different from the strategy imposed by Prolog, and partly automating the writing of rules. For example, arguments having a regular and predictable structure (parse tree building, logical form construction, etc.) can be automatically included in grammar rules. Such interpreters are written in Prolog and they transform a Prolog-like program into one that is directly executable. Fundamental techniques are proposed and illustrated for DCGs, such as automated construction of proof trees, interpretation of filler-gap dependencies, partial execution, and tabular parsing (i.e., chart parsing). The book ends with annotated sample programs that integrate most of the material presented in the preceding chapters. The bibliography is relevant and comprehensive, given the aims of the book.

To conclude, I feel that this book includes most of the material one would wish to see in an introductory book. By a small number of linguistic concepts and constructions, the authors present the main techniques a grammar writer can use in a large number of situations. There are, however, some points I would have liked to see included in this book. Very few parsers directly use the linguistic framework proposed here but rather are inspired by GPSGs, LFGs, HPSGs, UGs, etc. Thus it would have been of particular interest to have guidelines on Prolog techniques useful to such computational-linguistics frameworks. Some aspects, like feature percolation and control in GPSGs, are probably not trivial to deal with.

In spite of these nonessential restrictions, I feel that this book should be read by everyone in the field having a minimum interest in Prolog. This very convincing book also suggests the reading of more complex publications in the field like Dahl and St.-Dizier (1985, 1988) and JLP (1986).

REFERENCES

Colmerauer, A. 1978. Metamorphosis Grammars. In *Lecture Notes in Computer Science*, 63: 133–189, L. Bolc, (ed.). Springer-Verlag.
Dahl, V. and Saint-Dizier, P. (Eds.) 1985, 1988. *Natural Language Understanding and Logic Programming I* and *II*. North-Holland.
Sterling, L. and Shapiro, E. 1986. *The Art of Prolog*. MIT Press, Cambridge, MA.
JLP 1986. Special Issue of the *Journal of Logic Programming* on Natural Language Understanding 3(4), December 1986.

Patrick Saint-Dizier was one of the organizers of the Workshop on Natural-Language Understanding and Logic Programming, and is editor of the forthcoming proceedings, to be published by North-Holland. In 1987, he was a visitor at Simon Fraser University. Saint-Dizier's address is: Institut de recherche en informatique et systèmes aléatoires, Campus Universitaire de Beaulieu, Avenue du Général Leclerc, 35042—Rennes Cédex, France.

EFFICIENT PARSING FOR NATURAL LANGUAGE: A FAST ALGORITHM FOR PRACTICAL SYSTEMS (THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE: NATURAL-LANGUAGE PROCESSING AND MACHINE TRANSLATION)

Masaru Tomita

Reviewed by
Mirosław Bańko
Warsaw University

This book introduces a context-free parsing algorithm that is intended to be particularly useful in natural-language processing. Practical applications of the algorithm are discussed, including on-line parsing, interactive machine translation, and a technique to disambiguate a sentence by consulting the user interactively. The book opens with a diagnosis of the current state of work on parsing context-free languages. Two kinds of algorithms for context-free languages are available: specific algorithms for programming languages, and general algorithms intended to handle any context-free grammar. Neither of them, however, is suitable for natural languages. The former are very fast but not powerful enough to cope with some syntactic phenomena of natural languages; the latter are even too powerful, and therefore not so efficient as they could be. Tomita's solution to the problem is to find something in between: an algorithm that would be nongeneral in the class of context-free languages yet still powerful enough to handle the great many ambiguities inherent in any natural language.

The algorithm described can be viewed as an extension of the LR parsing algorithm, which is used for programming languages (Aho and Ullman 1977). Like the LR algorithm, it works strictly from left to right and is entirely table-driven; unlike the LR algorithm, however, it is intended to parse ambiguous sentences, and does it most efficiently, using the data structures that have been specially designed for it. The current state of parsing is represented in the form of a directed acyclic graph, and no part of an input sentence is parsed more than once in the same way. All the possible parses of an ambiguous sentence are produced in parallel and stored in a **packed shared forest** (PSF) for later computation. While the number of parses grows exponentially, the size of the PSF increases polynomially. The only other algorithm that produces a polynomial-sized parse forest without requiring a grammar to be in Chomsky Normal Form is Earley's (1970) general algorithm intended to handle any type of context-free grammar. Tomita shows, however, that Earley's original algorithm has a

defect and may occasionally produce an inadequate parse-forest representation.

It is claimed in the book that the algorithm described "is more efficient than any existing general parsing algorithms in terms of recognition time as far as practical natural language grammars are concerned" (p. 73). The crucial point about this statement is, obviously, the notion of "practical natural language grammars". As mentioned before, Tomita's algorithm is intended for a subset of context-free grammars. In particular, it is not able to handle cyclic grammars, e.g.:

S → S

S → x

and turns out to be less efficient than Earley's algorithm with "densely" ambiguous grammars, such as:

S → SSSS

S → SSS

S → SS

S → x

Although it seems unlikely that any natural-language grammar contains cyclic rules or dense ambiguities, the limitations of Tomita's algorithm with regard to natural-language grammars remain somewhat unclear.

The major arguments for the usefulness of the algorithm are, therefore, empirical. Both Tomita's and Earley's algorithms have been implemented and tested against various grammars and sentences. The grammar of English most used in the experiment has about 220 rules, and the set of test sentences is made up of both actual and invented examples, more than 50 in number. The results of the experiment show that Tomita's algorithm is significantly faster than Earley's algorithm in practical applications. Moreover, its parsing time and space remain tractable when the length of an input sentence, the number of parses, or the size of a grammar increases.

The book is composed of 10 chapters and several appendices. Chapters 1 through 6, apart from some introductory remarks, provide an informal description of the algorithm, illustrated with plenty of clear examples; a formal specification (which is not necessary, however, to understand how the algorithm works); a comparison of Tomita's algorithm with other context-free parsing algorithms; and finally a description of the experiment mentioned above. The appendices give, among other things, the correctness proof of the algorithm, as well as programs, grammars, and test sentences used in the experiment.

Chapters 7, 8, and 9 constitute a relatively independent part of the book. They suggest some practical applications of Tomita's algorithm, such as left-to-right on-line parsing, interactive sentence disambiguation by asking the user for the preferred interpretation of the sentence in question, and interactive machine transla-

tion of small documents, letters, etc. To a wide circle of readers involved in computational linguistics, this part of the book may be even more interesting than the others. The core of the book, on the contrary, seems to be of great importance for those who deal in practice with natural-language parsing rather than those who observe the advances in this field from the outside.

REFERENCES

Aho, A.V. and Ullman, J.D. 1977. *Principles of Compiler Design.* Addison-Wesley, Reading, MA.

Earley, J. 1970. An Efficient Context-free Parsing Algorithm. *Communications of the ACM* 13(2): 94–102.

Mirosław Bańko, who is a lecturer at the Institute of Polish Language, Warsaw University, has MScs in both computer science and linguistics. His recent research has concentrated on the problems of evaluating natural-language parsers, and on parsing Polish. Bańko's address is: Instytut Języka Polskiego, Universytet Warszawski, Krakowski Przedmiéscie 26/28, 00-325 Warszawa, Poland.

BRIEFLY NOTED

Editor's note: Frequently, the need has been felt for a longer treatment of books that on the one hand do not warrant a full review, but on the other hand merit more than a bare appearance in the list of books received. Often, such books are those that are important in related fields such as theoretical linguistics and will be reviewed in the appropriate journals, but which may have relevance to computational linguistics as well. Sometimes, it is simply the case that a mere listing by title and author does not convey enough information to let the reader know whether the book may be in their area of interest. In addition, popular works or books of oblique interest to the field may merit a brief notice.

This new section will answer this need by giving brief notices of books. Most often, books appearing in this section will not have a review per se; rather, just a summary of the book will usually be given—based on that of the author or publisher, in the case of monographs, or through the table of contents in the case of collections. Such a treatment reflects the book review editor's judgment that the book is worthy of attention, but is not necessarily an endorsement of the content. Short opinions may also be given in some cases, where the nature of the book calls for an evaluation but not a full review.

ON THE DEFINITION OF WORD
(LINGUISTIC INQUIRY MONOGRAPH 14)

Anna-Maria Di Sciullo and Edwin Williams

On the Definition of Word develops a consistent and coherent approach to central questions about morphology and its relation to syntax. In sorting out the various senses in which the word *word* is used, it asserts that three concepts that have often been identified with each other are in fact distinct and not coextensive: **listemes** (linguistic objects permanently stored by the speaker), **morphological objects** (objects whose shape can be characterized in mor-