement example  248
    26.3  Generalizing refinement  251
    26.4  Refinement strategies  252

27  Program derivation and formal verification  254
    27.1  Axiomatic program derivation and verification  255
    27.2  Refinement calculus  262

28  From Z to code  265
    28.1  Data structures  265
    28.2  State schemas  268
    28.3  Refinement from Z to code  270
    28.4  Functions and relations  280
    28.5  Operation schemas  280
    28.6  Schema expressions  284
    28.7  Modules and programs  285
    28.8  A larger example  287
    28.9  A final example  290

Further reading  297

A  Glossary of Z notation  299

B  Omitted features  304
| C | Operator precedence | 305 |
| D | The Z mathematical tool-kit | 308 |
| E | Selected Laws | 316 |
| F | Solutions to selected exercises | 322 |
| G | Other formal notations | 326 |

Bibliography | 328
Index | 340