

## Some Books on Parsing and Related Subjects

- Grune, Dick, Criel Jacobs, *Parsing techniques, a practical guide*, Ellis Horwood, Chichester, England, 1990, 322 pp.

Mainly intuitive formula-free treatment of virtually the entire body of knowledge about parsing and some related topics, including formal languages, automata, efficient general parsing and traditional subjects like LL(1) and LALR(1). Extensive bibliography.

- Carroll, John, Darrell Long, *Theory of finite automata, with an introduction to formal languages*, Prentice-Hall International Ed., Englewood Cliffs, NJ 07632, 1989, 438 pp.

Extensive and easily readable on finite automata, restricted and easily readable on formal languages and parsing.

- Sippu, Seppo, Eljas Soisalon-Soininen, *Parsing theory, volume I: languages and parsing*, EATCS monograph on theoretical computer science, Springer Verlag, Berlin, 1988, 228 pp.

The theory behind parsing is treated in an evenly-paced exposition that is never more difficult than is necessary. Includes exercises.

- Sippu, Seppo, Eljas Soisalon-Soininen, *Parsing theory, volume II: LL and LR parsing ??*, EATCS monograph on theoretical computer science, Springer Verlag, Berlin, 19??.

See the above.

- Chapman, Nigel P., *LR parsing: theory and practice*, Cambridge University Press, New York, NY, 1987, 228 pp.

Detailed treatment of the title subject. Highly recommended for anybody who wants to acquire in-depth knowledge about LR parsing. Good on size of parse tables and attribute grammars.

- Révész, György E., *Introduction to formal languages*, McGraw-Hill, Singapore, 1985, 199 pp.

This nifty little book contains many results and elementary proofs of formal languages, without being “difficult”. It gives a description of the ins and outs of the Chomsky hierarchy, automata, decidability and complexity of context-free language recognition, including the hardest CF language. Parsing is discussed, with descriptions of the Earley, LL( $k$ ) and LR( $k$ ) algorithms, each in a few pages.

- Rayward-Smith, V.J., *A first course in formal languages*, Blackwell Scientific, Oxford, 1983, 123 pp.

Very useful intermediate between Révész and Hopcroft and Ullman. Quite readable (the subject permitting); simple examples; broad coverage. No treatment of LALR, no bibliography.

- Hopcroft, John E., Jeffrey D. Ullman, *Introduction to automata theory, languages, and computation*, Addison-Wesley, Reading, Massachusetts, 1979, 418 pp.

A must for readers interested in formal language theory and computational (im)possibilities.

- Backhouse, Roland C., *Syntax of programming languages*, Prentice Hall, London, 1979, 290 pp.

Grammars are considered in depth, as far as they are relevant to programming languages. FS automata and the parsing techniques LL and LR are treated in detail, and supported by lots of well-explained math. Often complete and efficient algorithms are given in Pascal. Much attention is paid to error recovery and repair, especially to least-cost repairs and locally optimal repairs. Definitely recommended for further reading.

- Gonzales, R.C., M.G. Thomason, *Syntactic pattern recognition*, Addison-Wesley, Reading, Mass., 1978, 283 pp.

This book provides numerous examples of syntactic descriptions of objects not normally considered subject to a syntax. Examples range from simple segmented closed curves, trees and shapes of letters, via bubble chamber events, electronic networks, and structural formulas of rubber molecules to snow flakes, ECGs, and fingerprints. Special attention is paid to grammars for non-linear objects, for instance web grammars, plex grammars and shape grammars. A considerable amount of formal language theory is covered. All serious parsing is done using the CYK algorithm; Earley,  $LL(k)$  and  $LR(k)$  are not mentioned. Operator-precedence, simple precedence and finite automata are occasionally used. The authors are wrong in claiming that an all-empty row in the CYK recognition matrix signals an error in the input.

Interesting chapters about stochastic grammars, i.e. grammars with probabilities attached to the production rules, and about grammatical inference, i.e. methods to derive a reasonable grammar that will produce all sentences in a representative set  $R^+$  and will not produce the sentences in a counterexample set  $R^-$ .

- Cleaveland, J. Craig, Robert C. Uzgalis, *Grammars for programming languages*, Elsevier, New York, 1977, 154 pp.

In spite of its title, the book is a highly readable explanation of two-level grammars, also known as van Wijngaarden grammars or VW grammars. After an introductory treatment of formal languages, the Chomsky hierarchy and parse trees, it is shown to what extent CF languages can be used to define a programming language. These are shown to fail to define a language completely and the inadequacy of CS grammars is demonstrated. VW grammars are then explained and the remainder of the book consists of increasingly complex and impressive examples of what a VW grammar can do. These examples include keeping a name list, doing type checking and handling block structure in the definition of a programming language. Recommended reading.

- Aho, Alfred V., Jeffrey D. Ullman, *The theory of parsing, translation and compiling, Volume I: parsing*, Prentice Hall, Englewood Cliffs, N.J., 1972, 542 pp.

The book describes the parts of formal languages and automata theory relevant to parsing in a strict mathematical fashion. Since much of the pertinent theory of parsing had already been developed in 1972, the book is still reasonably up to date and is a veritable trove of definitions, theorems, lemmata and proofs.

The required mathematical apparatus is first introduced, followed by a survey of compiler construction and by properties of formal languages. The rest of the book confines itself to CF and regular languages.

General parsing methods are treated in full: backtracking top-down and bottom-up, CYK and Earley. Directional non-backtracking methods are explained in detail, including general  $LL(k)$ ,  $LC(k)$  and  $LR(k)$ , precedence parsing and various other approaches. A last chapter treats several non-grammatical methods for language specification and parsing.

Many practical matters concerning parser construction are treated in volume II, where the theoretical aspects of practical parser construction are covered; recursive descent is not mentioned, though.

- Aho, Alfred V., Jeffrey D. Ullman, *The theory of parsing, translation and compiling, Volume II: compiling*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, 543-1002 pp.

In spite of the subtitle of the volume, the first 180 pages are a continuation of volume I and are concerned with parsing. Precedence functions, bounded-context,  $LR(k)$ , parsing automata and the underlying theory of deterministic parsing are treated in detail. Like volume I, even after 20 years the book has lost nothing of its original freshness.