

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

231

Roland Hausser

NEWCAT: Parsing
Natural Language Using
Left-Associative Grammar



Springer-Verlag

Berlin Heidelberg New York London Paris Tokyo

Editorial Board

D. Barstow W. Brauer P. Brinch Hansen D. Gries D. Luckham
C. Moler A. Pnueli G. Seegmüller J. Stoer N. Wirth

Author

Roland Hausser
Institut für Deutsche Philologie, Universität München
Schellingstraße 3, 8000 München 40, FRG

CR Subject Classifications (1985): D.3.1, F.4.2, I.2.7, H.2.3, H.3.1, J.1.5

ISBN 3-540-16781-1 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-16781-1 Springer-Verlag New York Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to "Verwertungsgesellschaft Wort", Munich.

© Springer-Verlag Berlin Heidelberg 1986
Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2145/3140-543210

Preface

The verb *to parse* means “to describe grammatically by stating the part of speech and explaining the syntactical relationship” (Webster’s New Collegiate Dictionary). The noun *parser* refers to computer programs which grammatically analyze sentences or text of a language. Parser programs have been written for both formal languages (programming languages)¹ and natural languages (e. g. English or German).

Natural language parsers are a precondition of comfortable man-machine communication. Automatic speech recognition, data base interfaces, machine translation, and a host of other important applications require efficient natural language parsers. For this reason natural language parsing has always been a primary goal of non-numeric programming.

The construction of natural language parsers is an interdisciplinary enterprise, requiring the cooperation of linguists and computer scientists. This cooperation is characterized by a convenient division of labor. The linguists take pride in basing their grammars solely on linguistic grounds, such as natural language “universals”. Whether or not their grammar is suitable for parsing programs is not considered an issue. The computer scientists, on the other hand, take pride in their ability to implement any grammar as a computer program as long as the grammar is a reasonably explicit formalism. How a grammar is implemented on a computer is considered irrelevant as long as the program runs reasonably fast, and the display of the output closely resembles the syntactic representations envisioned by the linguist.

However, despite great efforts for over thirty years, the parsing of natural language is still an unsolved mystery. There are many different parsing algorithms, each with its own merits and limitations. But somehow the structures found in natural language do not seem amenable to a general and efficient analysis with existing parsing programs. This is taken by many people as evidence that it is simply impossible to build computers which analyze (and understand) natural language with the ease and efficiency of a native speaker.

Why is the computational analysis of natural language such a difficult task? Is natural language or the theoretical approach at fault? So far the widely accepted separation of the “declarative” (grammatical) and the “procedural” (computational) aspects of parsing has prevented the investigating of whether contemporary formal grammars of natural language provide a suitable basis for parsing programs.

In this book it is shown that constituent structure analysis, predominant in today’s grammars, induces an irregular order of linear composition which is the direct cause of extreme computational inefficiency. An alternative left-associative grammar is proposed, which operates with a regular order of linear compositions. Left-associative grammar is based on building up and cancelling valencies. Left-associative parsers differ from all other systems in that the history of the parse doubles as the linguistic analysis. The efficiency and descriptive power of left-associative grammar is illustrated with two left-associative natural language parsers: one for German and one for English.

Munich/Stanford, May 1986

R. Hausser

¹ For conversion of higher level statements into assembly or machine language in compilers.

Acknowledgements

The research for this book was supported by a Heisenberg grant from the Deutsche Forschungsgemeinschaft, West Germany. The parser programs were written during two 3 month stays at the Center for the Study of Language and Information, Stanford University, in 1984 and 1985. The German parser NEW-CAT (a name derived from 'NEW CATegorial approach') was first described in the CSLI publication IN-CSLI-85-5 of December 1985.

Conceptually NEWCAT is based on many years of linguistic research in syntax and semantics which would have remained dormant in the form of paper and pencil studies without the opportunity to work with the computing facilities and the people maintaining them at CSLI. I would like to thank Betsy Macken, John Perry, and Stanley Peters for sponsoring my stays there.

At various stages in the development of the programs, I received help from people who were or still are working at CSLI. Doug Cutting, Frederic Vander Elst, Mike Moore, Atty Mullins, Paul Oppenheimer, and Greep (alias Steven Tepper) spent long hours figuring out what I wanted and how to write it in LISP. Brad Horak and Joe Zingheim maintained the dandytiger in my office in top running condition. Emma Pease helped me with formatting this book in Latex. David Brown, Marjorie Maxwell, and Susi Parker helped me with the practical aspects of life at CSLI.

I benefitted especially from discussions with Tryg Ager, Pentti Kanerva, Ron Kaplan, Martin Kay, Joachim Laubsch, Eric Ostrom, Carl Pollard, Stuart Shieber, and Hans Uszkoreit. Steven Tepper and Theo Vennemann made detailed comments on the semifinal draft. I am indebted to Dikran Kargueuzian for help and advice on several occasions. Last, but not least, I would like to thank Deborah Kerman for proofreading the book. All remaining mistakes of style and substance are the responsibility of the author.

This book was reproduced from a camera-ready copy supplied by the author who gladly acknowledges the generous access to computers provided by CSLI.

Contents

Introduction	7
1 Left-associative grammar	13
1.1 The constituent structure paradox	13
1.2 The irregular left-to-right order of constituent structure	16
1.3 Left-associative versus categorial grammar	18
1.4 Left-associative trees as structured lists	22
1.5 Parsing continuous text	27
2 Left-associative parsing	33
2.1 A production system with a simple control structure	33
2.2 Modularity and expansion of the grammar	36
2.3 The parsing history as linguistic analysis	41
2.4 Parsing ungrammatical input	50
2.5 Some computational contrasts with other parsers	52
3 The category system of left-associative grammar	57
3.1 Syntactic categories and syntactic rules	57
3.2 Determiner-noun agreement in DCAT	61
3.3 Adjective agreement in DCAT	65
3.4 Combining noun phrases and verbs in DCAT	70
3.5 Treating different kinds of noun phrases in DCAT	74
4 The local nature of possible continuations	95
4.1 The treatment of word order in DCAT	95
4.2 Passive and other constructions with auxiliaries in DCAT	112
4.3 Center-embedded versus extraposed relative clauses in DCAT	124
4.4 Syntactic equivalence in left-associative grammar	131
4.5 Remarks on the lexicon of left-associative grammar	145

5	A left-associative fragment of English	159
5.1	A distinctive categorization for English noun phrases	159
5.2	Combining noun phrases and verbs in ECAT	172
5.3	Passive and other constructions with auxiliaries in ECAT	188
5.4	Relative clauses in ECAT	198
5.5	Wh-interrogatives in ECAT	207

Appendices

A	The LISP functions of DCAT	219
A.1	The motor of a left-associative parser	220
A.2	The linguistic rules and rule packages	226
A.3	Auxiliary functions of the linguistic rules	245
A.4	Alphabetical list of DCAT-functions	249
A.5	The definitions of DLEX	253
B	A selection of DCAT test examples	273
B.1	List of category segments used in DCAT and DLEX	274
B.2	Declaratives with finite main verbs	276
B.3	Declaratives with auxiliaries and non-finite main verbs	280
B.4	Declaratives with topicalized non-finite verbs	285
B.5	Various predicate constructions in declarative main clauses . . .	291
B.6	Various predicate constructions in subordinate clauses	298
B.7	Passive constructions in declarative main clauses	302
B.8	Q-Passives in declarative main clauses	313
B.9	Passive in subordinate clauses	320
B.10	Q-passives in subordinate clauses	326
B.11	Multiple modal infinitives in main clauses	334
B.12	Multiple modal infinitives in subordinate clauses	338
B.13	Obligatory versus optional adverbs	347
B.14	The preposition <i>hinter</i> in various constructions	352
B.15	Adsentential clauses and adverbs in various positions	356
B.16	Relative clause agreement	360
B.17	Sentential complements	375
B.18	Infinitives with <i>zu</i> in main clauses	382
B.19	Infinitives with <i>zu</i> in subordinate clauses	395
B.20	Separable verbal prefixes	403
B.21	Yes/no-interrogatives	408
B.22	Wh-interrogatives	413
C	A selection of ECAT test examples	425
C.1	List of category segments used in ECAT and ELEX	425
C.2	Active voice constructions using the verb <i>give</i>	429
C.3	Passive voice constructions using the verb <i>give</i>	434

C.4	Genitive constructions	440
C.5	Auxiliaries taking noun phrases as the second argument	442
C.6	Auxiliaries taking an adjective as the second argument	444
C.7	Nominative agreement and the auxiliary <i>be</i>	445
C.8	Yes/no-interrogatives and related declaratives	448
C.9	Wh-interrogatives	455
C.10	The interrogative determiner <i>which</i>	465
C.11	<i>That</i> -clauses	467
C.12	Wh-interrogatives with <i>that</i> -clauses	470
C.13	Passives in subordinate clauses	474
C.14	Relative clauses modifying sentence final noun phrases	479
C.15	Relative clauses modifying mid-sentence noun phrases	486
C.16	Relative clauses modifying sentence initial noun phrases	492
C.17	The relative pronoun <i>who</i> as subject and object	495
C.18	Mixing relative clauses and <i>that</i> -clauses	501
C.19	Wh-interrogatives with relative clauses	506
C.20	Declaratives and interrogatives with “Wh-movement”	508
C.21	Subordinate clauses with and without complementizers	514
D	List of computer-generated sample derivations	521
D.1	The DCAT derivations	521
D.2	The ECAT derivations	532
	References	539