1. Introduction 1.1
1.1. Interlisp as a Programming Language 1.1
1.2. Interlisp as an Interactive Environment 1.3
1.3. Interlisp Philosophy 1.5
1.4. How to Use this Manual 1.7
1.5. References 1.8
2. Litatoms 2.1
2.1. Using Litatoms as Variables 2.2
2.2. Function Definition Cells 2.5
2.3. Property Lists 2.5
2.4. Print Names 2.7
2.5. Characters and Character Codes 2.12
3. Lists 3.1
3.1. Creating Lists 3.4
3.2. Building Lists From Left to Right 3.6
3.3. Copying Lists 3.8
3.4. Extracting Tails of Lists 3.9
3.5. Counting List Cells 3.10
3.6. Logical Operations 3.11
3.7. Searching Lists 3.12
3.8. Substitution Functions 3.13
3.9. Association Lists and Property Lists 3.15
3.10. Sorting Lists 3.17
3.11. Other List Functions 3.19
4. Strings 4.1
5. Arrays 5.1
6. Hash Arrays 6.1
6.1. Hash Overflow 6.3

6.2. User-Specified Hashing Functions 6.4

7. Numbers and Arithmetic Functions 7.1

7.1. Generic Arithmetic 7.3

7.2. Integer Arithmetic 7.4

7.3. Logical Arithmetic Functions 7.8

7.4. Floating Point Arithmetic 7.11

7.5. Other Arithmetic Functions 7.13

8. Record Package 8.1

8.1. FETCH and REPLACE 8.2

8.2. CREATE 8.3

8.3. TYPE? 8.5

8.4. WITH 8.5

8.5. Record Declarations 8.6

8.5.1. Record Types 8.7

8.5.2. Optional Record Specifications 8.14

8.6. Defining New Record Types 8.15

8.7. Record Manipulation Functions 8.16

8.8. Changetran 8.17

8.9. Built-In and User Data Types 8.20

9. Conditionals and Iterative Statements 9.1

9.1. Data Type Predicates 9.1

9.2. Equality Predicates 9.2

9.3. Logical Predicates 9.3

9.4. The COND Conditional Function 9.4

9.5. The IF Statement 9.5

9.6. Selection Functions 9.6

9.7. PROG and Associated Control Functions 9.7

9.8. The Iterative Statement 9.9

9.8.1. I.s.types 9.10

9.8.2. Iteration Variable I.s.oprs 9.12

9.8.3. Condition I.s.oprs 9.15

9.8.4. Other I.s.oprs 9.16

9.8.5. Miscellaneous Hints on I.S.Oprs 9.17

9.8.6. Errors in Iterative Statements 9.19

9.8.7. Defining New Iterative Statement Operators 9.20

10. Function Definition, Manipulation, and Evaluation 10.1

10.1. Function Types 10.2

10.1.1. Lambda-Spread Functions 10.3

10.1.2. Nlambda-Spread Functions 10.4

10.1.3. Lambda-Nospread Functions 10.5

10.1.4. Nlambda-Nospread Functions 10.6

10.1.5. Compiled Functions 10.6

10.1.6. Function Type Functions 10.6

10.2. Defining Functions 10.9

10.3. Function Evaluation 10.11

10.4. Iterating and Mapping Functions 10.14

10.5. Functional Arguments 10.18

10.6. Macros 10.21

10.6.1. DEFMACRO 10.24

10.6.2. Interpreting Macros 10.28

11. Variable Bindings and the Interlisp Stack 11.1

11.1. The Spaghetti Stack 11.2

11.2. Stack Functions 11.4

11.2.1. Searching the Stack 11.5

11.2.2. Variable Bindings in Stack Frames 11.6

11.2.3. Evaluating Expressions in Stack Frames 11.7

11.2.4. Altering Flow of Control 11.8

11.2.5. Releasing and Reusing Stack Pointers 11.9

11.2.6. Backtrace Functions 11.11

11.2.7. Other Stack Functions 11.13

11.3. The Stack and the Interpreter 11.14

11.4. Generators 11.16

11.5. Coroutines 11.18

11.6. Possibilities Lists 11.20

12. Miscellaneous 12.1

12.1. Greeting and Initialization Files 12.1

12.2. Idle Mode 12.4

12.3. Saving Virtual Memory State 12.6

12.4. System Version Information 12.11

12.5. Date And Time Functions 12.13

12.6. Timers and Duration Functions 12.16

12.7. Resources 12.19

12.7.1. A Simple Example 12.20

12.7.2. Trade-offs in More Complicated Cases 12.22

12.7.3. Macros for Accessing Resources 12.23

12.7.4. Saving Resources in a File 12.23

12.8. Pattern Matching 12.24

12.8.1. Pattern Elements 12.25

12.8.2. Element Patterns 12.25

12.8.3. Segment Patterns 12.27

12.8.4. Assignments 12.28

12.8.5. Place-Markers 12.29

12.8.6. Replacements 12.29

12.8.7. Reconstruction 12.30

12.8.8. Examples 12.31