

Prolog for Natural Language Processing

Annie Gal, Guy Lapalme, Patrick Saint-Dizier, and Harold Somers
(SYSECA S.A., Université de Montréal, Université Paul Sabatier, and UMIST)

Chichester, England: John Wiley (Wiley Professional Computing Series), 1991, xiii + 306 pp.
Paperbound, ISBN 0-471-93012-1, \$47.95.
Program diskette, 5.25-inch IBM compatible, ISBN 0-471-93089-X, \$26.60, £12.50

Reviewed by
Norman M. Fraser
Logica Cambridge Ltd

As a discipline, NLP is not noted for its traditions. However, a welcome trend is beginning to emerge with the publication in the last three odd-numbered years of impressive texts on NLP and the Prolog programming language. First there was Pereira and Shieber's (1987) *Prolog and Natural Language Analysis*; then came Gazdar and Mellish's (1989) *Natural Language Processing in PROLOG*; the latest offering is *Prolog for Natural Language Processing* by Gal, Lapalme, Saint-Dizier, and Somers (a revised translation of Gal et al. [1989]).

There is currently a glut of general Prolog texts on the market. Happily, the same cannot be said in the more specialized field of Prolog-based NLP; each of the available texts justifies its place on the bookshelf. Pereira and Shieber (1987) is a theoretically focused work, written by researchers, for researchers. Gazdar and Mellish (1989) is the broadest of the three, providing a general introduction to NLP, with illustrations from Prolog. Gal et al. (1991) is the most practical of the bunch, focusing presentation around problems for which neat logic programming solutions exist. The whole emphasis is on hands-on rapid prototyping; giving the student a feel for how to think about problems in a logic-programming fashion. A particular strength of the presentation is that each problem raised is followed through all the way to a working solution.

Chapter 1 offers a brief overview of NLP. Chapter 2 introduces Prolog. This ranks as one of the most accessible whistle-stop tours of the language that I have yet seen. It is therefore particularly regrettable that the answer to the very first exercise includes an egregious error (`item1`, an atom, is identified as a variable). Each of the chapters includes useful exercises, although (sensibly) answers are only published for Chapter 2.

The volume is strongest where it sticks to straightforward NLP problem solving with Prolog. Digressions into linguistic theory are generally regrettable. This is most clearly seen in Chapter 3 ("The Linguistic Background"), which displays an extraordinary fixation with a dated version of transformational grammar. For example, consider the following assertion:

Transformational generative grammar... is the source of inspiration for the majority of Prolog grammars, or "logic grammars" (p. 46).

Is this true? Where is the evidence? If I were forced with a gun to my head to make an unargued, impressionistic assertion of this kind, I would say that TG is the source of

inspiration for a small minority of Prolog grammars, the vast majority being squarely on the side of monostratal theories. In a section entitled "An Example of a Transformation," a rather vague description is given of "one of the most useful or usual [transformations]: *wh*-movement" (p. 58). Don't the authors realize that specific transformations such as *wh*-movement are nowadays as "useful" or "usual" as a starting handle on a family car? The one remaining transformation—Move- α —is, at this moment, fighting to avoid extinction.

The sting in the tail, however, is that transformations or transformational grammar are not mentioned again after the end of Chapter 3. Functional grammar, on the other hand, which warrants only a three-page introduction at the end of Chapter 3, crops up several times later in the book.

Chapter 4 ("Using Logic Grammars") is a good introduction, particularly in respect of definite clause grammars, which provide the foundation for much of the work covered in the book. A few lingering effects of Chapter 3 can still be felt, as in Section 4.6, which promises to deal with "deep structures" (first mention), thus raising expectations that a real live transformation might be about to appear. What is actually on offer is a simple surface analysis of various kinds of "movement" phenomena. A few lines later we read: "As we saw in the previous chapter, relative clauses are . . .". Alas, the previous chapter did not so much as mention relative clauses.

Chapter 5 ("Semantic Representation") is an accessible introduction to linguistic semantics, using the formalism of three-branch quantifier trees. The manner of presentation adopted is most effective, and involves enumerating a number of linguistic phenomena (relative clauses, noun complements, verb complements, determiners, adjectives) together with a description of the target three-branch quantifier tree for each example. Once this foundation has been laid, the problem of composing a semantics for a given sentence is tackled. Where this introduction really starts to pay off is in its presentation of the vital next step: how to use the results of semantic analysis to achieve functions such as data retrieval and update. All too often, introductory texts and courses on NLP are so preoccupied with language description that they stop just short of practical applications of language use.

Chapter 6 ("Automatic Text Generation") provides the best introduction to text generation to be found in the three available Prolog-and-NLP volumes. It covers areas such as canned messages, template generation, rule-based generation, paraphrases, reversible grammars, and explanation, earthing the discussion in examples from expert systems, generation from data, and machine translation.

Chapter 7 ("Co-operative Responses") skillfully brings together the threads of the previous practical chapters. By the end of this chapter, even the most reluctant student of NLP should have begun to catch something of the excitement of creating a system capable of engaging in pseudo-natural dialog (along with the know-how to be able to do it). Chapter 8 provides a general conclusion, a brief description of the "language industry," and a few pointers to the literature.

The volume includes 63 pages of program listings. These are also available on a 5¼ inch DOS-format disk, available at a cost of £12.50 or \$26.60 from the publisher. Programs are written in C-Prolog. Compatibility files are supplied for use with Edinburgh, BIM, Quintus, and Sixtus Prologs.

I want to be able to say positive things about this volume, which I enjoyed reading and which could, I believe, prove to be useful and interesting to teachers and students of practical NLP. I hesitate before doing so because of the unacceptably high number of minor irritations, including typos, misleading index and in-text references, curious citations, and confusing and pointless forays into theoretical linguistics. I hope that a corrected second edition appears soon so that I can recommend it without reservation.

Exercise:

Write a definite clause grammar capable of generating the titles of all existing textbooks combining Prolog and NLP and of predicting the set of all such textbook titles.

References

- Gal, Annie; Lapalme, Guy; Saint-Dizier, Patrick; and Somers, Harold (1989). *Prolog pour l'analyse automatique du langage naturel*. Eyrolles.
- Gazdar, Gerald and Mellish, Chris (1989). *Natural Language Processing in PROLOG*. Addison-Wesley.
- Pereira, Fernando C. N. and Shieber, Stuart M. (1987). *Prolog and Natural Language Analysis*. Center for the Study of Language and Information.

Norman M. Fraser is a consultant working in the Speech and Language Division of Logica Cambridge Ltd and an Honorary Visiting Research Fellow at the University of Surrey. A graduate of the Universities of Edinburgh and London, his research interests include dependency grammar, parsing, inheritance, dialogue management and the specification and evaluation of spoken language systems. Fraser's address is: Betjeman House, 104 Hills Road, Cambridge, CB2 1LQ, UK; e-mail: norman@logcam.co.uk