CUCS-54-83

RESEARCHER:
An Overview

Michael Lebowitz

May, 1983

RESEARCHER: An Overview[1]

Michael Lebowitz
Department of Computer Science
Computer Science Building, Columbia University
New York, NY 10027

Abstract

Described in this paper is a computer system, RESEARCHER, being developed at Columbia that reads natural language text in the form of patent abstracts and creates a permanent long-term memory based on concepts generalized from these texts, forming an intelligent information system. This paper is intended to give an overview of RESEARCHER. We will describe briefly the four main areas dealt with in the design of RESEARCHER: 1) knowledge representation, where a canonical scheme for representing physical objects has been developed, 2) memory-based text processing, 3) generalization and generalization-based memory organization that treats concept formation as an integral part of understanding, and 4) generalization-based question answering.

1 Introduction

Natural language processing and memory organization are logical components of intelligent information systems. At Columbia, we are developing a computer system, RESEARCHER, that reads natural language text in the form of patent abstracts (disc drive patents provide the initial domain) and creates a permanent long-term memory based on generalizations that it makes from these texts. In terms of task, RESEARCHER is similar to IPP [Lebowitz 80, Lebowitz 83a, Lebowitz 83b], a program developed at Yale, that read news stories. The need to deal with complex object representations and descriptions has introduced a whole new range of problems not considered for that program.

---

[1]Much of the work described here was carried out by a group of Computer Science PhD students, including Kenneth Wasserman, Cecile Paris, Tom Ellman and Laila Moussa, Master's students including Mark Lerner, and undergraduates including Erik Urdang and Galina Datskovsky. Comments by Kathleen McKeown on a draft of this paper were greatly appreciated. This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-82-C-0427.

In this paper we propose to give an overview of RESEARCHER. We will describe briefly the four main problems dealt with by the program: representation, text processing, generalization and memory organization, and question answering.

2 Representation

The first problem to be worked on in any new domain in AI is the design of a scheme to represent relevant concepts. AI researchers have not extensively investigated representing complex physical objects (although [Lehnert 78; Kosslyn and Shwartz 77; Norman and Rumelhardt 75; Brachman 79] have addressed some of the issues we are concerned with). We have developed a frame-based system with the flavor of Schank's Conceptual Dependency [Schank 72], that deals with objects instead of actions. This scheme is described in detail in [Wasserman and Lebowitz 83].

The basic frame-like structure used to represent objects is known as a *memette*. Memettes are used as part of a hierarchical set of prototypes, described in Section 4.[2] A given memette may be describing a fairly general object (e.g., a prototypical disc drive) or a more specific object (a model 19023 floppy disc drive). In somewhat simplified form, the basic structure of a memette is shown in Figure 1.

(NAME: <name-of-object>
TYPE: *unitary* or *composite*
STRUCTURE: <shape-descriptor> if *unitary*
or
<a list of relation records> if *composite*)

Figure 1: Representation Schema

The NAME slot of a memette is simply the name of the physical object being described, if known. The TYPE slot indicates whether this is a single indivisible structure (*unitary*) or a conglomeration of two or more pieces (*composite*). The STRUCTURE field contains either a description of the shape of an object, if it is

[2] The term *prototype* is used here to refer to a generalized, idiosyncratic description derived from specific instances.

unitary, or a set of relation records, if it is composite. Shape-descriptors are graphical representations of objects based mostly on visual properties. Relations are the key to this object representation scheme; each one is generally a binary relation between parts of the complex object.

To date we have not explored shape descriptors in great detail. We expect to have a system that uses prototypical shapes (in much the same way we use prototypical object descriptions), combining declarative and image-like representations in much the same way as Kosslyn and Shwartz's model [Kosslyn and Shwartz 77].

We have studied object relations in much greater detail, and have developed a canonical scheme for describing the various ways that two objects can relate to each other. This scheme is used to represent, among other things, the meanings of words or phrases such as "above", "on top of" and "surrounding" that are used to describe physical relations.

Space does not permit a complete description of our relation representation scheme, which is fully described in [Wasserman and Lebowitz 83]. Figure 2 shows the major elements used in relation representation. Various combinations of values for the fields shown in Figure 2 provide wide coverage of the kinds of relations that objects can have with each other.

Certain combinations of relation fields occur together often enough that relations like objects and shapes, can frequently be described, both in text and our representations, in terms of prototypes. The normal way to represent an object is in terms of prototypical relations (such as ON-TOP-OF and SURROUNDS) that are in turn represented canonically with the fields in Figure 2.

As an example of how our representation scheme is used, consider the following sentence, taken from an abstract of a US patent about a computer disc drive

PROPERTY	DESCRIPTION	VALUE(S)
distance	distance between objects (e.g., near, remote)	0 - 10 0 - close, 10 - far
contact	strength of contact (e.g., touching, affixed)	-10 to 10 -10 - close, 10 - loose
location	relative direction between objects (e.g., above, left)	2D or 3D angle with reference frame
orientation	relative object orientation (e.g., parallel, perpendicular)	2D or 3D angle
enclosure	description of full or partial enclosure (e.g., encircled, cornered)	"full" or "partial" plus an enclosure (shape) description

Figure 2: Canonical Relation Fields

PAT1 - Enclosed Disc Drive having Combination Filter Assembly

A combination filter system for an enclosed disc drive in which a breather filter is provided in a central position in the disc drive cover and a recirculating air filter is concentrically positioned about the breather filter

A possible memette structure for this patent in shown in Figure 3.

The basic idea here is that we have a set of objects related to each other by prototypical relations (which can be broken down into their canonical components in order to make low-level inferences, when needed). Note that some of the information shown in Figure 3 is not stated explicitly in PAT1. For example, the *case* is specified as a unitary memette; since virtually nothing was said about the enclosure, this information was assumed by the reader (from knowledge of prototypical cases). The structure of the *case* is assumed to be box-shaped and open on top (the latter fact was implied by the existence of a *cover*). Likewise the *disc-drive* itself is considered to be composite, although this information would have had to be acquired outside the context of this example.

Four prototypical relations, with their corresponding role fillers are used in this small memette structure: ON-TOP-OF, SURROUNDS and SURROUNDS[centrally]

```
(NAME: enclosed-disc-drive-with-filter
 TYPE: composite
 STRUCTURE: ((SURROUNDS enclosure disc-drive)))

(NAME: enclosure
 TYPE: composite
 STRUCTURE: ((ON-TOP-OF cover case)))

(NAME: case
 TYPE: unitary
 STRUCTURE: (box open-on-top))

(NAME: disc-drive
 TYPE: composite
 STRUCTURE: unknown)

(NAME: cover
 TYPE: composite
 STRUCTURE: ((SURROUNDS[centrally] cover breather-filter)
             (SURROUNDS[centrally] recirculating-air-filter
                                   breather-filter)))

(NAME: breather-filter
 TYPE: unknown)

(NAME: recirculating-air-filter
 TYPE: unknown)
```

Figure 3: Representation for PAT1

(twice). These define the relations among the case, cover, and various filters that are described in PAT1.

Unitary memettes do not contain any relation records under their STRUCTURE property; instead, they have a single shape-descriptor. "Box open-on-top" was given as the shape-descriptor of the *case*. This is not a particularly functional piece of information. As yet there has be no strong need to codify shape-descriptors.

3 Text Processing

RESEARCHER begins its processing of a patent by determining from the text a conceptual representation of the kind described in Section 2. In the ultimate version of the program, this process will be strongly integrated with memory search and generalization. The conceptual analysis performed by RESEARCHER is based on the memory-based understanding techniques designed for IPP [Lebowitz 83b]. This processing involves a top-down goal of recognizing structures in memory integrated with simple, bottom-up syntactic techniques.

Naturally, since patents are quite different from news stories, both because they describe complex physical objects and because they make considerable use of special purpose language, the precise techniques used in RESEARCHER are distinct from those used in IPP. RESEARCHER is still predictive in nature. However, since patents are not focused on events, as are news stories, the action-based predictions of IPP (or systems such as ELI [Riesbeck and Schank 76], CA [Birnbaum and Selfridge 81] and FRUMP [DeJong 79]) must be extensively modified.

Specifically, the predictions used for understanding in RESEARCHER are based on the physical descriptions built up, in much the same way as IPP made predictions from events. The goal of RESEARCHER's understanding process is to record in memory how a new object being described differs from generalized objects already known (keeping in mind that these are idiosyncratic), and ultimately to generalize new prototypes.

Processing in RESEARCHER concentrates on words that refer to physical objects in memory and words that describe physical relations between such objects. Such words are known as Memory Pointers (MPs) and Relation Words (RWs). These words guide RESEARCHER's processing, and make use of any information gathered bottom-up, in much the same way as IPP used S-MOPs and Action Units [Lebowitz 80]. Conceptual analysis in this domain involves careful processing of MP phrases (usually noun phrases) to identify memettes, modifications to memettes, and repeated mentions of memettes. RWs are used to create the relations between memettes described in section 2.

Particular care in this domain has to be given to phrases of the sort "X relation1 Y relation2 Z". It is frequently hard to tell if relation2 relates Z to Y or X. So, in the phrase "read/write head above a disc connected to a cable" it is not apparent from the surface structure whether the disc or the read/write head is connected to the cable. Prepositional phrase attachment is a well-known problem, and is especially crucial in the patent domain. We have discovered that a set of heuristics that maintains a single memette in focus (as in [Grosz 77; Sidner 79]) based on the memettes and relations involved will solve most of these problems (although in the long run we expect to use, in addition, a model of the device being described). In

the example mentioned, a spatial relation between concrete objects tends to leave the first in focus, and hence the second relation ("connected to", here) relates the read/write head to the cable. Such heuristics are typical of the style of processing used in RESEARCHER.

Figure 4 shows the output from RESEARCHER's processing of the initial part of PAT1, illustrating the kind of processing involved.

```
*(process-patent PAT1)

Running RESEARCHER at 6:22:03 PM
Patent: PAT1

(A COMBINATION FILTER SYSTEM FOR AN ENCLOSED DISC DRIVE IN WHICH A BREATHER
FILTER IS PROVIDED IN A CENTRAL POSITION IN THE DISC DRIVE COVER AND A
RECIRCULATING AIR FILTER IS CONCENTRICALLY POSITIONED ABOUT THE BREATHER
FILTER *STOP*)

Processing:

A              : New instance word -- skip
COMBINATION    : Token refiner - save and skip
FILTER         : Token refiner - save and skip
SYSTEM         : MP word -- memette UNKNOWN-ASSEMBLY#
New UNKNOWN-ASSEMBLY#0 instance (&MEM0) [&MEM0]
Assuming FILTER# is part of &MEM0 (UNKNOWN-ASSEMBLY#0) in this case
New FILTER# instance (&MEM1)
Augmenting &MEM0 (UNKNOWN-ASSEMBLY#0) with feature: CONFIGURATION = COMPLEX
FOR (FOR3)     : Assuming &MEM0 (UNKNOWN-ASSEMBLY#0) is part of the following
AN             : New instance word -- skip
ENCLOSED       : Relation word -- save and skip
DISC DRIVE     : Phrase
-> DISC-DRIVE : MP word -- memette DISC-DRIVE#
Assuming &MEM0 (UNKNOWN-ASSEMBLY#0) is part of DISC-DRIVE# in this case
New DISC-DRIVE# instance (&MEM2)
New ENCLOSURE# instance (&MEM3)
Relating memettes &MEM3 (ENCLOSURE#) (SUBJECT) &MEM2 (DISC-DRIVE#) (OBJECT)
 [R-SURROUNDS]

<rest of processing>
```

Figure 4: RESEARCHER Processing PAT1

In the output trace in Figure 4, we can see how RESEARCHER identifies the various objects mentioned in the text as instances of general structures described in memory (such as DISC-DRIVE# and FILTER#) RESEARCHER creates new memettes to represent the specific instances of these structures, &MEM0 for the "system", for example, and records how these instances differ from the abstract prototypes.

We can also see in this example how RESEARCHER processing concentrates on object description and relations among objects. In its processing of "filter system", RESEARCHER uses a heuristic that in MP-MP constructs, where either MP describes a non-specific complex object (e.g., "system"), the other object is probably part of that complex object. Also note that the word "enclosed", though technically an adjective here, is still treated as a Relation Word (most of which are prepositions), describing a relation between an implicit enclosure which surrounds the object to follow (the disc drive, in this case). Finally, we note that the word "for", in this context, serves as a flag indicating that the parts of the preceding object, the filter system, are to follow.

The full processing of PAT1 leads to the representation shown in Figure 5.

```
Text Representation:

A list of relations:

Subject:              Relation:          Object:
'SYSTEM'              R-PART             FILTER#
DISC-DRIVE#           R-PART             'SYSTEM'
'SYSTEM'              R-PART             BREATHER-FILTER#
DISC-DRIVE#           R-PART             COVER#
'SYSTEM'              R-PART             RECIRCULATING-FILTER#
ENCLOSURE#            R-SURROUNDS        DISC-DRIVE#
COVER#                R-SURROUNDS        BREATHER-FILTER#
                      LOCATION:CENTER
RECIRCULATING-FILTER# R-SURROUNDS        BREATHER-FILTER#
                      LOCATION:CENTER

[Journal end: P13.JOURNAL.2, Mon 28 Mar 83 6:22:32PM]
```

Figure 5: RESEARCHER Representation of PAT1

The output in Figure 5 indicates that RESEARCHER has identified the important physical relations mentioned in PAT1. It basically includes all the relations shown earlier in Figure 3, which was our target representation for PAT1. In the complete text of PAT1, many more relations are described, which is not surprising, considering the complexity of the device in question.

Two technical notes are in order about Figure 5. The first is that the relations built up do not actually relate the abstract memettes (such as FILTER#) to each other, but rather instances of them (such as &MEM1). The abstract names have

been used for readability. Secondly, the R-PART relations that are shown are intended to capture cases where we know an object is a part of a complex assembly, but do not know the precise physical relationship. Presumably, these relations could be replaced later (either from further text or from memory), with more specific relations.

4 Memory Organization and Generalization

In order to store for later query information about the patents that are read, RESEARCHER makes use of *Generalization-Based Memory*. This method, which was developed for IPP [Lebowitz 83a] and is related to Schank's Memory Organization Points [Schank 82], involves storing information about given items in memory in terms of generalized prototypes. The idea is that we locate the prototype in memory that best describes an example, and then store only how the example varies from the prototype. This allows redundant information to be stored only once. It also allows questions to be answered in terms of descriptive prototypes.

For Generalization-Based Memory to be effective, it is not adequate to simply make use of pre-specified prototypes. It is necessary for the system to create new prototypes through a generalization process. This process involves identifying similar objects and creating new concepts from them (using a comparison technique of the sort used by IPP [Lebowitz 83a], and related to traditional "learning from examples" programs such as those in [Winston 72; Winston 80; Langley 81; Quinlan 79; Mitchell 78].

In the disc drive domain, typical concepts the generalization process might identify as being useful would be floppy disc drives or double sided discs. Crucially RESEARCHER must do this without being specifically provided with examples of these concepts. Instead, when storing instances from a stream of input, it would store floppy disc drives together in its Generalization-Based Memory, and notice the similarities among them.

The representations for two similar, slightly simplified disc drive patents, used to test the initial version of RESEARCHER's generalization module are shown in Figure 6.

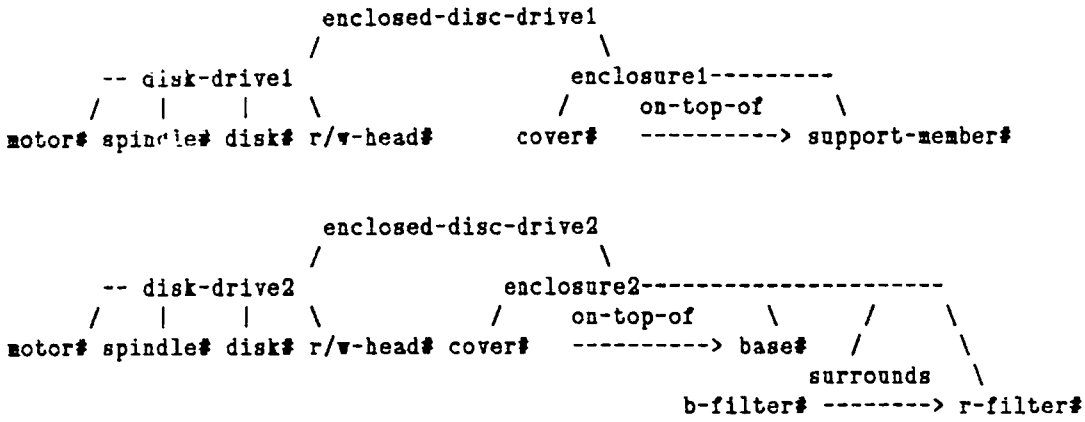

Figure 6: Similar Disc Drives

Clearly the two disc drives in Figure 6 have much in common that can be the source of a new concept derived through generalization -- an enclosed disc drive Figure 7 shows the concept created by RESEARCHER's generalization module.

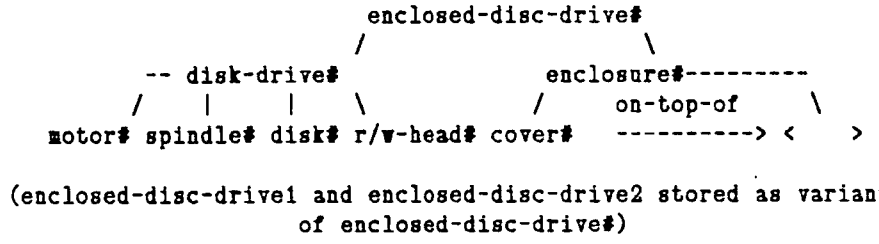

Figure 7: Generalized Enclosed Disc Drive

The idea illustrated in Figure 7 is that RESEARCHER finds the parts of two objects that are similar, and abstracts them out into a generalized concept. In this example, the two devices contained similar disc drives and enclosures. Each had a cover on top of some other object. So these similarities form the basis of a generalized enclosed disc drive. Only the additional parts and relations of each instance need be recorded in memory along with the generalization.

Adapting Generalization-Based Memory for use on structural descriptions of the sort described in Section 2 has proved to be a complex and difficult problem, revolving around the assorted relations among the objects in the descriptions. Here we will only present one of the major problems and suggest the nature of the possible solution.

The central problem in generalizing structural descriptions is the process of matching two representations (either of two objects or an object and a prototype), determining what parts and relations correspond (as was pointed out for simpler examples in [Winston 72]). Clearly, if we wish to determine that the disk mounts in two drives are similar, they must be compared with each other. Since, as mentioned in Section 2, the central part of the description of complex objects is a set of relations, we must associate the relations in one object with those in the other.

The matching process here is quite a difficult one. The main problem is that we are dealing with structured objects, and the parts of very similar objects may be aggregated differently in various descriptions. So, for example, a read/write head might be described as a direct part of a disc drive in one patent, but part of a "read/write assembly" in another. This makes the inherent similarity hard to identify.

At the moment, we deal with this "level problem" with simple heuristics that allow only a limited amount of "level hopping" during the comparison process (to avoid the need to consider every possible correspondence among levels), and a bit of combinatoric force. However, we feel that the ultimate solution lies in more extensive use of Generalization-Based Memory. If a new object can be identified as an instance of a generalized concept, with only a few minor differences (which will be done with a discrimination-net-based search of the sort described in [Lebowitz 83a]), then the levels of aggregation will be set. In effect, the existing concepts create a canonical framework for describing new objects. In addition, by using Generalization-Based Memory, we need compare only a small number of *differences* between objects, rather than complex descriptions.

5 Question Answering

The presentation of information from a complex set of data in order to answer user questions is an interesting problem in its own right (as has been pointed out by many researchers, including [Lehnert 77; McKeown 82]). As part of RESEARCHER, we have included a question answering module that concentrates on taking advantage of Generalization-Based Memory to more effectively convey information to a user. Here we can only provide a flavor of the approach we are taking.

RESEARCHER accepts questions in natural language format. It uses the same parser used to process texts to create a conceptual representation of the question. This is much the same approach as taken in BORIS [Dyer 82]. Also in similar fashion to BORIS, we eventually expect the question parsing process to identify actual structures in memory, greatly simplifying the answering process.

Once RESEARCHER has developed a conceptual representation of a question, it searches memory to find an answer using an approach similar to [Lehnert 77]. That is, a set of heuristics is used to decide upon the type of question and what constitutes a reasonable answer. The answer heuristics focus on using generalizations that occur in memory to quickly convey large amounts of information, and then describing how particular instances may differ from the generalizations. We are also looking at how generalization-based heuristics might aid in determining what aspects of very complex representations to try and convey to a questioner.

6 Conclusion

The development of RESEARCHER has lead to interesting results in a number of areas. Natural language that involves complex physical objects is an exciting topic, one that can lead to many interesting applications. We believe that the representation scheme described here, the application of memory-based parsing and Generalization-Based Memory, as well as generalization-based question answering will all help lead to the successful development of powerful, robust, dynamic understanding systems.

References

[Birnbaum and **Selfridge 81**] Birnbaum, L. and Selfridge, M. Conceptual analysis of natural language. In R. C. Schank and C. K. Riesbeck, Ed., *Inside Computer Understanding*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1981. pp. 318 - 353.

[**Brachman 79**] Brachman, R. J. Taxonomy, descriptions and individuals in natural language processing. Proceedings of the 17th Annual Meeting of the Association for Computational Linguistics, La Jolla, California, 1979, pp 33 - 38

[**DeJong 79**] DeJong, G. F. Skimming stories in real time: An experiment in integrated understanding. Technical Report 158, Yale University Department of Computer Science, 1979.

[**Dyer 82**] Dyer, M. G. In-depth understanding: A computer model of integrated processing for narrative comprehension. Technical Report 219, Yale University Department of Computer Science, 1982.

[**Grosz 77**] Grosz, B. J. Representation and use of focus in a system for understanding dialogs. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, International Joint Conference on Artificial Intelligence, Cambridge, MA, 1977.

[**Kosslyn and Shwartz 77**] Kosslyn, S. M. and Shwartz, S. P. "A Simulation of Visual Imagery." *Cognitive Science 1* (1977), 265 - 295.

[**Langley 81**] Langley, P. "Data-driven discovery of natural laws." *Cognitive Science 5*, 1 (1981), 31 - 54.

[**Lebowitz 80**] Lebowitz, M. Generalization and memory in an integrated understanding system. Technical Report 186, Yale University Department of Computer Science, 1980. PhD Thesis.

[**Lebowitz 83a**] Lebowitz, M. "Generalization from natural language text." *Cognitive Science 7*, 1 (1983), 1 - 40.

[**Lebowitz 83b**] Lebowitz, M. "Memory-based parsing." *Artificial Intelligence 21* (1983), 285 - 326.

[**Lehnert 77**] Lehnert, W. G. *The Process of Question Answering.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

[**Lehnert 78**] Lehnert, W.G. Representing physical objects in memory. Technical Report 131, Yale University Department of Computer Science, 1978.

[**McKeown 82**] McKeown, K. R. *Generating natural language text in response to questions about database structure.* Ph.D. Thesis, University of Pennsylvania, 1982.

[**Mitchell 78**] Mitchell, T. M. Version spaces: An approach to concept learning. Technical Report HPP-79-2, Stanford University, Stanford, California, 1978.

[**Norman and Rumelhardt 75**] Norman, D. A. and Rumelhart, D. E. *Explorations in Cognition.* W. H. Freeman and Company, San Francisco, CA, 1975

[**Quinlan 79**] Quinlan, J. R. Induction over large data bases. Technical Report HPP-79-14, Stanford University Computer Science Department, 1979.

[**Riesbeck and Schank 76**] Riesbeck, C. K. and Schank, R. C. Comprehension by computer: Expectation-based analysis of sentences in context. In W. J. M. Levelt and G. B. Flores d'Arcais, Ed., *Studies in the Perception of Language.* John Wiley and Sons, Chichester, England, 1976.

[**Schank 72**] Schank, R. C. "Conceptual Dependency: A theory of natural language understanding." *Cognitive Psychology 3*, 4 (1972), 532 - 631.

[**Schank 82**] Schank, R. C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, New York, 1982

[**Sidner 79**] Sidner, C. L. *A Computational model of co-reference comprehension in English.* Ph.D. Thesis, Massachussets Institute of Technology, Cambridge, MA 1979

[**Wasserman and Lebowitz 83**] Wasserman, K. and Lebowitz, M. "Representing complex physical objects." *Cognition and Brain Theory 6*, 3 (1983) 333-352.

[**Winston 72**] Winston, P. H. Learning structural descriptions from examples. In P. H. Winston, Ed., *The Psychology of Computer Vision*, McGraw-Hill, New York 1972.

[**Winston 80**] Winston, P. H. "Learning and reasoning by analogy." *Communications of the ACM 23* (1980), 689 - 702.