

Contents

Foreword by Hal Abelson	ix
Preface	xv
Acknowledgments	xxi
1 Inductive Sets of Data	1
1.1 Recursively Specified Data	1
1.2 Deriving Recursive Programs	12
1.3 Auxiliary Procedures and Context Arguments	22
1.4 Exercises	25
2 Data Abstraction	31
2.1 Specifying Data via Interfaces	31
2.2 Representation Strategies for Data Types	35
2.3 Interfaces for Recursive Data Types	42
2.4 A Tool for Defining Recursive Data Types	45
2.5 Abstract Syntax and Its Representation	51
3 Expressions	57
3.1 Specification and Implementation Strategy	57
3.2 LET: A Simple Language	60
3.3 PROC: A Language with Procedures	74
3.4 LETREC: A Language with Recursive Procedures	82
3.5 Scoping and Binding of Variables	87
3.6 Eliminating Variable Names	91
3.7 Implementing Lexical Addressing	93

4	State	103
4.1	Computational Effects	103
4.2	EXPLICIT-REFS: A Language with Explicit References	104
4.3	IMPLICIT-REFS: A Language with Implicit References	113
4.4	MUTABLE-PAIRS: A Language with Mutable Pairs	124
4.5	Parameter-Passing Variations	130
5	Continuation-Passing Interpreters	139
5.1	A Continuation-Passing Interpreter	141
5.2	A Trampoline Interpreter	155
5.3	An Imperative Interpreter	160
5.4	Exceptions	171
5.5	Threads	179
6	Continuation-Passing Style	193
6.1	Writing Programs in Continuation-Passing Style	193
6.2	Tail Form	203
6.3	Converting to Continuation-Passing Style	212
6.4	Modeling Computational Effects	226
7	Types	233
7.1	Values and Their Types	235
7.2	Assigning a Type to an Expression	238
7.3	CHECKED: A Type-Checked Language	240
7.4	INFERRED: A Language with Type Inference	248
8	Modules	275
8.1	The Simple Module System	276
8.2	Modules That Declare Types	292
8.3	Module Procedures	311
9	Objects and Classes	325
9.1	Object-Oriented Programming	326
9.2	Inheritance	329
9.3	The Language	334
9.4	The Interpreter	336
9.5	A Typed Language	352
9.6	The Type Checker	358

A For Further Reading	373
B The SLLGEN Parsing System	379
B.1 Scanning	379
B.2 Parsing	382
B.3 Scanners and Parsers in SLLGEN	383
Bibliography	393
Index	401