

J.N. Oliveira

**PROGRAM DESIGN BY  
CALCULATION**

(DRAFT)

University of Minho  
*(in preparation)*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Why Program Design by Calculation? . . . . .	3
1.2	Why this Book? . . . . .	4
1.3	Book Structure . . . . .	5
1.4	How to Read This Book . . . . .	5
<b>I</b>	<b>Calculating with Functions</b>	<b>7</b>
<b>2</b>	<b>An Introduction to Pointfree Programming</b>	<b>9</b>
2.1	Introducing functions and types . . . . .	10
2.2	Functional application . . . . .	11
2.3	Functional equality and composition . . . . .	11
2.4	Identity functions . . . . .	14
2.5	Constant functions . . . . .	14
2.6	Monics and epics . . . . .	15
2.7	Isos . . . . .	17
2.8	Gluing functions which do not compose — products . . . . .	18
2.9	Gluing functions which do not compose — coproducts . . . . .	25
2.10	Mixing products and coproducts . . . . .	28
2.11	Natural properties . . . . .	31
2.12	Universal properties . . . . .	32
2.13	Guards and McCarthy’s conditional . . . . .	34
2.14	Gluing functions which do not compose — exponentials . . . . .	36
2.15	Elementary datatypes . . . . .	41
2.16	Finitary products and coproducts . . . . .	43
2.17	Initial and terminal datatypes . . . . .	45
2.18	Sums and products in HASKELL . . . . .	46

2.19 Exercises . . . . .	49
2.20 Bibliography notes . . . . .	51
<b>3 Recursion in the Pointfree Style</b>	<b>53</b>
3.1 Motivation . . . . .	53
3.2 Introducing inductive datatypes . . . . .	59
3.3 Observing an inductive datatype . . . . .	64
3.4 Synthesizing an inductive datatype . . . . .	68
3.5 Introducing (list) catas, anas and hylos . . . . .	69
3.6 Inductive types more generally . . . . .	74
3.7 Functors . . . . .	75
3.8 Polynomial functors . . . . .	77
3.9 Polynomial inductive types . . . . .	79
3.10 F-algebras and F-homomorphisms . . . . .	80
3.11 F-catamorphisms . . . . .	81
3.12 Parameterization, type functors and cata-absorption . . . . .	84
3.13 A catalogue of standard polynomial inductive types . . . . .	88
3.14 Functors and type functors in HASKELL . . . . .	90
3.15 The mutual-recursion law . . . . .	91
3.15.1 “Banana-split”: a corollary of the mutual-recursion law . . . . .	95
3.16 Inductive datatype isomorphism . . . . .	97
3.17 Bibliography notes . . . . .	97
<b>4 Why Monads Matter</b>	<b>99</b>
4.1 Partial functions . . . . .	99
4.2 Putting partial functions together . . . . .	100
4.3 Lists . . . . .	102
4.4 Monads . . . . .	103
4.4.1 Properties involving (Kleisli) composition . . . . .	105
4.5 Monadic application (binding) . . . . .	106
4.6 Sequencing and the $\text{do}$ -notation . . . . .	107
4.7 Generators and comprehensions . . . . .	108
4.8 Monads in HASKELL . . . . .	109
4.8.1 Monadic I/O . . . . .	110
4.9 Bibliography notes . . . . .	111

<b>II</b>	<b>Moving Away From (Pure) Functions</b>	<b>113</b>
<b>5</b>	<b>Quasi-inductive datatypes</b>	<b>115</b>
5.1	Introducing Non-inductive Datatypes . . . . .	115
5.2	Structural Induction . . . . .	121
5.3	Well-founded coalgebras and induction . . . . .	122
5.4	Datatype invariants and proof obligations . . . . .	124
5.5	Binary relations and finite mappings . . . . .	126
5.6	Finite mapping induction principle . . . . .	128
5.7	An overview of the powerset and finite mapping algebras . . . . .	129
5.8	The relational database model . . . . .	129
5.9	Exercises . . . . .	131
5.10	Bibliography notes . . . . .	133
<b>III</b>	<b>Calculating with Relations</b>	<b>135</b>
<b>6</b>	<b>Introduction to the Pointfree Relational Calculus</b>	<b>137</b>
6.1	Functions are not enough . . . . .	137
6.2	Specifications as “properties” . . . . .	139
6.3	Relational approach . . . . .	139
6.4	Pre/post specification style . . . . .	140
6.5	From predicates to relations . . . . .	141
6.6	Basic relational combinators . . . . .	144
6.7	Converse . . . . .	146
6.8	Meet and converse . . . . .	147
6.9	Pointwise vs pointfree notation . . . . .	148
6.10	Orders and their taxonomy (A) . . . . .	151
6.11	Derived combinators . . . . .	153
6.12	Entireness and simplicity . . . . .	154
6.13	Surjectiveness and injectiveness . . . . .	155
6.14	Binary relation taxonomy . . . . .	156
6.15	Reasoning about functions . . . . .	156
6.16	Galois connections . . . . .	161
6.17	Converse in a Galois connection . . . . .	163
6.18	Functions in a Galois connection . . . . .	164
6.19	Relational division . . . . .	165
6.20	Meet and join . . . . .	169

6.21	Relational <i>split</i> . . . . .	169
6.22	Relational <i>either</i> . . . . .	171
6.23	Meaning of VDM-SL specs . . . . .	175
6.24	Relational semantics of VDM-SL . . . . .	177
6.25	Relational McCarthy conditional . . . . .	179
6.26	Reasoning about VDM-SL . . . . .	180
6.27	Bibliography notes . . . . .	184
<b>7</b>	<b>An Introduction to Relational Hylomorphisms</b>	<b>185</b>
7.1	“How” does one specify? . . . . .	185
7.2	Divide-and-conquer (formally) . . . . .	186
7.3	Relators . . . . .	187
7.4	Properties of relators . . . . .	188
7.5	Equations and fixpoints . . . . .	190
7.6	Solving (Fixpoint) Equations I . . . . .	191
7.7	Solving (Fixpoint) Equations II . . . . .	192
7.8	Solving (Fixpoint) Equations III . . . . .	193
7.9	Solving (Fixpoint) Equations IV . . . . .	193
7.10	Solving relational equations . . . . .	194
7.11	Laws of the Fixpoint Calculus . . . . .	194
7.12	$\mu$ -fusion theorem . . . . .	197
7.13	Hylo(cata)-fusion . . . . .	199
7.14	Hylo(ana)-fusion . . . . .	200
7.15	Examples: VDM collective types . . . . .	201
7.16	Relational cata(ana)morphisms . . . . .	202
7.17	Inductive coreflexives . . . . .	203
7.18	VDM-SL data type $\text{map } A \text{ to } B$ . . . . .	205
7.19	Hylos as unique solutions . . . . .	206
7.20	Accessibility and membership . . . . .	207
7.21	Hylo-factorization Theorem . . . . .	210
7.22	Virtual data-structuring . . . . .	211
7.23	Final note on inductive relation $\prec$ . . . . .	212
7.24	Bibliography notes . . . . .	212
<b>8</b>	<b>Theorems for Free</b>	<b>213</b>
8.1	Parametric polymorphism: why? . . . . .	213
8.2	Free theorem of type $t$ . . . . .	214
8.3	Reynolds arrow operator . . . . .	215

8.4	Pointwise version of FT . . . . .	217
8.5	Pointwise version of FT . . . . .	217
8.6	Second example: FT of $(\_ -)$ . . . . .	218
8.7	Bibliography notes . . . . .	219
<b>IV Data Refinement by Calculation</b>		<b>221</b>
<b>9</b>	<b>On Data Representation</b>	<b>223</b>
9.1	Introduction . . . . .	223
9.2	Data refinement . . . . .	225
9.3	Representation relations . . . . .	226
9.4	Right invertibility . . . . .	227
9.5	Refinement inequations . . . . .	227
9.6	Relational representation . . . . .	229
9.7	Functional representation . . . . .	230
9.8	Concrete invariants . . . . .	230
9.9	A fundamental iso abstraction . . . . .	231
9.10	Pointfree $untot = (i_1^\circ)$ . . . . .	232
9.11	Properties of $\leq$ : . . . . .	233
9.12	Structural data refinement . . . . .	234
9.13	The finite map bifunctor . . . . .	239
9.14	Transposing relations . . . . .	242
9.15	Transposing finite relations . . . . .	243
9.16	Recursive data refinement . . . . .	244
9.17	Recursion “removal” . . . . .	245
9.18	Closure and wellfoundedness . . . . .	248
9.19	O.O. Data Implementation . . . . .	249
9.20	Multiple inheritance . . . . .	249
9.21	Bibliography notes . . . . .	250
<b>10</b>	<b>On Algorithmic Refinement</b>	<b>253</b>
10.1	Implicit/explicit refinement . . . . .	253
10.2	Handling refinement equations . . . . .	256
10.3	Solving refinement equations . . . . .	256
10.4	Properties of $\vdash$ . . . . .	257
10.5	Stepwise refinement . . . . .	258
10.6	Refinement is a partial order . . . . .	259

10.7	Main refinement strategies . . . . .	260
10.8	Data refinement in full . . . . .	261
10.9	Analysis of refinement equation . . . . .	262
10.10	Solving refinement equations . . . . .	263
10.11	Functional solutions . . . . .	263
10.12	Calculation of <code>while/for</code> loops . . . . .	265
10.13	Bibliography notes . . . . .	265
<b>A</b>	<b>Haskell Support Library in Haskell</b>	<b>267</b>
A.0.1	Set.hs . . . . .	267
<b>B</b>	<b>Solutions to Selected Exercises</b>	<b>269</b>

# List of Figures





# List of Exercises

Exercise 2.1 . . . . .	15
Exercise 2.2 . . . . .	17
Exercise 2.3 . . . . .	24
Exercise 2.4 . . . . .	24
Exercise 2.5 . . . . .	28
Exercise 2.6 . . . . .	28
Exercise 2.7 . . . . .	30
Exercise 2.8 . . . . .	30
Exercise 2.9 . . . . .	30
Exercise 2.10 . . . . .	32
Exercise 2.11 . . . . .	32
Exercise 2.12 . . . . .	34
Exercise 2.13 . . . . .	35
Exercise 2.14 . . . . .	35
Exercise 2.15 . . . . .	36
Exercise 2.16 . . . . .	40
Exercise 2.17 . . . . .	43
Exercise 2.18 . . . . .	44
Exercise 2.19 . . . . .	46
Exercise 2.20 . . . . .	49
Exercise 2.21 . . . . .	50
Exercise 2.22 . . . . .	50
Exercise 2.23 . . . . .	50
Exercise 2.24 . . . . .	50
Exercise 2.25 . . . . .	51
Exercise 3.26 . . . . .	64
Exercise 3.27 . . . . .	73
Exercise 3.28 . . . . .	74

Exercise 3.29 . . . . .	74
Exercise 3.30 . . . . .	79
Exercise 3.31 . . . . .	89
Exercise 3.32 . . . . .	90
Exercise 3.33 . . . . .	90
Exercise 3.34 . . . . .	91
Exercise 3.35 . . . . .	91
Exercise 3.36 . . . . .	96
Exercise 4.37 . . . . .	103
Exercise 4.38 . . . . .	103
Exercise 4.39 . . . . .	106
Exercise 4.40 . . . . .	111
Exercise 4.41 . . . . .	111
Exercise 5.42 . . . . .	120
Exercise 5.43 . . . . .	120
Exercise 5.44 . . . . .	122
Exercise 5.45 . . . . .	124
Exercise 5.46 . . . . .	126
Exercise 5.47 . . . . .	130
Exercise 5.48 . . . . .	130
Exercise 5.49 . . . . .	131
Exercise 5.50 . . . . .	132
Exercise 5.51 . . . . .	132
Exercise 5.52 . . . . .	132