

Table of Contents

PART 0. Introduction and Historic Overview	1
PART I. Grammars for Formal Languages and Linguistic Research	5
Chapter 1. What are Logic Grammars	5
1. Logic Grammars – Basic Features	5
2. Grammar Symbols	5
3. Use of Variables and Unification	7
3.1. Unification	7
3.2. Derivation Graphs	9
3.3. Symbol Arguments: Producers and Consumers of Structure	10
4. Tests within Rules	11
5. Operators	12
6. Analysis and Generation	13
7. A Formal Language Example	14
Chapter 2. Relation of Logic Grammars to Prolog	17
1. Basic Prolog Concepts	17
2. Prolog as a Language Recognizer	19
3. Prolog as a Structure Builder	20
Bibliographic Commentary for Part I	23
PART II. Getting Started: How to Write a Simple Logic Grammar	25
Chapter 3. Step-by-Step Development of a Natural Language Analyser	25
1. Propositional Statements	25
2. Obtaining Representations for Propositional Statements	27
3. Syntactic and Semantic Agreement	30
4. Noun Phrases	31
5. Negative Sentences	36
6. Interrogative Clauses	39
Chapter 4. Choosing Internal Representations for Natural Language	43

1. Logical Form for Querying Knowledge Represented in Logic	43
2. First-Order Logic Representations	45
2.1. Semantic Well Formedness	46
2.2. Ambiguity	46
2.3. Natural Language Determiners and Modifiers	46
2.3.1. Three-Branched Quantification	47
2.3.2. Quantifier Scoping	52
2.4. Negation	53
2.5. Meaning of Plural Forms	54
2.6. Representing Sets	55
3. Lambda-Calculus Representations	55
Chapter 5. Developing a Logic Grammar for a Formal Application	61
1. An Analyzer for Logic Programs	61
2. Simple Lexical Analysis	64
3. Structural Representation of a Logic Program	67
4. "Compiling" Logic Programs	69
5. Compiling Proof Tree Generation into Rules	70
Bibliographic Commentary for Part II	75
 PART III. Different Types of Logic Grammars – What Each Is Useful For	77
Chapter 6. Basic Types of Logic Grammars	77
1. Metamorphosis Grammars	78
2. Definite Clause Grammars	79
3. Extraposition Grammars	81
4. Discussion	82
Chapter 7. Building Structure	85
1. Parse Tree Construction	85
2. Meaning Representation Buildup	86
3. Automating Syntactic and Semantic Structure Buildup	88
Chapter 8. Modifier Structure Grammars	91
1. Separation of Syntax and Semantics	91
2. Quantifier Rescoping	91
3. Coordination	92
4. Implementation	93
5. Discussion	93
Chapter 9. Definite Clause Translation Grammars and their Applications	95

1. Definite Clause Translation Grammars	95
2. A Compiler	109
2.1. Introduction	109
2.2. Lexical Analysis	111
2.3. Between Lexical and Syntactic Analysis	113
2.4. Syntactic Analysis	115
2.5. Code Generation	116
2.5.1. The Target Machine	117
2.5.2. Code Generated for Statements	119
2.5.3. Code for Expressions	122
2.6. Assembly and Allocation	126
3. A Metagrammatical Extension of DCTG Notation	128
4. Grammatical Data Typing	135
4.1. The Natural Numbers	135
4.2. Lists	137
4.3. Trees	139
4.4. Infix and Prefix Notation	139
4.5. Comments on Grammatical Typing	140
Chapter 10. Further Expressive Power – Discontinuous Grammars	143
1. The Discontinuous Grammar Family	143
2. Thinking in Terms of Skips – Some Examples	145
2.1. Coordination	145
2.2. Right Extraposition	146
2.3. Interaction between Different Discontinuous Rules	149
2.4. Avoiding Artifices through Straightforward Uses of Skips	149
2.5. Rules with More Than One Skip	150
2.6. DGs and Free Word Order Languages	150
2.6.1. Totally Free Word or Constituent Order	150
2.6.2. Lexically Induced Rule Format and Free Word Order	151
3. Static Discontinuity Grammars and Government-Binding Theory	153
3.1. Rendering Context-Free Simplicity with Type-0 Power	153
3.2. Government-Binding-Oriented Constraints	155
3.3. Definition	157
3.4. Implementation Considerations	158
3.5. Transporting the Static Discontinuity Feature into Logic Programming	158
Bibliographic Commentary for Part III	161

PART IV. Other Applications	165
Chapter 11. Other Formalisms	165
1. Restriction Grammars	165
2. Puzzle Grammars	167
3. Discussion	169
Chapter 12. Bottom Up Parsing	171
1. Introduction	171
2. Compiling Context-Free Rules	172
Bibliographic Commentary for Part IV	177
 PART V. Logic Grammars and Concurrency	179
Chapter 13. Parsing with Commitment	179
1. Introduction	179
1.1. Sample Grammar	182
1.2. The One Character Lookahead Relation	182
2. Compilation to Sequential Logic Programs	183
3. Compilation to Concurrent Logic Program Clauses	185
4. Generalized Deterministic Grammars	187
5. Related Work	188
6. Parallel Parsing for Natural Language Analysis	189
6.1. Sample Grammar	189
6.2. Parallel Parsing Method	191
Bibliographic Commentary for Part V	196
 Appendices	197
I. Input/Output in Logic Grammars	197
1. Input of a Sentence	197
II. Implementation of Logic Grammar Formalisms	199
1. SYNAL: Compiling Disc. Grammars to Prolog	199
2. A Short Interpreter for Static Discontinuity Grammars	201
3. Implementations of DCTGs	203
3.1. An Interpreter for Definite Clause Translation Grammars	203
3.2. Compiling Definite Clause Translation Grammars to Prolog	205
3.3. Typing DCTGs	212
 LITERATURE	215
INDEX	225