

目 次

まえがき.....	iii
謝 辞.....	vii
訳者序文.....	viii

第 I 部 進歩的 LISP プログラミング

第 1 章 LISP 総論	3
1.1 データ構造	3
1.2 プログラムの構造	4
1.3 S 式に対する基本演算	11
1.4 木 構 造	13
1.5 リ ス ト	16
1.6 マップ関数	18
1.7 ラムダ式	19
1.8 ア ト ム	21
1.9 LISP の内部	23
1.10 同 等 性	28
1.11 局所変数対自由変数	29
1.12 LISP 書法	30
1.12.1 段 付 け	30
1.12.2 名 前 付 け	31
1.12.3 注 釈	32
1.12.4 関数の大きさ	32
1.12.5 功 妙 さ	32
第 2 章 FEXPR と LEXPR	34
2.1 予約語の定義	34
2.2 不特定個の引数をもつ関数の定義	36

2.3	FEXPR を使う時の問題点	37
2.4	INTERLISP における FEXPR	40
第 3 章	マクロと読み込みマクロ	42
3.1	マクロ	42
3.2	読み込みマクロ	47
3.3	[*] 読み込みマクロ	50
3.4	インタプリタにおけるマクロ展開	53
3.5	FEXPR によるマクロのシミュレート	56
第 4 章	データ型の定義	57
4.1	データ型の必要性	57
4.2	型の定義に対する保守的手法	59
4.3	進歩的手法	61
4.4	革新的手法	64
第 5 章	評価順序を制御する関数	66
5.1	繰返し	66
5.2	基本的な LOOP マクロ	68
5.3	LOOP の構文と意味	69
5.4	LOOP の定義	70
5.5	LOOP の例	73
5.6	マップ関数と FOR マクロ	74
5.7	FOR マクロの例	76
5.8	FOR マクロの定義	77
5.9	結論	81
第 6 章	LISP における入出力	83
6.1	文字ストリングと S 式	84
6.2	LISP における読み込み	86
6.3	LISP における印刷	86
6.4	WRITE マクロ	89
6.5	LISP におけるファイル	92
6.6	UCI-LISP におけるファイルの入力	92

6.7	UCI-LISP におけるファイルの出力	96
6.8	定義される関数と入出力との分離	98
第7章	LISP 式の編集	102
7.1	イン・コア編集	102
7.2	編集の例	103
7.3	編集命令関数	106
7.4	内部関数定義	107
7.5	命令関数の定義	111
7.6	作業領域での編集についての最後の注意	117
第8章	コンパイル	119
8.1	なぜコンパイルするのか?	119
8.2	コンパイラがすること	120
8.3	コンパイル: コードにおける変数	122
8.4	EVAL と SET に対するコンパイルの影響	124
8.5	EXPR 以外のコンパイル	125
8.6	結 論	126
第9章	データ駆動型プログラミング	128
9.1	はじめに	128
9.2	: = マクロ	129
9.3	組織化の手段としてのデータ駆動型プログラミング	132
第10章	いろいろなプログラミング技法	134
10.1	リスト上の集合演算	134
10.2	ヘッダー付きリストとキュー	136
10.3	ハッシュ表	140

第II部 AI プログラミング技法

第11章	単純な弁別ネット	147
11.1	一般的な弁別ネット	147
11.2	データベース弁別ネット——アトムのリスト	150

11.3	データベース弁別ネット——一般のS式	152
第12章	アジェンダ制御構造	159
12.1	最適優先木探索	159
12.2	コルーティンとアジェンダ	163
12.3	ストリーム	165
第13章	推論による情報検索	170
13.1	述語論理	170
13.2	推論による検索	173
13.3	単一化アルゴリズム	177
13.4	一つの推論検索システム	181
13.5	他の話題	185
13.5.1	前向き推論	185
13.5.2	含意の検索	187
13.5.3	AI言語	188
13.5.4	パタン照合	192
13.6	推論情報検索に対する賛否両論	194
第14章	変数を持つ弁別ネット	197
14.1	プランの検索	197
14.2	事実の取り出し	201
14.3	弁別ネットのいろいろ	205
第15章	スロットとその収納値によるデータベース	215
15.1	属性リストの拡張	215
15.2	XRL 入門	216
15.3	XRL におけるパターン照合	222
15.4	FORM の検索	227
15.5	さらに FORM の照合について	231
第16章	データ依存関係	234
16.1	データ依存関係の必要性	234
16.2	データベースの論理式の消去	235

16.3	循環証明の扱い	238
16.4	単調でない依存関係	243
16.5	データ依存関係の応用	253
16.5.1	証明を見つける	253
16.5.2	説明を行う	255
16.5.3	依存関係に基づいたバックトラック	257
16.5.4	時の表現	262
16.6	データベースの使用についてのアドバイス	266
16.7	補遺——DBGC の正しさ	272
第 17 章	別の制御構造	276
17.1	はじめに	276
17.2	SCHUM の設計	277
17.3	SCHUM によるプログラミング	286
17.4	SCHUM のインプリメント	294
17.5	インタプリタ	297
第 18 章	時間関係に基づいたバックトラック	313
18.1	バックトラック無しの基本的遷移ネットワーク文法	314
18.2	拡張遷移ネットワーク文法	319
18.3	バックトラック付きの ATN——状態保存法	321
18.4	遷移保存法によるバックトラック	328
18.5	SCHUM を用いたバックトラック	329
18.6	問題解決技法の利用	334
第 III 部 プロジェクト例		
第 19 章	Tale-Spin の考え	345
19.1	物語の生成	347
19.2	事象の表現	347
19.3	物語生成の例	350
19.4	物語生成の失敗	352
第 20 章	Tale-Spin の詳細	355
20.1	物語の断片の表現	355

20.2	基本ループ	359
20.2.1	ゴール・モニタ	359
20.2.2	物語の断片モニタ	360
20.3	物語の失敗	362
20.4	バックトラック	363
20.5	読者用の問題	363
補遺：LISP 関数と予約語		367
参考文献		381
索引		385

