

# Contents

<i>Preface</i>	v
<i>List of Figures</i>	xiii
<b>Introduction and Preliminaries</b>	<b>1</b>
1. Introduction	3
1.1 Motivation . . . . .	3
1.2 Background . . . . .	5
1.2.1 Parsing natural language . . . . .	5
1.2.2 Robustness in grammar-driven parsers . . . . .	7
1.2.3 Parsing schemata . . . . .	8
1.3 Outline of the book . . . . .	9
1.3.1 Contributions . . . . .	10
1.3.2 Structure of the book . . . . .	10
2. Preliminaries	15
2.1 Context-free grammars . . . . .	15
2.2 Parsing algorithms and schemata . . . . .	18
2.3 The formalism of parsing schemata . . . . .	25
2.3.1 Deduction systems . . . . .	25
2.3.2 Parsing systems and parsing schemata . . . . .	26
2.3.3 Correctness of parsing schemata . . . . .	30
2.3.4 Relations between parsing schemata . . . . .	30
2.4 Advantages of parsing schemata . . . . .	32

<b>Compiling and Executing Parsing Schemata</b>	<b>35</b>
3. A compiler for parsing schemata	37
3.1 Motivation and goals . . . . .	37
3.1.1 Design goals . . . . .	39
3.1.2 Related work . . . . .	40
3.2 System architecture . . . . .	41
3.3 Generated code . . . . .	42
3.4 Reading schemata . . . . .	44
3.5 The code generation process . . . . .	48
3.5.1 Element types . . . . .	48
3.5.2 Deduction step classes . . . . .	50
3.5.3 Deduction step code generation . . . . .	51
3.5.4 Search specifications . . . . .	52
3.6 Indexing . . . . .	53
3.6.1 Static analysis and index descriptors . . . . .	54
3.6.2 Generation of indexing code . . . . .	57
3.6.3 Deduction step indexing . . . . .	59
3.7 Discussion . . . . .	60
4. Practical complexity of constituency parsers	63
4.1 Parsing natural language with CFGs . . . . .	64
4.2 Parsing with TAGs . . . . .	70
4.2.1 Tree-adjointing grammars . . . . .	70
4.2.2 Substitution and adjunction . . . . .	73
4.2.3 Properties of TAG . . . . .	75
4.3 Parsing schemata for TAG . . . . .	76
4.4 Parsing schemata for the XTAG English grammar . . . . .	78
4.4.1 Grammar conversion . . . . .	79
4.4.2 Feature structure unification . . . . .	80
4.4.3 Tree filtering . . . . .	83
4.5 Comparing several parsers for the XTAG grammar . . . . .	86
4.6 Parsing with artificially-generated TAGs . . . . .	89
4.7 Overhead of TAG parsing over CFG parsing . . . . .	96
4.8 Discussion . . . . .	99

## Parsing Schemata for Error-Repair Parsers 101

5. Error-repair parsing schemata	103
5.1 Motivation . . . . .	103
5.2 Error repair in parsing schemata . . . . .	104
5.2.1 The formalism of error-repair parsing schemata . .	105
5.2.2 A tree distance for edit distance-based repair . . .	108
5.3 Lyon's error-repair parser . . . . .	111
5.3.1 Lyon is correct . . . . .	113
5.4 Obtaining minimal distance parses . . . . .	121
5.5 Global and regional error repair . . . . .	124
5.5.1 Performance of global and regional parsers . . . .	128
5.6 Discussion . . . . .	129
6. Transforming standard parsers into error-repair parsers	131
6.1 From standard parsers to error-repair parsers . . . . .	131
6.1.1 Outline of the transformation . . . . .	132
6.2 Formal description of the error-repair transformation . . .	135
6.2.1 Some properties of trees and items . . . . .	135
6.2.2 Some properties of deduction steps . . . . .	137
6.2.3 The error-repair transformation . . . . .	139
6.3 Proof of correctness of the error-repair transformation . .	143
6.3.1 Proof of Theorem 6.1 . . . . .	145
6.3.2 Proof of Theorem 6.2 . . . . .	148
6.4 Optimising the results of the transformation . . . . .	155
6.5 Discussion . . . . .	158

## Parsing Schemata for Dependency Parsers 161

7. Dependency parsing schemata	163
7.1 Motivation . . . . .	163
7.2 The formalism of dependency parsing schemata . . . . .	165
7.3 Parsing schemata for projective dependency parsers . . . .	169
7.3.1 Collins (1996) . . . . .	169
7.3.2 Eisner (1996) . . . . .	171
7.3.3 Eisner and Satta (1999) . . . . .	172
7.3.4 Yamada and Matsumoto (2003) . . . . .	173

7.3.5	Lombardo and Lesmo (1996) and other Earley-based parsers . . . . .	174
7.3.6	Nivre (2003) . . . . .	176
7.3.7	Covington's projective parser (Covington, 2001) . . . . .	180
7.4	Relations between dependency parsers . . . . .	180
7.4.1	Yamada and Matsumoto (2003) $\xrightarrow{sr}$ Eisner (1996) . . . . .	181
7.4.2	Eisner and Satta (1999) $\xrightarrow{sr}$ Eisner (1996) . . . . .	182
7.4.3	Other relations . . . . .	183
7.5	Proving the correctness of dependency parsers . . . . .	184
7.5.1	Eisner and Satta (1999) is correct . . . . .	184
7.5.2	Yamada and Matsumoto (2003) is correct . . . . .	185
7.5.3	Eisner (1996) is correct . . . . .	186
7.6	Parsing schemata for non-projective dependency parsers . . . . .	186
7.6.1	Pseudo-projectivity . . . . .	187
7.6.2	Attardi (2006) and the $MH_k$ parser . . . . .	187
7.6.3	MST parser (McDonald <i>et al.</i> , 2005b) . . . . .	190
7.6.4	Covington's non-projective parser (Covington, 1990;2001) . . . . .	192
7.7	Parsing schemata for Link Grammar parsers . . . . .	193
7.7.1	Sleator and Temperley's LG parser . . . . .	196
7.7.2	Adapting projective dependency parsers to LG . . . . .	198
7.7.3	Eisner (1996) for LG . . . . .	200
7.7.4	Eisner and Satta (1999) for LG . . . . .	201
7.7.5	Yamada and Matsumoto (2003) for LG . . . . .	203
7.8	Discussion . . . . .	204
8.	Mildly non-projective dependency parsing . . . . .	207
8.1	Motivation . . . . .	207
8.2	Preliminaries . . . . .	209
8.3	The $WG_1$ parser . . . . .	211
8.3.1	$WG_1$ parsing schema . . . . .	211
8.3.2	Proof of correctness for $WG_1$ . . . . .	214
8.3.3	Computational complexity of $WG_1$ . . . . .	226
8.4	The $WG_k$ parser . . . . .	227
8.4.1	$WG_k$ parsing schema . . . . .	227
8.4.2	Proof of correctness for $WG_k$ . . . . .	229
8.4.3	Computational complexity of $WG_k$ . . . . .	230
8.5	Parsing ill-nested structures . . . . .	231

8.5.1	The $MG_1$ and $MG_k$ parsers . . . . .	231
8.5.2	Complexity of $MG_k$ . . . . .	234
8.5.3	Proof of correctness for $MG_k$ . . . . .	234
8.5.4	Mildly ill-nested dependency structures . . . . .	241
8.6	Discussion . . . . .	243
<b>Conclusion</b>		<b>245</b>
9.	Conclusions	247
9.1	Future work . . . . .	250
	<i>List of Acronyms</i>	253
	<i>Bibliography</i>	255
	<i>Index</i>	269