

# Table of Contents

|   |    |
|---|----|
| Chapter 1. Generic Haskell: Practice and Theory . . . . .         | 1  |
| <i>R. Hinze and J. Jeuring</i>                                    |    |
| 1 Introduction . . . . .  | 1  |
| 1.1 Type Systems . . . . .  | 2  |
| 1.2 Haskell’s <b>data</b> Construct . . . . .                     | 3  |
| 1.3 Towards Generic Programming . . . . .                         | 7  |
| 1.4 Towards Generic Haskell . . . . .                             | 12 |
| 1.5 Stocktaking . . . . .   | 17 |
| 1.6 Related Work . . . . .  | 18 |
| 2 Generic Haskell—Practice . . . . .                              | 22 |
| 2.1 Mapping Functions . . . . .                                   | 22 |
| 2.2 Kind-Indexed Types and Type-Indexed Values . . . . .          | 25 |
| 2.3 Embedding-Projection Maps . . . . .                           | 26 |
| 2.4 Reductions . . . . .  | 27 |
| 2.5 Pretty Printing . . . . .                                     | 30 |
| 2.6 Running Generic Haskell . . . . .                             | 33 |
| 3 Generic Haskell—Theory . . . . .                                | 35 |
| 3.1 The Simply Typed Lambda Calculus as a Type Language . . . . . | 35 |
| 3.2 The Polymorphic Lambda Calculus . . . . .                     | 40 |
| 3.3 Specializing Type-Indexed Values . . . . .                    | 42 |
| 3.4 Bridging the Gap . . . . .                                    | 45 |
| 4 Conclusion . . . . .  | 52 |
| Chapter 2. Generic Haskell: Applications . . . . .                | 57 |
| <i>R. Hinze and J. Jeuring</i>                                    |    |
| 1 Introduction . . . . .  | 57 |
| 2 Generic Dictionaries . . . . .                                  | 62 |
| 2.1 Introduction . . . . .  | 63 |
| 2.2 Signature . . . . .   | 65 |
| 2.3 Properties . . . . .  | 66 |
| 2.4 Type-Indexed Tries . . . . .                                  | 67 |
| 2.5 Empty Tries . . . . .   | 70 |
| 2.6 Singleton Tries . . . . .                                     | 71 |
| 2.7 Look up . . . . .   | 72 |
| 2.8 Inserting and Merging . . . . .                               | 73 |
| 2.9 Deleting . . . . .  | 75 |
| 2.10 Related Work . . . . .                                       | 76 |
| 3 XComprez: A Generic XML Compressor . . . . .                    | 77 |
| 3.1 Implementing an XML Compressor as a Generic Program . . . . . | 78 |
| 3.2 Analysis . . . . .  | 83 |
| 3.3 Conclusions . . . . .   | 84 |

|   |  |     |
|---|--|-----|
| 4   | The Zipper .....   | 85  |
| 4.1   | The Basic Idea .....   | 85  |
| 4.2   | Data Types as Fixed Points of Pattern Functors .....           | 86  |
| 4.3   | Type Indices of Higher Kinds .....                             | 86  |
| 4.4   | Locations .....  | 87  |
| 4.5   | Navigation Functions .....                                     | 88  |
| 5   | Conclusions .....  | 93  |
| Chapter 3. Generic Properties of Datatypes .....            |  | 97  |
| <i>R. Backhouse and P. Hoogendijk</i>                       |  |     |
| 1   | Introduction .....   | 97  |
| 2   | Theorems for Free .....  | 98  |
| 2.1   | Verifiable Genericity .....                                    | 102 |
| 3   | Commuting Datatypes — Introduction .....                       | 102 |
| 3.1   | Structure Multiplication .....                                 | 104 |
| 3.2   | Broadcasts .....   | 106 |
| 4   | Allegories and Relators .....                                  | 109 |
| 4.1   | Allegories .....   | 109 |
| 4.2   | Relators .....   | 110 |
| 4.3   | Composition and Relators Are Parametric .....                  | 112 |
| 4.4   | Division and Tabulation .....                                  | 113 |
| 4.5   | Domains .....  | 114 |
| 5   | Datatype = Relator + Membership .....                          | 114 |
| 5.1   | Pointwise Closure .....  | 114 |
| 5.2   | Regular Relators .....   | 116 |
| 5.3   | Natural Transformations .....                                  | 119 |
| 5.4   | Membership and Fans .....                                      | 121 |
| 6   | Commuting Datatypes — Formal Specification .....               | 125 |
| 6.1   | Naturality Requirements .....                                  | 126 |
| 6.2   | Composition .....  | 128 |
| 6.3   | Half Zips and Commuting Relators .....                         | 128 |
| 7   | Consequences .....   | 129 |
| 7.1   | Shape Preservation .....                                       | 129 |
| 7.2   | All Regular Datatypes Commute .....                            | 130 |
| 8   | Conclusion .....   | 131 |
| Chapter 4. Basic Category Theory for Models of Syntax ..... |  | 133 |
| <i>R.L. Crole</i>   |  |     |
| 1   | Introduction .....   | 133 |
| 1.1   | Prerequisites .....  | 133 |
| 1.2   | The Aims .....   | 134 |
| 1.3   | Learning Outcomes .....  | 134 |
| 2   | Syntax Defined from Datatypes .....                            | 134 |
| 2.1   | An Example with Distinguished Variables and without Binding .. | 135 |
| 2.2   | An Example with Distinguished Variables and Binding .....      | 136 |
| 2.3   | An Example with Arbitrary Variables and Binding .....          | 137 |
| 2.4   | An Example without Variables but with Binding .....            | 138 |

|  |   |     |
|--|---|-----|
| 3  | Category Theory .....   | 139 |
| 3.1  | Categories .....  | 139 |
| 3.2  | Functors .....  | 143 |
| 3.3  | Natural Transformations .....   | 144 |
| 3.4  | Products .....  | 146 |
| 3.5  | Coproducts .....  | 150 |
| 3.6  | Algebras .....  | 151 |
| 3.7  | The Functor $1 + (-): \mathbf{Set} \rightarrow \mathbf{Set}$ .....          | 152 |
| 3.8  | The Functor $A + (-): \mathbf{Set} \rightarrow \mathbf{Set}$ .....          | 153 |
| 3.9  | The Functor $1 + (A \times -): \mathbf{Set} \rightarrow \mathbf{Set}$ ..... | 154 |
| 4  | Models of Syntax .....  | 154 |
| 4.1  | A Model of Syntax with Distinguished Variables<br>and without Binding ..... | 157 |
| 4.2  | A Model of Syntax with Distinguished Variables<br>and with Binding .....    | 162 |
| 4.3  | A Model of Syntax with Arbitrary Variables and Binding .....                | 165 |
| 4.4  | A Model of Syntax without Variables but with Binding .....                  | 168 |
| 4.5  | Where to Now? .....   | 168 |
| 5  | Appendix .....  | 170 |
| 5.1  | Lists .....   | 170 |
| 5.2  | Abstract Syntax Trees .....   | 170 |
| 5.3  | Inductively Defined Sets .....  | 170 |
| 5.4  | Rule Induction .....  | 173 |
| 5.5  | Recursively Defined Functions .....   | 175 |
| Chapter 5. A Mathematical Semantics for Architectural Connectors ..... |   | 178 |
| <i>J.L. Fiadeiro, A. Lopes and M. Wermelinger</i>                      |   |     |
| 1  | Introduction .....  | 178 |
| 2  | System Configuration in CommUnity .....                                     | 180 |
| 2.1  | Component Design .....  | 180 |
| 2.2  | Configurations .....  | 184 |
| 3  | Architectural Description in CommUnity .....                                | 191 |
| 3.1  | Architectural Connectors .....  | 191 |
| 3.2  | Examples .....  | 196 |
| 4  | An ADL-Independent Notion of Connector .....                                | 201 |
| 4.1  | Architectural Schools .....   | 201 |
| 4.2  | Adding Abstraction to Architectural Connectors .....                        | 207 |
| 5  | Towards an Algebra of Connectors .....                                      | 213 |
| 5.1  | Role Refinement .....   | 214 |
| 5.2  | Role Encapsulation .....  | 216 |
| 5.3  | Role Overlay .....  | 218 |
| 6  | Concluding Remarks .....  | 219 |
| Author Index .....   |   | 223 |

Generic Programming

Advanced Lectures

Backhouse, R.; Gibbons, J. (Eds.)

2003, X, 230 p., Softcover

ISBN: 978-3-540-20194-6