

Table of Contents

Unofficial Texinfo Format	10
Dedication	12
Foreword	13
Preface to the Second Edition	19
Preface to the First Edition	21
Acknowledgements	25
1 Building Abstractions with Procedures	29
1.1 The Elements of Programming	34
1.1.1 Expressions	35
1.1.2 Naming and the Environment	39
1.1.3 Evaluating Combinations	40
1.1.4 Compound Procedures	44
1.1.5 The Substitution Model for Procedure Application	47
1.1.6 Conditional Expressions and Predicates	51
1.1.7 Example: Square Roots by Newton's Method	57
1.1.8 Procedures as Black-Box Abstractions	63
1.2 Procedures and the Processes They Generate	71
1.2.1 Linear Recursion and Iteration	72
1.2.2 Tree Recursion	79

1.2.3	Orders of Growth	86
1.2.4	Exponentiation	89
1.2.5	Greatest Common Divisors	95
1.2.6	Example: Testing for Primality	98
1.3	Formulating Abstractions with Higher-Order Procedures	107
1.3.1	Procedures as Arguments	109
1.3.2	Constructing Procedures Using Lambda	116
1.3.3	Procedures as General Methods	122
1.3.4	Procedures as Returned Values	129

2 Building Abstractions with Data 139

2.1	Introduction to Data Abstraction	144
2.1.1	Example: Arithmetic Operations for Rational Numbers	145
2.1.2	Abstraction Barriers	151
2.1.3	What Is Meant by Data?	155
2.1.4	Extended Exercise: Interval Arithmetic	159
2.2	Hierarchical Data and the Closure Property	165
2.2.1	Representing Sequences	168
2.2.2	Hierarchical Structures	180
2.2.3	Sequences as Conventional Interfaces	188
2.2.4	Example: A Picture Language	208
2.3	Symbolic Data	229
2.3.1	Quotation	229
2.3.2	Example: Symbolic Differentiation	234
2.3.3	Example: Representing Sets	241
2.3.4	Example: Huffman Encoding Trees	255
2.4	Multiple Representations for Abstract Data	267
2.4.1	Representations for Complex Numbers	270

2.4.2	Tagged data	276
2.4.3	Data-Directed Programming and Additivity	283
2.5	Systems with Generic Operations	295
2.5.1	Generic Arithmetic Operations	297
2.5.2	Combining Data of Different Types	304
2.5.3	Example: Symbolic Algebra	316

3 Modularity, Objects and State 338

3.1	Assignment and Local State	340
3.1.1	Local State Variables	341
3.1.2	The Benefits of Introducing Assignment	350
3.1.3	The Costs of Introducing Assignment	357
3.2	The Environment Model of Evaluation	366
3.2.1	The Rules for Evaluation	369
3.2.2	Applying Simple Procedures	373
3.2.3	Frames as the Repository of Local State	377
3.2.4	Internal Definitions	384
3.3	Modeling with Mutable Data	388
3.3.1	Mutable List Structure	389
3.3.2	Representing Queues	402
3.3.3	Representing Tables	410
3.3.4	A Simulator for Digital Circuits	419
3.3.5	Propagation of Constraints	437
3.4	Concurrency: Time Is of the Essence	454
3.4.1	The Nature of Time in Concurrent Systems	456
3.4.2	Mechanisms for Controlling Concurrency	463
3.5	Streams	484
3.5.1	Streams Are Delayed Lists	485

3.5.2	Infinite Streams	498
3.5.3	Exploiting the Stream Paradigm.....	509
3.5.4	Streams and Delayed Evaluation	527
3.5.5	Modularity of Functional Programs and Modularity of Objects	536

4 Metalinguistic Abstraction 545

4.1	The Metacircular Evaluator	549
4.1.1	The Core of the Evaluator	552
4.1.2	Representing Expressions	559
4.1.3	Evaluator Data Structures	571
4.1.4	Running the Evaluator as a Program	577
4.1.5	Data as Programs.....	583
4.1.6	Internal Definitions	588
4.1.7	Separating Syntactic Analysis from Execution	597
4.2	Variations on a Scheme – Lazy Evaluation	604
4.2.1	Normal Order and Applicative Order	605
4.2.2	An Interpreter with Lazy Evaluation	608
4.2.3	Streams as Lazy Lists	620
4.3	Variations on a Scheme – Nondeterministic Computing.....	624
4.3.1	Amb and Search.....	627
4.3.2	Examples of Nondeterministic Programs	634
4.3.3	Implementing the Amb Evaluator.....	645
4.4	Logic Programming	663
4.4.1	Deductive Information Retrieval	668
4.4.2	How the Query System Works	684
4.4.3	Is Logic Programming Mathematical Logic?	696
4.4.4	Implementing the Query System.....	705