

Contents

Foreword	vii
Preface	xi
Acknowledgments	xvii
1 Inductive Sets of Data	1
1.1 Recursively Specified Data	1
1.2 Recursively Specified Programs	9
1.3 Scoping and Binding of Variables	28
2 Data Abstraction	39
2.1 Specifying Data via Interfaces	39
2.2 An Abstraction for Inductive Data Types	42
2.3 Representation Strategies for Data Types	55
2.4 A Queue Abstraction	66
3 Environment-Passing Interpreters	69
3.1 A Simple Interpreter	71
3.2 The Front End	75
3.3 Conditional Evaluation	80
3.4 Local Binding	81
3.5 Procedures	84
3.6 Recursion	92
3.7 Variable Assignment	98
3.8 Parameter-Passing Variations	107
3.9 Statements	120

4	Types	125
4.1	Typed Languages	125
4.2	Type Checking	132
4.3	Enforcing Abstraction Boundaries	143
4.4	Type Inference	152
5	Objects and Classes	169
5.1	Object-Oriented Programming	171
5.2	Inheritance	173
5.3	The Language	179
5.4	Four implementations	183
6	Objects and Types	205
6.1	A Simple Typed Object-Oriented Language	205
6.2	The Type Checker	211
6.3	The Translator	229
7	Continuation-Passing Interpreters	241
7.1	A Continuation-Passing Interpreter	243
7.2	Procedural Representation of Continuations	261
7.3	An Imperative Interpreter	264
7.4	Exceptions and Control Flow	277
7.5	Multithreading	284
7.6	Logic Programming	295
8	Continuation-Passing Style	301
8.1	Tail Form	302
8.2	Converting to Continuation-Passing Style	308
8.3	Examples of the CPS Transformation	317
8.4	Implementing the CPS Transformation	327
8.5	Modeling computational effects	338
A	The SLLGEN Parsing System	345
B	For Further Reading	359
	Bibliography	361
	Index	367