

## Semantic Processing for Finite Domains

**Martha Stone Palmer**

(National University of Singapore)

Cambridge, England: Cambridge  
University Press, 1990, ix + 199 pp.  
(Studies in natural language processing)  
Hardbound, ISBN 0-521-36226-1, \$49.50

*Reviewed by*  
*Brian M. Slator*  
*Northwestern University*

Martha Stone Palmer has written a pretty good book. The subject matter is important, and the methodology, while narrowly drawn, is interesting and well done. The book is based on a thesis, and has some of the usual failings of books of this sort; but there is plenty of substance to go around, either for those intellectually interested in the subject, or for those interested in implementing a semantic analysis program for a finite domain.

This notion of finite domain is a theme that reappears throughout the book, and it is a crucial caveat. The road to Natural Language Understanding is littered with systems that failed to “scale up,” but, of course, the closed-world assumption is a time-honored tradition in the business and, indeed, it has been the unifying assumption since machine translation tackled Russian physics in the 1950s. The tables are turned in this book, because this system is explicitly not intended to scale up; rather, a very constrained domain is formalized, and an interpreter is devised that is “easily transportable to other limited domains which can be similarly formalized” (p. 111).

The limited domain in this instance is pulleys; in particular, the abstracted domain of physics word problems, where particles are suspended from frictionless pulleys with idealized strings. A set of example problems are listed in the appendices. Each is two to four sentences long and mostly of this form:

Two pulleys [...] are connected by a fine string hanging over a smooth fixed pulley. [...] Find  $X$  such that [...]. (p. 174)

In this,  $X$  is something like weight, tension, or acceleration. The sentences are not trivial, and most of the problems require knowing at least one formula in order to solve them.

The semantic processing operates under two clear input/output assumptions:

- The input sentences are parsed beforehand, and all syntactic constituents are correctly identified and labeled.
- A problem-solving system is waiting at the other end to actually solve the word problem.

The goals of the semantic processing are, as is usual in systems of this sort, to map from syntactic constituents to verb representation, to deepen that representation through inference, and to integrate the final representation into memory.

Much of the implementation is captured in the lexical entries for the verbs, which are defined as Prolog Horn clauses. Given a syntactic analysis, the verb’s definition

is then "proven," which sometimes requires pragmatic information to provide missing information. The semantic processing is done through "analysis as synthesis," which essentially means generating hypotheses and rejecting any that do not match the known constraints. The underlying intuition is that the necessary steps for analysis and inferencing can be expressed via the lexical entries as a grammar. In effect, the formalization of the domain permits a "language of inference" where the grammatical sentences of the language are semantic interpretations that can hold. The advantage of this formulation is that multiple levels of intermediate representation are collapsed together, which is perspicuous, and the lexical entries for the verbs can be kept to a minimum, since the different semantic/case patterns for each are derived (i.e., generated grammatically) rather than stored, all of which is economical. In other words, the lexical entries for verbs are procedures, and semantic processing amounts to filling in semantic roles as a result of executing these. The whole thing is neatly and economically done.

By the same token, the writing is tightly packed and somewhat uncompromising. The style is straight-ahead and "no nonsense," which makes for economy of expression, but also makes for a difficult read and some slow going. As one would expect from a book of this sort, the scholarship is in place. The historical survey is quite thorough, within its limited range, and is even insightful, but not for the faint of heart or the novice. Indeed, the survey section is written at such a level that considerable context is required just to wade through it. Further, the book has no index, which is sometimes annoying, particularly to the student.

The issue of context, in one form or another, is the one that this book most fails at handling. We are told little or nothing about the problem-solver that this system is intended to feed into, and this makes it quite difficult to judge whether the system is doing what one would hope. Similarly, we are told little about the parsing system that feeds into this one, and this is most worrying. There are, for example, a great many people who believe that decoupling syntax from semantics, as happens here, is an error in principle.

The problem being solved here is quite neatly compartmentalized, and this falls into line with the recent opposing trend in computational linguistics, toward separating syntax from semantics. The system described in this book assumes parsed input where all constituents are correctly labeled and all noun references are correctly identified. Many would argue that, in general, some sort of semantic processing is needed in order to do that degree of syntactic analysis in the first place. But this is where the finite domain assumption comes to the rescue. In the pulley world, *hang* is not defined in terms of 22 senses, as it is in the *American College Dictionary*, and *at* always marks a LOCATION and never a TIME. This means problems for anyone looking for a general solution to the language understanding problem. However, as mentioned at the outset, the assumption at play here concerns the value of substituting transportability for scalability (so long as the domain is finite, clearly delineated, and formalizable).

As is often the case with books in this business, the examples used to illustrate various points are sometimes more than a little odd. For example, "A particle is attached to a string at its end" (p. 122), "John shot the turkey with a bullet from a rifle" (p. 63), "The end of the rope is pulled three feet" (p. 89), and "The stone wall had been crushed by nothing more than a mallet" (p. 13), are all offered, without comment, as examples of standard English usage. They are not, of course, but this merely outlines the problem of finding good examples to illustrate processing, which accounts for why we see the same ones over and over again in the literature.

Quibbling aside, Palmer's book succeeds at several levels. There is a nice balance between theory and practice. The semantics are formal, but not maddeningly so. All

examples are given in Prolog notation, and this is convenient for the Prolog-literate. The entire system, as rendered in the appendices, appears to be about 100 statements of Prolog. This alone is enough to make a book tempting.

*Brian M. Slator* is a research associate at the Northwestern University Institute for the Learning Sciences, where his research involves developing case-based systems for consulting and tutoring. He received his Ph.D. from New Mexico State University for work on the semantic structure of dictionaries. His first book, *Word Meaning and Language Understanding*, will be published soon. Slator's address is: The Institute for the Learning Sciences, Northwestern University, Evanston, IL 60201; e-mail: [slator@ils.nwu.edu](mailto:slator@ils.nwu.edu)