# LEARNING STRATEGIES AND AUTOMATED KNOWLEDGE ACQUISITION: AN OVERVIEW

by

*Ryszard S. Michalski*

Chapter in the Book "Computational Models of Learning",
Edited by Leonard Bolc.

SYMBOLIC COMPUTATION

Leonard Bolc (Ed.)

# Computational Models of Learning

With Contributions by
G. L. Bradshaw P. Langley R. S. Michalski
S. Ohlsson L. A. Rendell H. A. Simon J. G. Wolff

Springer-Verlag

# SYMBOLIC COMPUTATION

*Artificial Intelligence*

Springer Series
SYMBOLIC COMPUTATION - *Artificial Intelligence*

---

N. J. Nilsson: Principles of Artificial Intelligence. XV, 476 pages, 139 figs., 1982

J. H. Siekmann, G. Wrightson (Eds.): Automation of Reasoning 1. Classical Papers on Computational Logic 1957-1966. XXII, 525 pages, 1983

J. H. Siekmann, G. Wrightson (Eds.): Automation of Reasoning 2. Classical Papers on Computational Logic 1967-1970. XXII, 638 pages, 1983

L. Bolc (Ed.): The Design of Interpreters, Compilers, and Editors for Augmented Transition Networks. XI, 214 pages, 72 figs., 1983

R. S. Michalski, J. G. Carbonell, T. M. Mitchell (Eds.): Machine Learning. An Artificial Intelligence Approach. XI, 572 pages, 1984

L. Bolc (Ed.): Natural Language Communication with Pictorial Information Systems. VII, 327 pages, 67 figs., 1984

J. W. Lloyd: Foundations of Logic Programming. Second, extended edition, XII, 212 pages, 1987

A. Bundy (Ed.): Catalogue of Artificial Intelligence Tools. Second revised edition, IV, 168 pages, 1986

M. M. Botvinnik: Computers in Chess. Solving Inexact Search Problems. With contributions by A. I. Reznitsky, B. M. Stilman, M. A. Tsfasman, A. D. Yudin. Translated from the Russian by A. A. Brown. XIV, 158 pages, 48 figs., 1984

C. Blume, W. Jakob: Programming Languages for Industrial Robots. XIII, 376, 145 figs., 1986

L. Bolc (Ed.): Natural Language Parsing Systems. XVIII, 367 pages, 155 figs., 1987

L. Bolc (Ed.): Computational Models of Learning. IX, 208 pages, 34 figs.,. 1987

# Computational Models of Learning

Edited by Leonard Bolc

With Contributions by
G. L. Bradshaw P. Langley R. S. Michalski
S. Ohlsson L. A. Rendell H. A. Simon
J. G. Wolff

With 34 Figures

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo

**Leonard Bolc**
**Institute of Informatics, Warsaw University**
**PKiN, pok. 850, PL-00-901 Warszawa, Poland**

ISBN 3-540-16318-2 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-16318-2 Springer-Verlag New York Berlin Heidelberg

Library of Congress Cataloging in Publication Data.
Computational models of learning.
(Symbolic computation. Artificial intelligence) Includes index.
Contents: Learning strategies and automated knowledge acquisition / R. S. Michalski - Heuristics for empirical discovery / P. Langley, H. A. Simon, and G. L. Bradshaw - Transfer of training in procedural learning / S. Ohlsson - [etc.]
1. Machine learning. 2. Artificial intelligence.
I. Bolc, Leonard, 1934- . II. Bradshaw, G. L. III. Series.
Q325.C626 1987 006.3'1 87-16527
ISBN 0-387-16318-2 (U.S.)



Typesetting, printing and binding: Appl, Wemding
2145/3140-543210

# Table of Contents

# Learning Strategies and Automated Knowledge Acquisition

## An Overview

Ryszard S. Michalski[1]

**Abstract**

Fundamental learning strategies are discussed in the context of knowledge acquisition for expert systems. These strategies reflect the type of inference performed by the learner on the input information in order to derive the desired knowledge. They include learning from instruction, learning by deduction, learning by analogy and learning by induction. Special attention is given to two basic types of learning by induction: learning from examples (concept acquisition) and learning from observation (concept formation without teacher). A specific form of learning from observation, namely, *conceptual clustering*, is discussed in detail, and illustrated by an example. Conceptual clustering is a process of structuring given observations into a hierarchy of conceptual categories.

An inductive learning system generates knowledge by drawing inductive inferences from the given facts under the guidance of *background knowledge*. The background knowledge contains previously learned concepts, goals of learning, the criteria for evaluating hypotheses from the viewpoint of these goals, the properties of attributes and relations used to chracterize observed events, and various inference rules for transforming concepts or expressing them at different levels of abstraction.

## 1. Introduction

Learning ability is no doubt central to human intelligence. This ability permits us to adapt to the changing environment, to develop a great variety of skills, and to acquire expertise in an almost unlimited number of specific domains. The human ability to learn is truly remarkable: people are capable of learning from information carried by multiple physical media and expressed in an unbounded variety of forms. This information can be stated at different levels of abstraction, with different degrees of precision, with or without errors, and with different degrees of relevancy to the knowledge ultimately acquired.

Implanting learning capabilities in machines is one of the central goals of Artificial Intelligence. It is the subject of a new field of Machine Learning. Due to the enormous complexity of learning processes, development of general-purpose, versatile learning systems is a long-term goal. With the development of expert systems, however, implementing some forms of machine learning has become an urgent task, even if the forms of such implementation are very limited.

The urgency of this task stems from an explosive growth of interest and social need to develop expert systems for many different applications, from medicine and agriculture to law, education and computer design. Expert systems are com-

[1] Department of Computer Science, University of Illinois, Urbana, Illinois 61801

puter programs (or devices) that simulate the expertise of a human expert in solving problems in some specific domain. They are capable of conducting formal inference on their knowledge base in the interaction with the external information provided by a user, in order to provide a solution to a problem or an advice in decision making. Examples of some early expert systems include:

- DENDRAL (developed at Stanford University) for determining the molecular structure of organic compounds from mass spectrograms.
- MACSYMA (developed at MIT) which serves as a general mathematical aids system (e.g., for symbolic integration, simplification of mathematical expressions, etc.).
- R1 (developed at Carnegie-Mellon University) for determining configurations of VAX computer systems.
- INTERNIST (developed at the University of Pittsburgh) for diagnosing diseases of interest in internal medicine.
- PLANT/ds and PLANT/cd (developed at the University of Illinois) - two related agricultural expert systems, the first for diagnosing soybean diseases, and the second for predicting black cutworm damage to corn.

The major component of an expert system is its knowledge base, i.e., formally represented knowledge in the given domain of application. Building such a knowledge base is typically done as a cooperative effort between a "knowledge engineer" and a domain expert. The knowledge engineer conducts interviews with an expert and codifies the expert's knowledge in some knowledge representation system.

Such a system often consists of production rules (condition-action rules) or a semantic network. A semantic network is a graph whose nodes represent concepts and whose links represent relations between the concepts. These two forms of knowledge representation have special appeal, because of their comprehensibility and relative ease of use for implementing inference processes. For some applications, however, these forms may not be sufficient. For example, in system PLANT/cd, a large part of the domain knowledge is encoded as a set of procedures that form a simulation model of the growth of corn and the growth of black cutworms (Boulanger, 1983).

Encoding expert knowledge into a system is a time-consuming, difficult process that is prone to error. For this reason, knowledge acquisition is a "bottleneck" in the development of expert systems. The process of knowledge acquisition can be simplified by applying interactive programming aids for developing and debugging rule bases. Such an aid is provided, for example, by the system TEIRESIAS, developed by Davis (1978). A long term solution, however, is seen in the development of machine learning. The importance of the field of machine learning to further progress in the development of expert systems has been indicated by many authors (e.g., Waltz et al., 1983).

In this paper we review basic strategies of learning and discuss them in the context of automated knowledge acquisition. We specifically concentrate on knowledge acquisition through inductive learning. The latter encompasses two strategies: learning from examples, and learning by observation and discovery.

## 2. Fundamental Learning Strategies

The knowledge acquisition process can be greatly simplified if an expert system can learn decision rules from examples of decisions made by human experts, or from its own errors. This type of learning strategy is called *learning from examples* (or *concept acquisition*). It has been studied widely in the last ten years or so, and many important results have been obtained (e.g., Winston, 1970; Michalski, 1972; Lenat, 1976; Mitchell, 1978; Buchanan et al. 1979; Pao and Hu, 1982, Hu and Pao, 1982; Dietterich and Michalski, 1983; Langley, Bradshaw and Simon, 1983; Michalski, 1983; Reinke, 1984; Quinlan, 1986; Winston, 1986).

Learning from examples is one of several fundamental learning strategies. These strategies are identified by viewing a learning system as an inference system. Namely, they are distinguished by the major type of inference the learning system (human or machine) performs on the information provided, in order to derive the desired kowledge. At one extreme, the system performs no inference, but directly accepts and uses the information given to it (or built into it). At the other extreme, the system performs a complex, search-based inductive inference that on occasion leads to discovery of new knowledge. The following learning strategies are important points along the above spectrum:

### *A. Direct Implanting of Knowledge*

This strategy requires little or no inference on the part of the learner. It includes rote learning, learning by imitation, learning by being constructed or by being programmed. This strategy is a widely used method for providing knowledge to a computer system: we incorporate knowledge into its hardware, we program it, and we build databases for all kinds of applications. Although building databases is not typically considered as machine learning, it can be considered as a special case of such a process. Some databases go beyond this learning strategy, if they can perform some amount of inference, usually mathematical or statistical.

### *B. Learning from Instruction*

In this form of learning, also called *learning by being told,* a learner selects and transforms the knowledge from the input language to an internally-usable representation and integrates it with prior knowledge for effective retrieval and use. This is the most widely used strategy of human learning: it includes learning from teachers, books, publications, exhibits, displays, and similar sources. A machine version of this strategy is a system capable of accepting instruction or advice and applying the learned knowledge effectively to different tasks. Simple versions of this strategy constitute the basic method for providing knowledge to expert systems today (e.g., Davis, 1978, Hass and Hendrix, 1983).

## C. Learning by Deduction

A learning system that uses this strategy conducts deductive (truth-preserving) inference on the knowledge it possesses and knowledge supplied to it. This is done in order to restructure given knowledge into more useful or more effective forms, or to determine important consequences of the knowledge. For example, given a set of numbers: 1, 2, 6, 24, 120, 720, a learning system might represent them in an equivalent, but shorter form as $n!$, $n = 1 \ldots 6$. To do so, the system must, of course, know the concept of a factorial.

A form of deductive learning, called *analytical* or *explanation-based learning,* has recently become an active research area. In analytical learning, the system is already equipped with a description of the target concept, but the description is expressed at the level of abstraction too high to be directly usable (operational). The system uses the domain knowledge to determine or explain why a given fact is an example of the concept. This process takes a form of formal proof, and produces a new concept description that is operational. This typically means that the concept is reexpressed in terms of properties used in the concept example.

As an illustration, consider a system that already knows that a cup is a stable, open, liftable vessel. Suppose that it is now presented a specific instance of a cup, described in terms such as an upward concavity, flat bottom, the presence of a handle, color, size, and other features. By using the domain knowledge that links the known high level concept description of the cup with the features used in the instance, the system constructs an operational description stating that a cup is an upward concave object with a flat bottom and a handle (Mitchell, Keller and Kedar-Cabelli, 1986; DeJong and Mooney, 1986).

The explanation-based learning is a useful technique, applicable to many problems. In order to be used, however, the system has to be equipped with a sufficient amount of relevant domain knowledge. This domain knowledge has to be inputed to the system somehow - either by handcrafting it to the system, or by analogical or inductive learning.

## D. Learning by Analogy

This strategy involves transforming or extending existing knowledge (or skill) applicable in one domain to perform a similar task in another domain. For example, the learning-by-analogy strategy might be applied to learn water skiing when a person already knows snow skiing. Learning by analogy requires a greater amount of inference on the part of the learner than does learning from instruction. Relevant knowledge or skill must be retrieved from the memory and appropriately transformed to be applicable in a new situation or to a new problem. Examples of systems capable of learning by analogy are described by Carbonell (1983), Winston (1984) and Burstein (1984).

## E. Learning from Examples

Given a set of examples and (optionally) counter-examples of a concept, the learner induces a general concept description. The amount of inference performed by the learner is greater than in learning by deduction or analogy, because the learner does not have prior knowledge of the concept to be learned, or knowledge of a similar concept. Thus, it cannot create the desired knowledge by deduction, or by analogy to what it already knows. The desired knowledge must be created anew by drawing inductive inference from available examples or facts, i.e., by inductive learning. Learning from examples, also called *concept acquisition*, can be a one-step (batch) process or a multi-step (incremental) process. In the batch case, all examples are presented at once. In incremental learning, examples (positive or negative) are introduced one-by-one or in small groups; the learner forms one or more tentative hypotheses consistent with the data at a given step, and subsequently refines the hypotheses after considering new examples. The latter strategy is commonly used in human learning.

Adaptive control systems can be viewed as a special case of systems learning from examples. A distincitve feature of them is that they improve their performance by adjusting internal parameters rather then by structural changes.

Examples of a concept may be provided by a human teacher, by the environment in which the system operates. They can be generated by a deliberate effort of a teacher, or by a random, heuristic or exhaustive search through a space of operators acting upon given situations. If an operator produces a desired result, then we have an positive example, otherwise a negative example. The inductive learning system then generalizes these examples to form general decision rules or control heuristics.

When a system determines examples by a search or other active effort, we have a form of learning called *learning by experimentation*. Such a method was used, for example, in the LEX symbolic intergration learning system (Mitchell, Utgoff and and Banerji, 1983).

Learning from examples is one form of inductive learning. Another form is learning by observation and discovery.

## F. Learning by Observation and Discovery

This "learning without teacher" strategy includes a variety of processes, such as creating classifications of given observations, discovering relationships and laws governing a given system, or forming a theory to explain a given phenomenon, or The learner is not provided with a set of instances exemplifying a concept, nor is given access to an oracle (or teacher) who can classify internally-generated instances as positive or negative. Also, rather than concentrating attention on a single concept at a time, the learner may have to deal with observations that represent several concepts. This adds a new difficulty, namely solving the focus-of-attention problem, which involved deciding how to manage the available time and resources in acquiring several concepts at once.

Learning from observation can be subclassified according to the *degree of* *interaction* between the learner and the external environment. Two basic cases can be distinguished:

(a) *passive observation,* where the learner builds a description of a given set of observations. For example, such a description may be a taxonomy of the observations (e.g., Michalski and Stepp, 1983), or an empirical law characterizing the observations, as in the BACON system (Langley, Simon & Bradshaw, 1983).

(b) *active experimentation,* where the learner makes changes in the given environment and observes the results of those changes. The changes may be random or dynamically controlled by some heuristic criteria. The choice of tasks and directions in the experimentation can be controlled by criteria such as *interestingness* (e.g., Lenat, 1976) or utility (e.g., Rendell, 1983a).

The learning strategies, (a) to (f), were presented above in order of increasing amounts of effort required from the learner and decreasing amounts effort required from the teacher. This order thus reflects the increasing difficulty of constructing a learning system capable of given learning strategy.

In human learning, the above order of strategies often reflects also an increasing confidence in the acquired knowledge. We all know that when we are given a general rule (a directive, a theory) without any explanation and examples supporting it, our confidence in it will not be very high; it will directly depend on the trust we have in the giver. Our confidence in a rule will be greater if we can try the rule on examples, and still greater, if we develop the rule through our own experience.

On the other hand, it is much more difficult to determine correct or highly useful knowledge by induction than to acquire it by instruction. This holds, of course, only if the teacher's knowledge is correct and/or highly useful, i.e., if we have a "perfect" teacher. Because this assumption may not hold in reality, the learning by instruction strategy is also associated with a risk of acquiring incorrect or low-grade kowledge. This explains the emphasis educators place on providing students with best teachers.

The higher the learning strategy, the more complex inference has to be performed by the learner, and thus the more cost and effort is involved in deriving the desired knowledge. It is much easier for the student to learn how to solve a problem by just being told the solution than by having to discover it on his/her own. Learning by instruction requires, however, a teacher who knows the algorithm or the concepts to be learned, and is capable of articulating them in the language of the learner. But when such a teacher is not available, another strategy must be used. For example, it is difficult to define the concept of a chair, or the shape of the characters of the alphabet. Therefore, such concepts are taught by showing examples rather than by instruction.

In many situations, the best way to explain a concept is to relate it to a similar concept and describe the differences. This is learning by analogy. In order to learn this way, however, the learner must know the referenced concepts. The more knowledgeable a learner is, the more potentially effective learning by analogy is. One can expect therefore that learning by analogy should tend to be in general more effective with adults than with children.

There are lessons for machine knowledge acquisition to be drawn from the above considerations. One is that if we know precisely how to solve a problem, we should tell the computer the solution directly (i.e., program it). Teaching by instruction will be simpler and more productive than using a deductive or inductive learning strategy. Such teaching will be facilitated by having an appropriate knowledge representation language and debugging tools. As there are many areas in which precise solutions are known and relevant concepts can be defined, this strategy has wide applications. Therefore, the development of appropriate knowledge representation languages and support tools (both general and specific to a given domain) constitutes a major research area.

When a learner already possesses a relevant knowledge, but the knowledge is not directly applicable to the given task, a deductive learning strategy may be applied, e.g., explanation-based learning. This strategy will produce operational, useful knowledge from an abstract, unusable knowledge. Equipping expert systems with such deductive inference capabilities, that is, with mechanisms for deductively transforming knowledge bases from one form to another, logically equivalent or more specific, is thus an important direction of research.

There are many application areas where precise concept definitions or algorithms are unknown or difficult to construct even in an abstract, non-operational form. Examples of such areas are technical, medical or agricultural diagnosis, visual pattern recognition, speech recognition, machine design, robot assembly, and many others. Also, people often have difficulties in articulating their expertise, even when they know well how to perform a given task or are able to recognize a given concept without any difficulty. In such cases, applying an analogical or inductive machine learning strategy seems quite desirable.

As mentioned earlier, a prerequisite for analogical learning in that the system possesses a knowledge base of concepts and solutions to problems that are similar to the ones the system will be solving. Moreover, the system must be able to recognize the similarity between any new problem and a problem for which it already knows a solution, and must be able to modify the known solution appropriately. These are difficult and complex operations. For that reason it is often easier for the system to start from scratch than to modify a known solution. This phenomenon is well known to programmers, who sometimes prefer to write a program anew rather than to modify an existing program that performs a task similar to the desired one. An interesting problem arising here is how to decide which way is better in any given situation. The decision requires estimates of costs involved in applying both methods in a particular situation.

Analogical inference can be viewed as a combination of inductive and deductive learning. The inductive part determines the existence of analogy between problems (or concepts) and formulates appropriate knowledge transformations that unify the base and the target problems or concepts. The deductive part performs these transformations on the known solution or concepts to derive the desired solution. An interesting variant of learning by analogy is *derivational analogy* (Carbonell, 1986) in which the experience transfer involves recreating lines of reasoning and their justifications in solving problems similar to the one encountered.

The remainder of the paper will discuss in greater detail the inductive learning strategy. Through this strategy a fundamentally new knowledge can be created, and thus this strategy is of special importance to machine learning. We will start by giving a more precise meaning to this type of learning.

## 3. Inductive Learning: General Description

Inductive learning is a process of acquiring knowledge by drawing inductive inferences from teacher- or environment-provided facts. This process involves operations of generalizing, transforming, correcting and refining knowledge representations in order to accomodate given facts and satisfy various additional criteria. An important property of inductive learning is that knowledge acquired through it cannot, in principle, except for special cases, be completely validated. This is so because inductive inference produces hypotheses with a potentially infinite number of consequences, while only a finite number of confirming tests can be performed. This is a well-known predicament of induction, already observed by Hume in the 18th century.

Inductive inference is an underconstrained problem. Given any set of facts or input premises, one can potentially generate an infinite number of hypotheses explaining these facts. In order to perform inductive inference one thus needs some additional knowledge *(background knowledge)* to constrain the possibilities and guide the inference process toward one or a few most *plausible* hypotheses. In general, this background knowledge includes the goals of learning, previously learned concepts, criteria for deciding the preference among candidate hypotheses, the methods for interpreting the observations, and the knowledge representation language with corresponding inference rules for manipulating representations in this language, as well as the knowledge of the domain of inquiry.

There are two aspects of inductive inference: the generation of plausible hypotheses, and their confirmation. Only the first is of significance to machine inductive learning. The second one (impossible in principle except for special cases) is considered of lesser importance, because it is assumed that the generated hypotheses will be judged by human experts and tested by known methods of deductive inference and statistical confirmation.

Bearing in mind these considerations, let us formulate a general paradigm of inductive inference:

*Given:*

(a) *premise statements (facts),* F, that represent initial knowledge about some objects, situations or processes,

(b) *a tentative inductive asertion* (which may be null),

(c) *background knowledge* (BK) that defines the goal of inference, the preference criterion for ranking plausible hypotheses, assumptions and constraints imposed on the premise statements and the candidate inductive assertions, and any other relevant general or domain specific knowledge.

*Find:*

*an inductive assertion (hypothesis),* H, that, together with background knowledge BK, tautologically implies the premise statements.

An hypothesis together with background knowledge tautologically implies a set of facts, if the facts are a logical consequence of the hypothesis and background knowledge, that is the implication $H \,\&\, BK \Rightarrow F$ holds under all interpretations.

Since for a given BK an infinite number of assertions H can satisfy such an implication, a *preference criterion* (also called *bias*) is used to reduce the choice to one hypothesis or a few most preferable ones. Such a criterion may require, for instance, that the hypothesis be the shortest or the most economical description of all given facts, among all candidate descriptions.

Inductive learning programs already play an important role in the acquisition of knowledge for some expert systems. In some relatively simple domains they can determine decision rules by induction from examples of decisions made by experts. This form of knowledge acquisition relieves the expert from the tedious task of defining rules himself. Moreover, it requires the expert to do only what he can do best: make decisions. Experts are typically not trained to analyze and explain to others their decision making processes, expecially if they must express them in a formal way; therefore, such tasks are usually difficult for them to perform. Once rules are acquired from examples, expert can usually do a good job in evaluating them.

A less direct yet important application of inductive learning is to the refinement of knowledge bases initially developed by human experts. Here, inductive learning programs together with other supporting software can be used to detect and rectify inconsistencies, to remove undesirable redundancies, to cover gaps or to re-express the given rules in a simpler way (e.g., Reinke, 1984). Also, starting with initial human expert-based rules, an inductive learning program can improve these rules through feedback representing an evaluation of expert system's decisions.

Another use for inductive learning is to generate meaningful classifications of given sets of data, or to organize the sets of data (e.g., collections of rules) into a structure of conceptually simple components (Michalski and Stepp, 1983). We will illustrate this application by an example in section 5.

Most of the above applications have already been tried successfully on some relatively simple problems (e.g., Michalski, 1980; Quinlan, 1983). Current research tries to extend current machine learning techniques in a number of directions, such as: empolying richer knowledge representation languages (e.g., Michalski, 1983), exploring constraints of a domain to control generalization (e.g., Mitchell 1986, Mooney and Bennett, 1986), constructing causal explanations and models (e.g., Doyle, 1986), automating the process of generating new attributes and operators by utilizing the domain knowledge (i.e., the *constructive induction*, or the *new term* problem; e.g., Michalski, 1983), coping with the uncertainty and noise in the data (e.g., Quinlan, 1986; Michalski et al, 1986), integrating different learning strategies (e.g., Lebowitz, 1986), constructing conceptual classifications of structured objects (e.g., Stepp and Michalski, 1986; Norhausen, 1986), infering components of structures (e.g., Rose and Langley, 1986).

As mentioned above, we can distinguish between two types of inductive learning: learning from examples and learning by observation and discovery. Let us discuss these two strategies of learning in greater detail.

## 4. Learning from Examples

Within the category of learning from examples we can distinguish two major types: *instance-to-class* generalization and *part-to-whole* generalization. In the instance-to-class generalization, given are independent instances (examples) of some class of objects, and the task is to induce a general description of the class. The instances can be representations of physical objects, sounds, images, actions, processes, abstract concepts, etc. Most research on learning from examples is concerned with this type of problem. For example, such research includes learning descriptions of block structures (e.g., Winston, 1977) or automatically inducing diagnostic rules for soybean diseases (Michalski and Chilausky, 1980). For a review of methods of such generalization see (Dietterich and Michalski, 1983).

In part-to-whole generalization, given are only selected parts of an object (a scene, a situation, a procress) and the task is to hypothesize a description of the whole object. A simple example of this type of problem is to determine a rule characterizing a sequence of objects (or a process) from seeing only a part of the sequence (or process). A specific case of such a problem occurs in the card game Eleusis, where players are supposed to discover a "secret" rule governing a sequence of cards. A computer program capable of discovering such rules has been described by Dietterich and Michalski (1983). A more advanced version of the program has been described by Michalski, Ko and Chen (1985).

The problem of discovering Eleusis rules is an instance of a more general problem of *qualitative prediction,* that is concerned with predicting behavior of any discrete processes in a qualitative way (Michalski, Ko and Chen, 1987).

In instance-do-class generalization, facts can be viewed as implications of the form

$$Event ::> Class$$

where *event,* is a description of some object or situation, and *class* represents a decision class or concept to be assigned to this object or situation. (We denote the implication between a fact or pattern, and the class associated with it by the symbol "::>", in order to distinguish it from the general implication symbol "⇒".) The result of learning is a rule:

$$Pattern ::> Class$$

where *Pattern* is an expression in some formal language describing events that belong to the given *Class,* and no events that do not belong to this class. When an unknown event satisfies the *Pattern* then it is assigned to *Class.*

The pattern description can be expressed in a many forms, e.g., a propositional or predicate logic expression, a decision tree, a formal grammar, a sematic network, a frame, a script, a computer program. The complexity of the process of inducing a pattern description from examples depends on two factors: 1) the complexity of the description language used (e.g., the number and the type of operators the system understands), and 2) the intricacy of the pattern description itself. If the pattern description involves no intermediate concepts then the above rule describes one-level class descriptions. In multi-level class descriptions there are intermediate rules between the lowest level concepts involving only measurable properties of objects, and a top level description involving higher level concepts directly related to the given class or concept.

Another important classification of learning techniques is based on the degree to which *descriptors* (attributes, relations, predicates, operators) used in the observational statements are relevant to the decision classes. The degree measures the relationship between initially given descriptors and the descriptors used in the final class description. At the lowest level, the descriptors used in the observational statements are the same as the ones used in the class descriptions. That means that the given descriptors are directly relevant to class descriptions. Such a case is assumed in many methods. At the next level, the initial descriptors contain the relevant ones, but not all of them are relevant. In this case, the system must have the ability to determine the relevant descriptors among many given descriptors. At the highest level, initial descriptors may be completely different from the ones used in the final concept description. We illustrate this case in Sect. 5, where given descriptors are simple physical properties of some objects (in this case trains), and the final descriptors are not directly observable, abstract concepts (such as "trains with toxic or non-toxic loads").

Let us illustrate *learning from examples (concept acquisition)*, and differentiate it from *learning from observation*, by an example problem known as "East-bound and West-bound trains" [Michalski & Larson, 1977] shown in Fig. 1. In the original

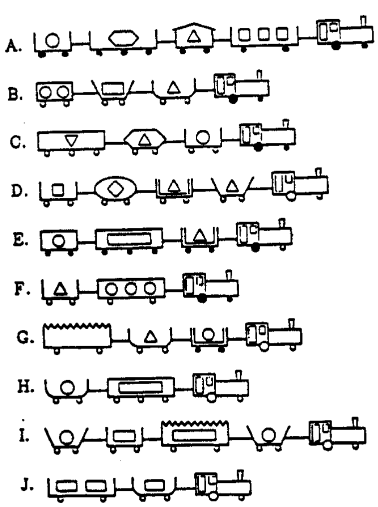


**Fig. 1.** The Unclassified TRAINS Problem

problem, given are two collections of trains, those that are "East-bound" (A to E) and those that are "West-bound" (F to J). The task is to determine a simple descriptive rule distinguishing between the East-bound and West-bound trains using examples of the trains.

These trains are highly structured objects. Each train consists of a sequence of cars of different shapes and sizes. The trains have different number of cars, and cars have different lengths. Thus, an adequate description of trains involves both qualitative and quantitative descriptors, e.g., numerical attributes such *as number of trains*, the *length of a car*, or the *number of loads in a car*, categorical attributes such as *shape of a car*, and relations such as *contains*, *in-front-of*. To illustrate one possible solution, let us present the discriminant descriptions of East-bound and West-bound trains found by the program INDUCE/2 [Hoff, Michalski, Stepp, 1983]. These discriminant descriptions, i.e., rules for distinguishing between the two classes of trains are expressed in the Annotated Predicate Calculus (APC). APC is a typed predicate calculus with additional operators (Michalski, 1983). The descriptions are:

East-bound (train) < :: ∃ (car) [contains(train, car)] & [length(car) = short] [shape(car) = closed]

*("A train is East-bound if it contains a short, closed car.")*

West-bound (train) < :: [num-cars(train) = 2] ∨ ∃ (car) [contains(train, car)] [shape(car) = jagged top]

*("A train is West-bound if there are two cars in the train or if there is a car with a jagged top.")*

These solutions are not easy to find without an aid of a computer program, but once found, they seem to be obvious.

An early practical application of the learning from examples strategy to building the knowledge base of an expert system is described in the paper by Michalski and Chilausky (1976). In the follow-up paper (Michalski and Chilausky, 1980), the learning from examples strategy was compared with the strategy of learning by being told in the context of building the afore-mentioned expert system PLANT/ds. This experiment resulted in inductively derived diagnostic rules (i.e., those obtained by machine learning from examples) that outperformed the rules determined by interviewing an expert (i.e., those acquired by the learning from instruction strategy). Reinke (1984) described a system for testing the consistency and completeness of a rule base using techniques of inductive inference. A recent example of an application of learning from examples to diagnostic problems in medicine is described in [Michalski et al, 1986].

## 5. Learning from Observation

The learning form observation strategy is applied when a collection of facts (observations) is given and one wants to develop a general description (a theory) explaining the facts. It is assumed that there is no teacher who can explain the facts or identify important or relevant concepts applicable to them.

The first step in developing a theory about a collection of facts is usually the creation of a classification (taxonomy). Such a classification can be considered a general description of these facts. Creating simple yet useful classifications is a challenging intellectual process of great importance.

So far, the problem of automatically creating classifications has been studied mainly in the areas of numerical taxonomy and cluster analysis. In these areas, the basic principle for creating a classification is to form classes of objects using some mathematical measure of similarity between the objects. This measure is defined over a finite, an a priori defined set of attributes characterizing the objects. Objects are put to the same class if they have a high degree of similarity, and to different classes if they have a low degree of similarity.

One difficulty with this approach is that classes (concepts) formed solely on the basis of a predefined measure of similarity can be difficult to interpret conceptually. In fact, the interpretation of obtained classifications is assumed in this approach to be the task of a data analyst. This approach does not take into consideration possible varying goals for classification, nor does it use general concepts or linguistic constructs that characterize a collection of observations as a whole (i.e., concepts that capture Gestalt properties). For example, if a collection of points forms a "T-joint", then in order to describe it this way, the system must contain in its background knowledge a method for recognizing such a concept. Without it, even if the computation of similarities (here, reciprocal of distances) puts all the points forming a "T-joint" into the same class, the system still would not "know" that the collection can be described this way.

An alternative approach to creating classifications is based on *conceptual clustering* (Michalski, 1980; Michalski and Stepp, 1983). In this approach, observations are partitioned into classes that represent some conceptual entities. Instead of *similarity*, the approach uses the measure of *conceptual cohesiveness* between objects. While the similarity of objects A and B is a function only of properties of these objects, i.e., is a two-argument function f(A, B), the conceptual cohesiveness is a function of the properties of objects A and B, of the surrounding objects, E (the environment), and of a set of concepts, C, available in the given description language for describing these two objects together. Thus, the conceptual cohesiveness is a four-argument function f(A, B, E, C).

In *conjunctive conceptual clustering* objects are assembled into classes that represent conjunctive concepts closely circumscribing or "fitting" the objects in the class, and satisfying some additional criteria measuring *clustering quality* (Michalski and Stepp, 1983). These criteria take into consideration the relation of the classes to a set of possible goals of classification, the complexity of generated class descriptions, their "disjointness", and other factors (Michalski and Stepp, 1983). The conjunctive concepts are descriptions in the form of conjunctions of statements specifying properties (attribute values) of objects representing the given concept, the relations among the object parts and the properties of the parts.

For illustration, let us consider an example (borrowed from Stepp and Michalski, 1986). Suppose that trains in Fig. 1 are not assigned to any classes, and the task is to create a meaningful classification(s) of these trains. What criteria would people use to create such a classification?

To answer this question, experiments were performed with 31 subjects, who where asked to solve this problem (Medin, Wattenmaker and Michalski, 1986). The subjects devised a total of 93 classifications of the trains. The most popular criterion for classification (used in 17 classifications) was simply the number of cars in the train. Thus, trains were classified into 3 groups: 2-car, 3-car and 4-car trains respectively. The second most popular classification (7 cases) was based on the color of engine wheels. Trains were classified to two groups: a group in which all engine wheels are white, and the group in which engine wheels have varied colors.

These results suggest that even in the absence of clear goals for a classification, people have tendency to use similar criteria for creating a classification. This similarity pattern was not very strong in the experiment, however, as indicated by a large number (40 out of 93) different classifications proposed. The same "Unclassified Trains" problem was given to the recently developed program CLUSTER/S (Stepp, 1984). The program generated several classifications. Two of them are shown in Fig. 2. The first classification A, uses as classification criterion the *num-*

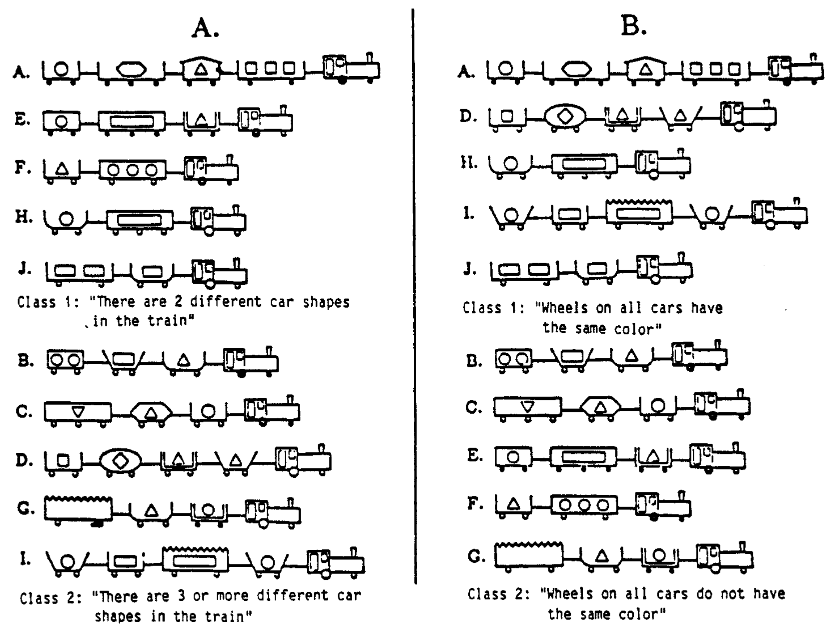


**Fig. 2.** Two classifications of TRAINS created by program CLUSTER/S

*ber of different car shapes* in the train. In the second classification B, the criterion used is whether the wheels on all cars in a train are the same or not. Although these classifications are different from the most frequent classifications made by people, they seem to be reasonable, and even appealing.

It should be mentioned that the initial descriptions of the trains (the observational statements) did not include statements about the number of different car shapes or whether the car wheels have the same color. How did the program generate such statements and use them in creating classifications?

The background knowledge of the program contained inference rules that, when applied to the original descriptions of the trains, can generate new possibly relevant descriptions (attributes and relations) characterizing given objects (here, trains) or their parts. Using various heuristics, the program selectively generates new descriptors and attempts to apply them in the process of determining candidate classifications. The program evaluates these classification according to the *classification quality* criterion LEF (lexicographic evaluation functional). The LEF criterion takes into consideration various properties of a classification, such as the degree to which it satisfies a set of goals (defined in the program's background knowledge), the degree of *fit* between a classification and the observed events (objects), and the importance of descriptors occurring in the class descriptions (see, Michalski, 1980; Medin, 1982; Michalski and Stepp, 1983 a,b; Stepp and Michalski, 1986).

To illustrate the concept of *importance* of a descriptor, let us assume that background knowledge of the system includes a rule which defines cars in the train that carry toxic chemicals. Suppose that such a rule is:

$$\begin{gathered}[\text{contains(train, car)}]\ \&\ [\text{car-shape(car)} = \text{opentop}]\ \&\ [\text{cargo-shape(car)} = \text{circle}] \\ [\text{items-carried(car)} = 1] \Leftrightarrow [\text{toxic-chemicals(train)}]\end{gathered}$$

In this rule, the equivalence operator is used to state that the negation of the condition part is sufficient to assert the negative of the consequence. After applying this rule to each train description, the right-hand side of the rule will be appended to the description (as an additional predicate) to indicate the presence or absence of toxic-chemicals on the given train. This predicate will in turn trigger other inference rules that are part of the program's background knowledge:

- toxic chemicals are dangerous,
- dangerous things are important,
- important things should have high selection value (high preference score).

As a result of this inference the program will propose a candidate classification of trains into those containing toxic chemicals and those not containing such chemicals.

The descriptor generation process outlined above constructs new attributes from combinations of existing attributes. This process is guided by various heuristics. For example, two or more numerical attributes can be combined into a single attribute using arithmetic operators. To suggest appropriate arithmetic operators, a trend analysis can be used, as in BACON 4 (Langley, Bradshaw, Simon, 1983).

Predicates or whole rules can be combined by logical operators to form new attributes. For example, the rule

[cold-blooded(a 1)] & [offspring birth(a 1) = egg] ⇒ [animal-type(a 1) = reptile]

yields a new attribute "animal-type" with a specified value "reptile". Using this rule and similar ones, one might classify some animals into groups of reptiles, mammals, and birds (even though the type of each animal is not stated in the original data about animals).

Such classification construction problems occur when one wants to organize and classify observations that require structural descriptions. Problems of this type include classifying physical or chemical structures, analyzing genetic sequences, building taxonomies of plants or animals, structuring visual scenes, and splitting a sequence of temporal events into episodes with simple meanings. In an expert system, a classification construction program could be used, for example, to structure a large knowledge base of decision rules, or to structure the database of facts about a given problem.

## 6. Summary

Fundamental learning strategies have been discussed including *direct implantation of knowledge, learning by instruction, learning by deductive inference, learning by analogy, learning from examples* and, finally, *learning by observation and discovery.* The order of these strategies reflects the increasing complexity of the inference performed on the information given to a learning system in order to derive the desired knowledge.

Learning from examples and learning from observation are two basic forms of inductive learning. The paper discussed and illustrated the importance of using background knowledge in applying these learning strategies. The capability to incorporate background knowledge in inductive learning is an important prerequisite for the successful application of this form of learning.

**Acknowledgement**

It is this author's pleasant duty to thank Gail Thornburg and Robert Stepp for comments on the earlier version the paper. He also gratefully acknowledges the partial support of the research presented here by the National Science Foundation under grant DCR 84-06801, by the Office of Naval Research under grant N00014-82-0186, and by the Defense Advanced Research Projects Agency under the Office of Naval Research contract No. N00014-K-85-0878. A part of this paper was written while the author was with the MIT Artificial Intelligence Laboratory. The support for the Laboratory's research is provided in part by the Defence Adavnced Research Projects Agency under the Office of Naval Research contract N00014-80-C-0505.

## References

Boulanger, A: The Expert System PLANT/cd: A Case Study in Applying the General Purpose Inference System ADVISE to Predicting Black Cutworm Damage in Corn, M.S. Thesis, UIUCDS-R-83-1134, Department of Computer Science, University of Illinois, Urbana, IL, July 1983.

Buchanan BG, Mitchell TM, Smith RG, Johnson CR Jr, Models of Learning Systems, Tech Rept STAN-CS-79-692, Computer Science Department, Stanford University, 1979.

Burstein MH: Concept Formation by Incremental Analogical Reasoning and Debugging, Proceedings of the 2nd International Machine Learning Workshop, Allerton House, University of Illinois at Urbana-Champaign, June 22-24, 1983.

Carbonell JG, Learning by Analogy: Formulating and Generalizing Plans From Past Experience, chapter in Machine Learning: An Artificial Intelligence Approach, Michalski RS, Carbonell J, Mitchell T (eds.), Tioga Publishing Company, 1983.

Carbonell JG, Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition, chapter in Machine Learning, Volume II, Morgan Kaufman Publishers, Inc., 1986.

Chilausky R, Jacobsen B, Michalski RS: An Application of Variable-Valued Logic to Inductive Learning of Plant Disease Diagnostic Rules, Proceedings of the 1976 International Symposium on Multiple-Valued Logic, Utah State University, Logan, Utah, May 25-28, 1976.

Davis R, Knowledge Acquisition in Rule Based Systems: Knowledge About Representation as a Basis For System Construction and Maintenance. Chapter in Pattern-Directed Inference Systems, Waterman DA, Hayes-Roth F (eds.), Academic Press, 1978.

DeJong G, Mooney R, Explantation-Based Learning: An Alternative Review, Machine Learning Journal, Vol I, No 2, pp 145-176, Kluwer Academic Publishers, 1986.

Dietterich TG, Michalski RS, A Comparative Review of Selected Methods for Learning From Examples, chapter in Machine Learning: An Artificial Intelligence Approach, Michalski RS, Carbonell J, Mitchell T, (eds.), Tioga Publishing Company, 1983.

Doyle RJ, Constructing and Refining Causal Explanations from an Inconsistent Domain Theory, Proceedings of AAAI-86, Fifth National Conference on Artificial Intelligence, Vol I, pp 538-544, Philadelphia, PA, August 1986.

Hass N, Hendrix GG, Learning by Being Told: Acquiring Knowledge for Information Management, chapter in Machine Learning: An Artificial Intelligence Approach, Michalski RS, Carbonell J, Mitchell T, (eds.), Tioga Publishing Company, 1983.

Hoff WA, Michalski RS, Stepp RE, INDUCE 2: A Program For Learning Structural Descriptions From Examples, Technical Report, Department of Computer Science, University of Illinois, Urbana, Illinois, September, 1983.

Hu Chi-Heng, Pao Yoh-Han, Processing of Pattern Based Information, Part II: Description of Inductive Inference in Terms of Transition Networks, Technical Report 102-82 Part II, Case Institute of Technology, Case Western Reserve University, Cleveland, Ohio 44106, 1982.

Langley P, Bradshaw GL, Simon HA, Rediscovering Chemistry With the BACON System, chapter in Machine Learning: An Artificial Intelligence Approach, Michalski RS, Carbonell J, Mitchell T, (eds.), Tioga Publishing Company, 1983.

Lebowitz M, Not the Path to Perdition: The Utility of Similarity-Based Learning, Proceedings of AAAI-86, Fifth National Conference on Artificial Intelligence, Vol I, pp 533-537, Philadelphia, PA, August, 1986.

Lenat D, AM An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search, Computer Science Department, Rept STAN-CS-76-750, Stanford University, Stanford, CA, 1976.

Lenat D, Hayes-Roth F, Klahr PH, Cognitive Economy in a Fluid Task Environment, Proceedings of the International Machine Learning Workshop, Michalski RS (ed.), Allerton House, June 22-24, 1983.

Medin DL, Structural Principles in Categorization, Developments of Perception and Cognition, Shepp B, Tighe T (eds.), Erlbaum, Hillsdale, NJ, pp 203-230, 1983.

Medin DL, Wattenmaker WS, Michalski RS, Constraints and Preferences in Inductive Learning: An Experimental Study Comparing Human and Machine Performance, ISG Report 86-1, Computer Science Department, University of Illinois, Urbana, January, 1986.

Michalski RS, A Variable-Valued Logic System as Applied to Picture Description and Recognition, Chapter in the book Graphic Languages, Nake F, Rosenfeld A (eds.), North-Holland Publishing Co., 1972.

Michalski RS, Knowledge Aquisition Through Conceptual Clustering: A Theoretical Framework and an Algorithm for Partitioning Data Into Conjunctive Concepts, Policy Analysis and Information Systems, Vol 4, No 3, pp 219-244, 1980b.

Michalski RS, Chilausky RL, Knowledge Acquisition by Encoding Expert Rules Versus Computer Induction From Examples: A Case Study Involving Soybean Pathology, International Journal for Man-Machine Studies, No 12, pp 63-87, 1980.

Michalski RS, A Theory and Methodology of Inductive Learning, chapter in Machine Learning: An Artificial Intelligence Approach, Michalski RS, Carbonell JG, Mitchell TM (eds.), Tioga Publishing Company, 1983.

Michalski RS, Ko H, Chen K, SPARC/E - An Eleusis Rule Generator and Game Player, UIUCDCS-F-85-941, ISG Report 85-11, Department of Computer Science, University of Illinois, Urbana, February 1985.

Michalski RS, Stepp RE, Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol PAMI-5, No 4, pp 396-410, July, 1983a.

Michalski RS, Stepp RE, Learning From Observation: Conceptual Clustering, chapter in Machine Learning, Michalski RS, Carbonell J, Mitchell T (eds.), Tioga Publishing Company, 1983b.

Michalski RS, Mozetic I, Hong J, Lavrac N, The Multi-Purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains, Proceedings of AAAI-86, Fifth National Conference on Artificial Intellingence, Vol 2, pp 1041-1045, Philadelphia, PA, August 1986.

Michalski RS, Ko H, Chen K, Qualitative Prediction: A Method and Program SPARC/G, Chapter in Expert Systems, Guetler C (ed.), Academic Press, 1987.

Mitchell TM, Version Spaces: An Approach to Concept Learning, Thesis PhD, Stanford University, Stanford, CA 1978.

Mitchell TM, Utgoff PE, Banerji R, Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics, chapter in Machine Learning: An Artificial Intelligence Approach, Michalski RS, Carbonell JG, Mitchell TM (eds.), Tioga Publishing Company, 1983.

Mitchell TM, Keller RM, Kedar-Cabelli ST, Explanation-based Generalization: A Unifying View, Machine Learning, pp 47-80, Vol 1, No 1, 1986.

Mooney RJ, Bennett SW, A Domain Independent Explanation-Based Generalizer, Proceedings of AAAI-86, Fifth National Conference on Artificial Intelligence, Vol I, pp 551-555, Philadelphia, PA, August 1986.

Nordhausen B, Conceptual Clustering Using Relational Information, Proceedings of AAAI-86, Fifth National Conference on Artificial Intelligence, Vol I, pp 508-512, Philadelphia, PA, August 1986.

Pao Yoh-Han, Hu Chi-Heng, Processing of Pattern Based Information, Part I: Inductive Inference Methods Suitable For Use in Pattern Recognition and Artificial Intelligence, Technical Report 101-82, Case Institute of Technology, Case Western Reserve University, Cleveland, Ohio 44106, 1982.

Quinlan JR, Learning Efficient Classification Procedures and Their Application to Chess End Games, Chapter in Machine Learning: An Artificial Intelligence Approach, Michalski RS, Carbonell JG, Mitchel TM (eds.), Tioga Publishing Co., pp 463-482, 1983.

Quinlan JR, The Effect of Noise on Concept Learning, Chapter in Machine Learning: An Artificial Intelligence Approach, Vol II, Michalski RS, Carbonell JG, Mitchell TM (eds.), Morgan Kaufmann Publishers, pp 149-166, 1986.

Reinke R, Knowledge Acquisition and Refinement Tools for the ADVISE Meta-Expert System, Thesis MS, Department of Computer Science, University of Illinois, Urbana, Illinois 1984.

Rendell LA, A New Basis For State-Space Learning Systems and a Successful Implementation, Artificial Intelligence, Vol 20, No 4, pp 369-392, 1983

Rose D, Langley P, STAHLp: Belief Revision in Scietific Discovery, AAAI-86, Fifth National Conference on Artificial Intelligence, Vol I, pp 528-532, Philadelphia, PA, August 1986.

Stepp RE, Conjunctive Conceptual Clustering: A Methodology and Experimentation, PhD Thesis, Department of Computer Science, University of Illinois, Urbana, 1984.

Stepp RE, Michalski RS, Conceptual Clustering: Inventing Goal-Oriented Classification of Structured Objects, Machine Learning: An Artificial Intelligence Approach Vol II, Michalski RS, Mitchell TM, Carbonell JG (eds.), Morgan Kaufmann, 1986.

Waltz D, Genesereth M, Hart P, Hendrix G, Joshi A, McDermott J, Mitchel T, Nilsson N, Wilensky R, Artificial Intelligence: An Assessment of the State-of-the-Art and Recommendation for the Future Directions, AI Magazine, Vol 4, No 3, Fall 1983.

Winston PH, Learning Structural Descriptions From Examples, Tech. Report AI TR-213, MIT, AI Lab, Cambridge, MA, 1977.

Winston PH, Learning by Augmenting Rules and Accumulating Sensors, Chapter in Machine Learning: An Artificial Intelligence Approach Vol II, Michalski RS, Mitchell TM, Carbonell JG (eds.), Morgan Kaufmann Publishers, Inc. 1986.