

WORKSHOP REPORT

Natural Language Understanding and Logic Programming

Patrick Saint-Dizier

The second Natural Language Understanding and Logic Programming (NLULP) Workshop was held on 17-19 August 1987 at Simon Fraser University, Vancouver, British Columbia, Canada, (the first NLULP workshop was held three years ago in Rennes, France)

The second Natural Language Understanding and Logic Programming (NLULP) Workshop was held on 17-19 August 1987 at Simon Fraser University, Vancouver, British Columbia, Canada, (the first NLULP workshop was held three years ago in Rennes, France). About 70 participants from 10 countries attended the various talks and discussions in a particularly friendly and cooperative atmosphere. This workshop was sponsored by Simon Fraser University, the National Sciences and Engineering Research Council of Canada, the American Association for Artificial Intelligence, and the French Institut National de Recherche en Informatique et Automatique. The final proceedings will be published in April 1988 by North Holland.

Before briefly introducing the main trends of the workshop, let me precisely define what is meant by natural language understanding and logic programming. In light of some of the talks and discussions, it turns out that this title applies to works where logic programming (in particular, Prolog, although there are a number of other logic-programming languages) is viewed as a convenient implementation framework and a clear (and sometimes simple) formal framework for describing linguistic phenomena. However, as pointed out in several talks, it appears that the current logic-programming languages do not have all the tools and theoretical framework which most linguists and computational linguists need. For this reason, several author's presentations attempted to improve the explanatory coverage and power of various logic-programming languages with respect to natural language. In this range of ideas, practical, as well as formal,

extensions have been proposed.

Although the NLULP community is still small, interest in this approach is rapidly increasing. I feel this increase is mainly the result of three major contributions to the field of logic programming: (1) the publication of several clear and relevant books and articles on the theoretical foundations of logic programming, (2) the addition of a number of good books on programming in Prolog, and (3) the commercialization of large and efficient Prolog interpreters and compilers available on many types of machines. These remarks implicitly suggest that the study of how natural language understanding (in its broadest sense) and logic programming could merge to produce well-designed and efficient systems is still an open and attractive research field. Another point worth mentioning is that about half the speakers and half the audience at the workshop came from companies, which clearly indicates the high interest of the industrial community in this approach.

The first day of the workshop was devoted to syntax. Apart from a paper presenting comparisons between current implementations of grammar formalisms, the two main areas of concentration were compiler design for natural language grammar systems and the improvement of current logic-based grammar formalisms. Besides functional grammars and unification-based grammars, some interest in government and binding theory has emerged.

The second day was devoted to semantics. The high-quality papers presented in this area show that natural language semantics in logic programming is no longer a neglected area. In these presentations, the



GURU: When you expect the world from your expert system.

Tired of being limited? It's time to discover GURU — an expert system environment with comprehensive inference engine and rule management controls

Incredible flexibility.

GURU lets you design an expert system that runs exactly the way you want. By using fuzzy variables, certainty factors, reasoning rigor, rule selection order, and numerous environmental and utility variables, you have virtually unlimited control over the consultation environment.

Unprecedented development efficiency.

Using GURU's case saving and replay, you can track the effects that rule changes have on system behavior. You can also use meta rules to examine or alter other rules during a consultation. And using GURU's knowledge tree, you can display the relationships and dependencies between an application's rules, variables, and goals.

Quick and thorough.

By mixing forward and backward chaining, goal search time can be shortened dramatically. And, using GURU's multiple rule firing capability you can re-fire rules as values change. GURU also comes equipped with seamlessly integrated 4th generation decision support capabilities such as data base, spreadsheet, and report generator.

GURU runs on PCs, LANs, and VAXs

To find out how GURU can exceed your expectations, call 1-800/344-5832 or 317/463-2581



PO Box 248
Lafayette, IN 47902
1-800/344-5832
317/463-2581

Explore the world of expert systems with GURU Tutor!

This full-featured development environment allows you to prototype GURU applications using rule sets, data bases, spreadsheets, and more for only \$75. To order, call us at 1-800/344-5832 or write VISA, MasterCard, and American Express accepted.

GURU is a registered trademark of mdbs, Inc. VAX of Digital Equipment Corp.

For free information circle no. 85

expression of Montague semantics, conceptual graphs, and H. Kamp's discourse representation structures in logic programming were explored. The results presented tend to show that this area is promising but that it certainly requires more mutual understanding between semanticists and logic programmers.

The third day was devoted to pragmatics. A number of topics were addressed, such as disambiguation of word senses, the use of integrity constraints to provide cooperative responses, and problems of the interpretation of questions. An interesting point that emerged was the intimate connection between pragmatics and deductive databases.

Besides talks, four panel sessions were held: Linguistic Theories and Logic Programming, What Is Logic Grammar?, Semantics and Pragmatics in Logic Programming, and Applications (industrial or academic). Technical and general problems stimulated controversial discussions that often lasted till late in the evening.

This type of workshop was felt by everyone to be useful and rewarding. The next workshop will be held in two years in Europe.

For free information circle no. 31

The Computer and the Mind

An Introduction to Cognitive Science
P.N. Johnson-Laird

In a field choked with seemingly impenetrable jargon, Philip Johnson-Laird has done the impossible: written a book about how the mind works that requires no advance knowledge of artificial intelligence, neurophysiology, or psychology, providing the single best introduction to cognitive science available.

"Philip Johnson-Laird has that rare gift of being a cognitive scientist of the first order, yet he addresses himself to the deep classical issues in psychology, in the philosophy of mind and in linguistics. He never avoids the hard problems in order to make life easier for the cognitive sciences. This is an admirable book." --Jerome Bruner

\$29.50

Intention, Plans, and Practical Reason

Michael E. Bratman

"Bratman presents an alternative, *plan-based*, conception of our cognitive architecture — one in which *intentions* play an irreducible, and central, role. This is an important book — forcefully argued, and clearly written. It also promises to play a significant role in a rapprochement between work in the philosophy of rational agency and research on planning within Artificial Intelligence."

--David J. Israel, SRI International,
Artificial Intelligence Center

\$25.00

Available at bookstores, or from

Harvard University Press

79 Garden St, Cambridge, MA 02138