Although it has not been discussed in this paper, the formalization of rules relating linguistic form to domain structure is of importance to the generation as well as to the analysis of a natural language. The ease of technology transfer from linguistic analysis to generation tasks is dependent on the extent to which linguistic knowledge is encoded in a data format rather than a procedural format. Since production rules can formally isolate the dependence of language form on domain structure, they serve to separate knowledge from use in the desired way.

REFERENCES

1 Abelson, R. Concepts for representing mundane reality in plans. In D. Bobrow and A. Collins (Eds.), *Representation and Understanding*, New York, Academic Press, 1975, pp. 273-309.

2 Balzer, R. Human use of world knowledge. USC/Information Sciences Institute, RR 73-7, March 1974.

3 Balzer, R., Goldman, N., and Wile, D. Informality in program specifications. USC/Information Sciences Institute, ISI/RR-77-59, 1977.

4 Bobrow, D. Natural language input for a computer problem-solving system. In M. Minsky (Ed.), *Semantic Information Processing*, MIT Press, Cambridge, Mass., 1968.

5 Bobrow, D. and Winograd, T. An overview of KRL, a knowledge representation language. Xerox Palo Alto Research Center, CSL-76-4, July 1976.

6 Celce-Murcia, M. Verb paradigms for sentence recognition. *American Journal of Computational Linguistics, 1*, 1976.

7 Chamberlin, D. Relational data base management systems. *Computing Surveys, 8*, 1976, 43-66.

8 Chodorow, M. and Miller, L. The interpretation of temporal order in coordinate conjunction. IBM, RC 6199, September 1976.

9 Codd, E.F. A relational model of data for large shared data banks. *Communications of the ACM, 13*, 1970, 377-387.

10 Goldman, N. Conceptual Generation. In R. Schank (Ed.) *Conceptual Information Processing*, American Elsevier, New York, 1975.

11 Grossman, R. Some data base applications of constraint expressions. MIT Laboratory for Computer Science, TR-158, 1976.

12 Hayes, J.R. and Simon, H. Understanding written problem instructions. In Gregg (Ed.) *Knowledge and Cognition*, Lawrence Erlbaum Associates, Potomac Md., 1974.

13 Kaplan, R. A general syntactic processor in natural language processing. In R. Rustin (Ed.), *Natural Language Processing*, Algorithmics Press, New York, 1973, pp. 193-241.

14 Myer, T. and Barnaby, J. Tenex Executive Manual. Bolt Beranek and Newman, Inc., April 1973.

15 Riesbeck, C. and Schank, R. Comprehension by computer: analysis of sentences in context. Yale University, Dept. of Computer Science, Research Report #78, October 1976.

16 Ruth, G. Protosystem I: an automatic programming system prototype. MIT Laboratory for Computer Science, TM-72, 1976.

17 Schank, R. and Abelson, R. Scripts, Plans, and Knowledge. *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, MIT Artificial Intelligence Laboratory, 1975, 151-157.

Partial Matching in Pattern-Directed Inference

The Role of Partial and Best Matches in Knowledge Systems

Frederick Hayes-Roth
The Rand Corporation Santa Monica, CA 90406

Partial matching is a comparison of two or more descriptions that identifies their similarities. Determining which of several descriptions is most similar to one description of interest is called the best match problem. Partial and best matches underlie several knowledge system functions, including: analogical reasoning, inductive inference, predicate discovery, pattern-directed inference, semantic interpretation, and speech and image understanding. Because partial matching is both combinatorial and ill-structured, admissible algorithms are elusive. Economical solutions require very effective use of constraints that, apparently, can be provided only by globally organized knowledge bases. Examples of such organizations are provided, and promising avenues of research are proposed.

Production Systems with Feedback

Steven W. Zucker
Department of Electrical Engineering
McGill University Montreal, H3C 3G1 Canada

Production systems provide a methodology for using knowledge embedded in (condition A action) pairs. An action is performed when its conditions match the state of a current data base. When multiple conditions match the data base simultaneously, however, the problem arises of which productions should be executed. Such multiple matches can derive from ambiguity and uncertainty in the data base and condition patterns. One solution to this multiple matching problem is to extend the production system with a form of feedback. This paper describes the implementation of feedback directed toward global data base consistency. Specific examples are chosen from a low-level vision system.

The Production System: Architecture and Abstraction

Stanley J. Rosenschein
Technion--Israel Institute of Technology
Haifa, Israel

The use of production systems as the primary method for encoding knowledge in large knowledge-based systems is discussed at two levels; their suitability as an architecture that can be efficiently supported and their appropriateness as a language of expression. Questions of efficiency are posed in the framework of a broad class of pattern-directed rewrite systems. Factors governing efficiency are discussed informally, and the usefulness of production systems as an information processing abstraction is examined critically. In this regard, several problems suggested by work on lexically motivated inference are described. It is argued that the use of a particular class of production systems demands a more detailed justification in domain-specific terms than is often given.