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

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

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 NEWCAT (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