The Functional Treatment of Parsing

René Leermakers
(Institute for Perception Research, Eindhoven)

Dordrecht: Kluwer Academic
Publishers (The Kluwer international
series in engineering and computer
science; Natural language processing
and machine translation, edited by
Jaime Carbonell, volume 242), 1993,
xvi + 158 pp.
Hardbound, ISBN 0-7923-9376-7, $79.95,
£57.50, Dfl 160.00

Reviewed by
Yves Schabes
Mitsubishi Electric Research Laboratories

The study and implementation of parsing technologies are traditionally thought of as the foundation of two apparently unrelated fields, namely, programming languages and natural language processing. These two fields have often been seen as addressing problems on two different ends of a spectrum. Programming languages have an unambiguous syntax and can be processed deterministically, whereas natural languages are ambiguous. Although the two fields can be seen as unrelated in this respect, they share the study of parsing technologies. Modern programming language design relies heavily on grammatical frameworks and their parsing algorithms for syntactic and semantic analysis. Similarly, natural language systems rely on formalisms and their parsing algorithms. As a computational linguist, one should be well aware of the parsing methods of the 1960s and onward. Functional programming has recently gained popularity in the programming language community. In natural language processing, the statement of parsing algorithms in a functional notation and the use of memoization to handle non-determinism in parsing algorithms (Norvig 1991) are analogous to this direction.

This book presents a functional treatment of parsing. Most well-known parsing algorithms for programming languages and natural languages are represented uniformly in a functional notation. The presentation relies heavily on new notations and simple mathematical concepts. The first chapters introduce a notation, "bunch" notation, for sets and their constructs. Then, an algorithmic interpretation for the previously defined language is defined. Leermakers achieves a tour de force: each (possibly non-deterministic) parsing algorithm is represented in a compact functional notation whose algorithmic interpretation yields an implementation of the algorithm. The interpretation of context-free grammars is stated in a similar style using the bunch notation. Traditional top-down parsing is stated in functional notation. The algorithmic interpretation of this functional notation yields a cubic-time parser for bilinear (at most binary branching) non–left-recursive grammars. Algorithmic conversions between top-down and bottom-up algorithms are presented as a grammar transformation that turns a grammar into a bilinear form in which left recursion has been eliminated. The top-down evaluation of the transformed grammar can be seen as a "bottom-up" parser reminiscent of a generalized left-corner parser. The approach is generalized to context-free grammars with regular expressions on the right-hand side. Then, the recognizer previously defined is extended to a parser with the use of parse forests for recovering derivations. Attribute grammars and the well-known LR parsing algorithm are also described. The last chapter consists of miscellaneous notes in the form of comments and bibliographical references.

Overall, the book is mostly consistent and well written. The content is rich, and I would recommend it to anyone interested in parsing algorithms. However, the style chosen for mathematical argumentation requires an attentive, nonlinear study. The definitions are clear syntactically, but they make sense only after reading the context in which they are used. The reader should also be aware that linguistics is not discussed. Some readers may actually be offended by the grammar used for illustrating context-free grammars, which treats intransitive verbs as transitive verbs with the empty string substituted as complement.

The functional paradigm is not well motivated. The book would also have been much stronger had experiments and details of implementation been included. Without this, one may wonder whether the proposed approach is of only mathematical interest. The fact that some complexity analysis given in the book takes into account only the length of the input and not the size of the grammar makes this problem even more acute. Also, the complexity of the most general parser, the left-corner (recursive ascent) parser is analyzed as $O(|G|pqn^3)$, where $|G|$ is the size of the grammar, p is the number of non-terminals, and q is the number of items of the form $C \rightarrow X \cdot \delta$ with the same X. This is bigger than $O(|G|n^3)$, which is the complexity of standard context-free grammar parsers, such as the CKY parser (Kasami 1965), the parser described by Graham, Harrison, and Ruzzo (1980), or efficient implementations of Earley's algorithm (Earley 1970).

Most of the references to previous work are found in the notes of the last chapter. Although they are adequate, some important references are missing. For example, the grammar transformation used for eliminating left recursion is very similar to the one described by Rosenkrantz and Lewis (1970) and Rosenkrantz (1967). Similarly, the interpretation of context-free grammars given in the book is similar to that using equations on languages (Chomsky and Schützenberger 1963), which is also described in numerous textbooks, such as Salomaa and Soittola 1978, and Gross and Lentin 1970.

The book is mostly free of editorial inconsistencies or errors. Some slipped through, however. For example, on page 55, the caption of Figure 5.3 should refer to Figure 3.1 instead of Figure 1.2.

Moreover, three different notations for context-free grammars are used. On page 16, a grammar rule is represented as a pair (A, α), which states that the non-terminal A can be rewritten as the string of symbols α. However, when the interpretation of a grammar is given, on page 22, the pair (A, η) represents that the string η can be derived from the non-terminal A. Then, a grammar rule $(A, \eta\rho) \rightarrow (X, \eta)(Y, \rho)$ states that the non-terminal A can be rewritten as $X\ Y$ and that the derived string from A $(\eta\rho)$ is the concatenation of the one from X (η) and the one from Y (ρ). Later, a context-free rule is written as $A \rightarrow X\ Y$.

Although most of the book is mathematically rigorous, there are a few cases where more rigor would have been appropriate. On pages 18 and 19, the existence and the uniqueness of the smallest solution for the interpretation of context-free grammars are proven very casually. Checking the applicability of Tarski's fixed-point theorem (Tarski 1955) on complete partial orders would have been one way to solve this question. The existence of a complete partial order in which the functions used are continuous would have guaranteed the existence and the uniqueness of the solution. Another way to prove this fact would have been to check the applicability of Banach's fixed-point theorem for contracting mapping in complete metric spaces (Banach 1922), as is usually done for formal power series (Chomsky and Schützenberger 1963).

Although most of the book follows a rigorous mathematical argumentation, the reader is from time to time entertained by casual, poetic, and Persian Gulf–inspired diversions such as these:

> The above gives already a fairly complete picture of context-free grammars—complete enough, in fact, for understanding much of this book. Chapter 3 describes the same concept in a pedantic way. (pp. 4–5)
>
> Context-free grammars are usually presented as one level in the Chomsky hierarchy of rewriting grammars [Chomsky, 1959]. This is a very one-sided way to depict context-free grammars, however. The formalism of context-free grammars is like a piece of art that can be placed in any of a great number of art styles but never really fits in. (p. 29)
>
> In theoretical settings, the treatment of rules with regular expressions is often a bit cumbersome. Hence, it is tempting to dismiss regular expressions with the excuse that they do not add to the weak generative power anyway. To make up for our succumbing to this temptation, we show in this section how to generalize results for normal context-free grammar to EBNF grammars. (p. 70)
>
> The way context-free grammars have been formally introduced is unconventional. Normally, they are presented as rewriting systems—as one level in the Chomsky hierarchy (see [Chomsky, 1959]). I always felt uneasy about this injustice: the formalism of context-free grammars is not part of one hierarchy; it is the mother of all grammar hierarchies. Therefore, I decided to start differently. (p. 143)

In conclusion, those familiar with parsing algorithms will enjoy reading this book. However, it may not be appropriate as an introduction to parsing algorithms.

References

Banach, Stefan (1922). "Sur les operations dans les ensembles abstraits et leurs applications aux equations integrales." *Fundamenta Mathematicae* 3:7–33.

Chomsky, Noam and Schützenberger, Marcel Paul (1963). "The algebraic theory of context-free languages." In *Computer Programming and Formal Systems,* edited by P. Braffort and D. Hirschberg. Amsterdam: North-Holland.

Earley, Jay C. (1970). "An efficient context-free parsing algorithm." *Communications of the ACM* 13(2):94–102.

Graham, Susan L.; Harrison, Michael A.; and Ruzzo, Walter L. (1980). "An improved context-free recognizer." *ACM Transactions on Programming Languages and Systems* 2(3):415–462.

Gross, Maurice and Lentin, André (1970). *Introduction to Formal Grammars.* 196–215. New York: Springer-Verlag.

Kasami, Tadao (1965). "An efficient recognition and syntax algorithm for context-free languages." Technical Report AF-CRL-65-758, Air Force Cambridge Research Laboratory, Bedford, Massachusetts.

Norvig, Peter (1991). "Techniques for automatic memoization with applications to context-free parsing." *Computational Linguistics* 17(1):91–98.

Rosenkrantz, Daniel J. (1967). "Matrix equations and normal forms for context-free grammars." *Journal of the Association for Computing Machinery* 14(3):501–507.

Rosenkrantz, Daniel J. and Lewis, Philip M., II (1970). "Deterministic left corner parsing." In *IEEE Conference Record of the 11th Annual Symposium on Switching and Automata Theory.* 139–152.

Salomaa, Arto, and Soittola, Matti (1978). *Automata-Theoretic Aspects of Formal Power Series.* New York: Springer-Verlag.

Tarski, Alfred (1955). "A lattice-theoretical fixpoint theorem and its applications." *Pacific Journal of Mathematics* 5:285–309.

Yves Schabes is a research scientist at Mitsubishi Electric Research Laboratories. In his thesis work and as a research associate at the University of Pennsylvania, he led the design and implementation of a wide-coverage tree-adjoining grammar for English. He has designed a number of other grammars and parsing algorithms and has investigated statistical models for natural language processing. Schabes's address is Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139. E-mail: schabes@merl.com