

Patterns of Functional Data Processing

Business Application and Big Data

Ch 1 What is Data?

- Data, Information, and Representations
- A word on Typing
- Primitive Data Types in Practise
 - String: Encoding problem and other algorithmic considerations
 - Numbers: Precision, Exactness, Range, and the numeric tower
 - Symbols: Namespace and domain
 - Date and Time: Timezones and Subtlety of Human Calendar System
- Abstract Data Type
- JSON as the parsimonious universal data
 - Comparison with typed object
 - Schema vs Type
- Specialist Composite Data - Trees and Graphs
 - Loop and the infinite expansion problem
 - Embedding them in JSON
 - Separate Representation

Ch 2 A quick review of Functional Programming

Ch 3 General Patterns

Pipeline and Chaining
Derived Information and tagging
Information Completeness

Extraction, Aggregation, and Summarisation
Splitting, Merging and Matching

Navigating complex structures
Reshaping structures
Equivalent Representation

Explicit State Machine
Execution Trace/Log and Reproduction

Functional Encapsulation

Ch 4 Interfacing with the Real World

- API, contract, and evolving schema
- Type theory and schema compatibility
- Domain, bounded context, and their conversion
- Declarative vs Procedural conversion

Ch 5 Relational Database from a Functional Perspective

- Data as Stream
- Query and Evaluation Plan
- A unified view of data processing
 - System boundaries and its implications (performance, robustness)
- Now what?

Ch 6 Patterns in Big Data

- Where did Big Data come from?
- OLAP
- MapReduce
 - Programmer's Contract
 - How it works under the hood
- Online Stream Processing

Epilogue - What is a pattern?

The Original Design Pattern and its preverting

Cybernetics and Human-Driven Programming

Humans are flawed, but it is worth it

Lambda Calculus for the Working Programmer - A Gateway (drug) to Programming Language Theory

Preface - Why learn Lambda Calculus?

Ch 1 The under-appreciated power of Functions

What is a (pure) function anyway? Math and CS fights it out

Rigor in thinking - formalising syntax

The world's smallest programming language

Lambda is all you need - Church and Turing's Great Idea

(Y-combinator and recursion)

Citizens of Functions, Unite! Point-free style and combinator calculus

All paths lead to Rome - Reduction, non-determinism and normal forms

More?

Ch 2 Modelling a Programming Language - Yes you Can!

Order out of Chaos - Imposing evaluation order

Elements of Practical Modelling

Rigor in thinking - formalising semantics

Implementing a language

Ch 3 Disciplined Type - a pathway to Enlightenment

Holy War and the Wall of Tears - on War and Peace

The Lambda Cube

The Other Impedance Mismatch of Object Orientation

Program Insurance Inc. - the promise of dependent typing

Steam Engine and the Industrial Revolution - Hindley-Miller Type Inference

Modernism - Type Inference in Practice

Tech and Magic - Gradual Typing and Hope for the future

Ch 4 It is lambda all the way down - Parallel World in Computing History

A Machine made of lambda!

Low level vs High level

...

Ch 5 Going Further

Epilogue - But really, why learn Lambda Calculus?

A Tale of Three Models

The Nature of Technology and Progress

Democratisation in 21st century, Programming for the Masses, and a Note of Hope Plus Call to Action

Mathematics of Neural Network (TBD)