LEARNING = INFERENCING + MEMORIZING: INTRODUCTION TO INFERENTIAL THEORY OF LEARNING

by

R. S. Michalski

In Foundations of Knowledge Acquisition, Vol. 2: Machine Learning, S. Chipman and A. Meyrowitz (Eds.), 1993.

Original 3/11/93

FOUNDATIONS OF KNOWLEDGE ACQUISITION: Machine Learning

edited
by

Alan L. Meyrowitz
Naval Research Laboratory

Susan Chipman
Office of Naval Research

KLUWER ACADEMIC PUBLISHERS
Boston/Dordrecht/London

TABLE OF CONTENTS

Chapter 1

LEARNING = INFERENCING + MEMORIZING

Basic Concepts of Inferential Theory of Learning and Their Use for Classifying Learning Processes

Ryszard S. Michalski
Center for Artificial Intelligence
George Mason University
Fairfax, VA 22030

ABSTRACT

This chapter presents a general conceptual framework for describing and classifying learning processes. The framework is based on the *Inferential Theory of Learning* that views learning as a search through a *knowledge space* aimed at deriving knowledge that satisfies a learning goal. Such a process involves performing various forms of inference, and memorizing results for future use. The inference may be of any type—deductive, inductive or analogical. It can be performed explicitly, as in many symbolic systems, or implicitly, as in artificial neural nets. Two fundamental types of learning are distinguished: *analytical* learning that reformulates a given knowledge to the desirable form (e.g., skill acquisition), and *synthetic* learning that creates new knowledge (e.g., concept learning). Both types can be characterized in terms of *knowledge transmutations* that are involved in transforming given knowledge (input plus background knowledge) into the desirable knowledge. Several transmutations are discussed in a novel way, such as deductive and inductive generalization, abductive derivation, deductive and inductive specialization, abstraction and concretion. The presented concepts are used to develop a general classification of learning processes.

Key words: learning theory, machine learning, inferential theory of learning, deduction, induction, abduction, generalization, abstraction, knowledge transmutation, classification of learning.

INTRODUCTION

In the last several years we have been witnessing a great proliferation of methods and approaches to machine learning. Research in this field now spans such subareas or topics as empirical concept learning from examples, explanation-based learning, neural net learning, computational learning theory, genetic algorithm based learning, cognitive models of learning, discovery systems, reinforcement learning, constructive induction, conceptual clustering, multistrategy learning, and machine learning applications. In view of such a diversification of machine learning research, there is a strong need for developing a unifying conceptual framework for characterizing existing learning methods and approaches.

Initial results toward such a framework have been presented in the form of *Inferential Theory of Learning* (ITL) by Michalski (1990a, 1993). The purpose of this chapter is to discuss and elaborate selected concepts of ITL, and use them to describe a general classification of learning processes.

The ITL postulates that learning processes can be characterized in terms of operators (called "knowledge transmutations"—see next section) that transform input information and initial learner's knowledge to the knowledge specified by the goal of learning. The main goals of the theory are to analyze and explain diverse learning methods and paradigms in terms of knowledge transmutations, regardless of the implementation-dependent operations performed by different learning systems. The theory aims at understanding the *competence* of learning processes, i.e., their logical capabilities. Specifically, it tries to explain what type of knowledge a system is able to derive from what type of input and learner's prior knowledge, what types of inference and knowledge transformations underlie different learning strategies and paradigms, what are the properties and interrelationships among knowledge transmutations, how different knowledge transmutations are implemented in different learning systems, etc. The latter issue is particularly important for developing systems that combine diverse learning strategies and methods, because different knowledge representations and computational mechanisms facilitate different knowledge transmutations.

Knowledge transmutations can be applied in a great variety of ways to a given input and background knowledge. Therefore, the theory emphasizes the importance of *learning goals*, which are necessary for guiding learning processes. Learning goals reflect the knowledge needs of the learner, and often represent a composite structure of many subgoals, some of which are consistent and some may be contradictory. As to the research methodology employed, the theory attempts to explain learning processes at the level of abstraction that allows it to be relevant both to cognitive models of learning, and those studied in machine learning.

The above research issues make the Inferential Theory of Learning different from and complementary to Computational Learning Theory (e.g., Warmuth and Valiant, 1991), which is primarily concerned with the *computational complexity* or *convergence* of learning algorithms. The presented work draws upon the ideas described in (Michalski, 1983 & 1990a; Michalski and Kodratoff, 1990b; and Michalski, 1993).

LEARNING THROUGH INFERENCE

Any act of learning aims at improving learner's knowledge or skill by interacting with some information source, such as an environment or a teacher. The underlying tenet of the Inferential Theory of Learning is that any learning can be usefully viewed as a process of creating or modifying knowledge structures to satisfy a learning goal. Such a process may involve performing any type of inference—deductive, inductive or analogical.

Figure 1 illustrates an information flow in a general learning process according to the theory. In each learning cycle, the learner generates new knowledge and/or a new form of knowledge by performing inferences from the input information and the learner's prior knowledge.

When obtained knowledge satisfies the learning goal, the knowledge is assimilated into the learner's knowledge base. The input information to a learning process can be observations, stated facts, concept instances, previously formed generalizations or abstractions, conceptual hierarchies, information about the validity of various pieces of knowledge, etc.

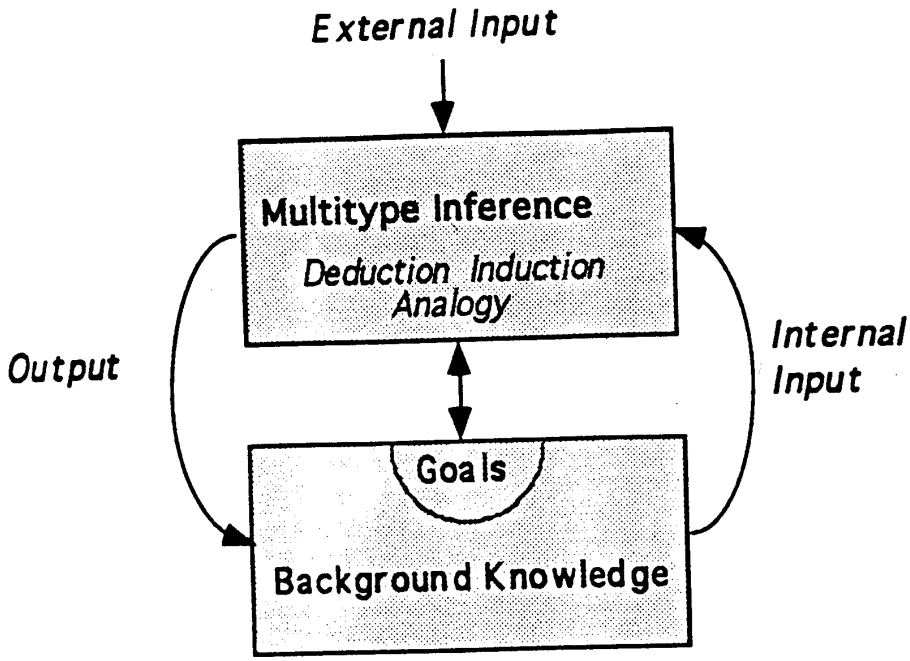


Figure 1. A schematic characterization of learning processes.

Any learning process needs to be guided by some underlying goal, otherwise the proliferation of choices of what to learn would quickly overwhelm any realistic system. A learning goal can be general (domain-independent), or domain-dependent. A general learning goal defines the type of knowledge that is desired by a learner. There can be many such goals, for example, to determine a concept description from examples, to classify observed facts, to concisely describe a sequence of events, to discover a quantitative law characterizing physical objects, to reformulate given knowledge into a more efficient representation, to learn a control algorithm to accomplish a task, to confirm a given piece of knowledge, etc. A domain-specific goal defines a specific knowledge needed by the learner.

At the beginning of a learning process, the learner determines what prior knowledge is relevant to the input and the learning goal. Such goal-relevant part of learner's prior knowledge is called *background knowledge* (*BK*). The BK can be in different forms, such as declarative (e.g., a collection of statements representing conceptual knowledge), procedural (e.g., a sequence of instructions for performing some skill), or a combination of both. Input and output knowledge in a learning process can also be in such forms. One way of classifying learning processes is based on the form of input and output knowledge involved in them (Michalski, 1990a).

The Inferential Theory of Learning (ITL) states that learning involves performing inference ("inferencing") from the information supplied and the learner's background knowledge, and memorizing its results that are found to be useful. Thus, one can write an "equation":

$$\textit{Learning} = \textit{Inferencing} + \textit{Memorizing} \qquad (1)$$

It should be noted that the term "inferencing" is used in (1) in a very general sense, meaning any type of knowledge transformation or manipulation, including syntactic transformations and random searching for a specified knowledge entity. Thus, to be able to learn, a system has to be able to perform inference, and to have a memory that supplies the background knowledge, and stores the results of inferencing.

As mentioned earlier, ITL postulates that any learning process can be described in terms of generic units of knowledge change, called *knowledge transmutations* (or *transforms*). The transmutations derive one type of knowledge from another, hypothesize new knowledge, confirm or disconfirm knowledge, organize knowledge into structures, determine properties of given knowledge, insert or delete knowledge, transmit knowledge from one physical medium to another, etc. Transmutations may performed by a learner explicitly, by well-defined rules of inference (as in many symbolic learning systems), or implicitly, by specific mechanisms involved in information processing (as in neural-net learning or genetic algorithm based learning). The capabilities of a learning system depend on the types and the complexity of transmutations a learning system is capable of performing.

Transmutations are divided to two classes: *knowledge generation* transmutations and *knowledge manipulation* transmutations. Knowledge generation transmutations change the content of knowledge by performing various kinds of inference. They include, for example, generalization, specialization, abstraction, concretion, similization, dissimilization, and any kind of logical or mathematical derivation (Michalski, 1993). Knowledge manipulation transmutations perform operations on knowledge that do not change its content, but its organization, physical distribution, etc. For example, inserting a learned component to a given structure, replicating a given knowledge segment in another knowledge base, or sorting given rules in a certain order are knowledge manipulation transmutations.

This chapter discusses two important classes of knowledge generation transmutations {generalization, specialization}, and {abstraction, concretion}. These classes are particularly relevant to the classification of learning processes discussed in the last section.

Because Inferential Theory views learning as an inference process, it may appear that it only applies to symbolic methods of learning, and does not apply to "subsymbolic" methods, such as neural net learning, reinforcement learning or genetic algorithm-based learning. It is argued that it also applies to them, because from the viewpoint of the input-output transformations, subsymbolic methods can also be characterized as performing knowledge transmutations and inference. Clearly, they can generalize inputs, determine similarity between inputs, abstract from details, etc.

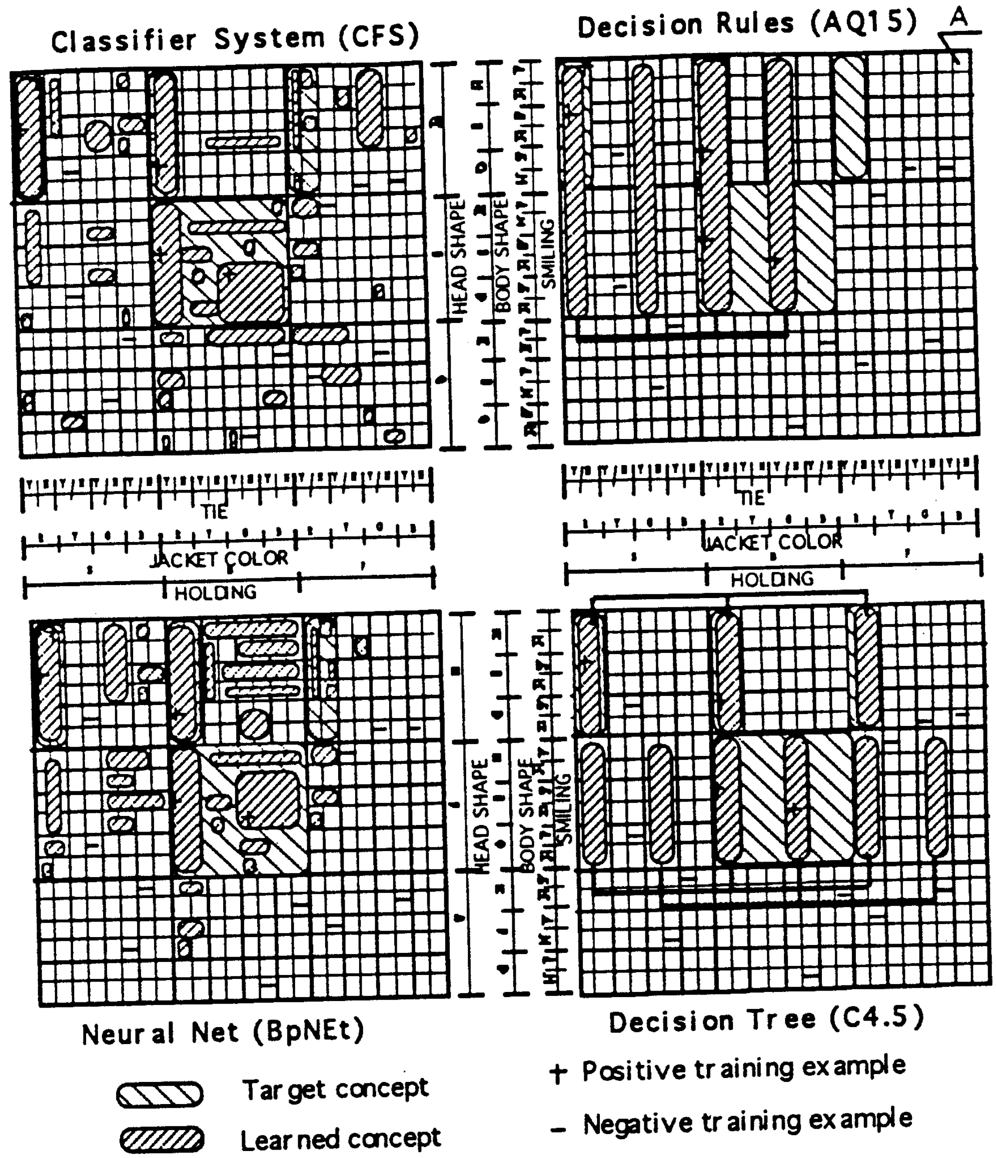
From the ITL viewpoint, symbolic and subsymbolic systems differ in the type of computational and representational mechanisms they use for performing transmutations. Whether a learning system works in parallel or sequentially, weighs inputs or performs logic-based transformations affects the system's speed, but not its ultimate competence (within limits), because a parallel algorithm can be transformed into a logically equivalent sequential one, and a discrete neural net unit function can be transformed into an equivalent logic-type transformation. These systems differ in the efficiency and speed of peforming different transmutations. This makes them more or less suitable for different learning tasks.

In many symbolic learning systems, knowledge transmutations are performed in an explicit way, and in conceptually comprehensible steps. In some inductive learning systems, for example, INDUCE, generalization transmutations are performed according to well-defined rules of inductive generalization (Michalski, 1983).

In subsymbolic systems (e.g., neural networks), transmutations are performed implicitly, in steps dictated by the underlying computational mechanism (see, e.g., Rumelhart et al., 1986). A neural network may generalize an input example by performing a sequence of small modifications of the weights of internode connections. Although these weight modifications do not directly correspond to any explicit inference rules, the end result, nevertheless, can be characterized as a certain knowledge transmutation.

The latter point is illustrated by Wnek et al. (1990), who described a simple method for visualizing generalization operations performed by various symbolic and subsymbolic learning systems. The method, called DIAV, can visualize the target and learned concepts, as well as results of various intermediate steps, no matter what computational mechanism is used to perform them.

To illustrate this point, Figure 2 presents a diagrammatic visualization of concepts learned by four learning systems: a classifier system using genetic algorithm (CFS), a rule learning program (AQ15), a neural net (BpNet), and a decision tree learning system (C4.5). Each diagram presents an "image" of the concept learned by the given system from the same set of examples: 6% of positive examples (5 out of the total 84 positive examples constituting the concept), and 3% of negative examples (11 out of possible 348).



The cell A corresponds to the description:
HEAD-SHAPE = R & BODY-SHAPE= R & SMILING = Yes &
HOLDING = F & JACKET COLOR = B & Tie = N

Figure 2. A visualization of the target concept and concepts learned by four learning methods.

In the diagrams, the shaded area marked "Target Concept" represents all possible instances the concept to be learned. The shaded area marked "Learned concept" represents a generalization of training examples hypothesized by a given learning system. The set-theoretic difference between the "Target concept" and the "Learned concept" represents errors in learning (an "Error image"). Each instance belonging to the "Learned concept" and not to the "Target concept," or to the "Target concept" and not to "Learned concept" will be incorrectly classified by the system.

To understand the diagrams, note that each cell of a diagram represents a single combination of attribute values, e.g., an instance of a concept. A whole diagram represents the complete description space (432 instances). The attributes spanning the description space characterize a collection of imaginary robot-like figures. Figure 3 lists the attributes and their value sets.

ATTRIBUTES	LEGAL VALUES
Head Shape	R- round, S- square, O- octagon
Body Shape	R- round, S-square, O-octagon
Smiling	Y- yes , N- no
Holding	S- sword, B- balloon, F- flag
Jacket Color	R- red, Y- yellow, G- green, B- blue
Tie	Y- yes, N- no

Figure 3. Attributes and their value sets.

To determine a logical description that corresponds to a given cell (or a set of cells), one projects the cell (or a set of cells) onto the ranges of attribute values associated with the scales aside of the diagram, and "reads out" the description. To illustrate this, the bottom part of Figure 2 presents a description of the cell marked in the diagram as A.

By analyzing the images of the concepts learned by different paradigms, one can determine the degree to which they generalized the original examples, can "see" the differences between different generalizations, determine how new or hypothetical examples will be classified according to the learned concepts, etc. For more details on the properties of the diagrams, on the method of "reading out" descriptions from the diagrams, and on the implemented diagrammatic visualization system, DIAV, see (Michalski, 1978, Wnek et al., 1990; Wnek and Michalski, 1992.)

The diagrams allow one to view concepts as images, and thus to abstract from the specific knowledge representation used by a learning method. This demonstrates that from the epistemological viewpoint taken by the ITL, it is irrelevant if knowledge is implemented in the form of a set of rules, a decision tree, a neural net or some other way. For example, in a neural net, the prior knowledge is represented in an implicit way, specifically, by the structure of the net, and by the initial settings of the weights of the connections. The learned knowledge is manifested in the new weights of the connections among the net's units (Touretzky and Hinton, 1988). The prior and learned knowledge incorporated in the net could be re-represented, at least theoretically, in the form of images, or, as explicit symbolic rules or numerical expressions, and then dealt with as any other knowledge. For example, using the diagrams in Figure 2, one can easily "read out" from them a set of rules equivalent to the concepts learned by the neural network and genetic algorithm.

The central aspect of any knowledge transmutation is the type of underlying inference, which characterizes a transmutation along the truth-falsity dimension. The type of inference thus determines the truth status of the derived knowledge. Therefore, before we discuss transmutations and their role in learning, we will first analyze basic types of inference.

BASIC TYPES OF INFERENCE

As stated earlier, ITL postulates that learning involves conducting inference on the input and current BK, and storing the results whenever they are evaluated as useful. Such a process may involve any type of inference, because any possible type of inference may produce knowledge worth remembering. Therefore, from such a viewpoint, a complete learning theory has to include a complete theory of inference.

Such a theory of inference should account for all possible types of inference.

Figure 4 presents an attempt to schematically illustrate all basic types of inference. The first major classification divides inferences into two fundamental types: deductive and inductive. The difference between them can be explained by considering an entailment:

$$P \cup BK \models C \tag{2}$$

where P denotes a set of statements, called ***premise***, BK represents the reasoner's background knowledge, and C denotes a set of statements, called ***consequent***. Deductive inference is deriving consequent C, given premise P and BK. Inductive inference is hypothesizing premise P, given consequent C and BK. Thus, deductive inference can be viewed as "tracing forward" the relationship (2), and inductive inference as "tracing backward" such a relationship. Because of its importance for characterizing inference processes, relationship (2) is called the ***fundamental equation for inference.***

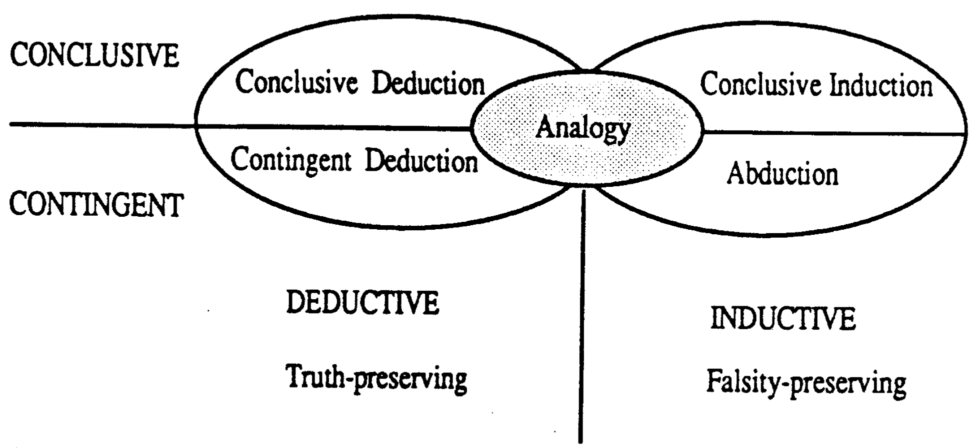


Figure 4. A classification of basic types of inference.

Inductive inference underlies two major knowledge generation transmutations: ***inductive generalization*** and ***abductive derivation.*** They differ in the type of BK they employ, and the type of premise P they hypothesize. Inductive generalization is based on tracing backward a tautological implication, specifically, the rule of *universal specialization.*, i.e., $\forall x, P(x) \Rightarrow P(a)$, and produces a premise P that is a generalization of C, i.e., is a description of a larger set of entities than the set described by C (Michalski, 1990a, 1993). In contrast, abductive derivation is based on tracing backward an implication that represents a domain knowledge, and produces a description that characterizes reasons for C. Other, less known, types of inductive inference are *inductive specialization* and *inductive concretion* (see section on *Inductive Transmutations*).

In a more general view of deduction and induction that also captures their approximate or commonsense forms, the entailment relationship "|=" may also include a "plausible" entailment, i.e., probabilistic or partial. The difference between the "conclusive" (valid) and "plausible" entailment leads to another major classification of inference types. Specifically, inferences can be divided into those based on *conclusive* or *domain-independent* dependencies, and those based on *contingent* or *domain-dependent* dependencies.

A conclusive dependency between statements or sets of statements represents a necessarily true logical relationship, i.e., a relationship that must be true in all possible worlds. Valid rules of inference or universally accepted physical laws represent conclusive dependencies. To illustrate a conclusive dependency, consider the statement "All elements of the set X have the property q." If this statement is true, then the statement "x, an element of X, has the property q" must also be true.

The above relationship between the statements is true independently of the domain of discourse, i.e., of the nature of elements in the set X, and thus is conclusive. If reasoning involves only statements that are assumed to be true, such as observations, "true" implications, etc., and conclusive dependencies (valid rules of inference), then deriving C, given P, is the *conclusive* (or *crisp*) *deduction*, and hypothesizing P, given C, is *conclusive* (or *crisp*) *induction.*

For example, suppose that BK is "All elements of the set X have the property q," and the input (premise P) is "x is an element of X."

Deriving a statement "x has the property q" is conclusive deduction. If BK is "x is an element of X" and the input (the observed consequent C) is "x has the property q," then hypothesizing premise P "All elements of X have the property q" is conclusive induction.

Contingent dependencies are domain-dependent relationships that represent some world knowledge that is not totally certain, but only probable. The contingency of these relationships is usually due to the fact that they represent incomplete or imprecise information about the totality of factors in the world that constitute a dependency. These relationships hold with different "degrees of strength."

To express both conclusive and contingent dependencies within one formalism, the concept of *mutual dependency* is introduced. Suppose S1 and S2 are sentences in PLC (Predicate Logic Calculus) that are either statements (closed PLC sentences; no free variables) or term expressions (open PLC sentences, in which some of the arguments are free variables). If there are free variables, such sentences can be interpreted as representing functions, otherwise they are statements with a truth-status. To state that there is a *mutual dependency* (for short, an *m-dependency*) between sentences S1 and S2, we write

$$S1 \Leftrightarrow S2: \ \alpha, \beta \qquad (3)$$

where α and β, called *merit parameters* , represent an overall *forward strength* and *backward strength* of the dependency, respectively. If S1 and S2 are statements, then an m-dependency becomes an *m-implication.* Such an implication reduces to a standard logical implication if α is 1, and β is undetermined, or α is undetermined and β is 1, otherwise it is a *bi-directional plausible implication..* In such an implication, if S1 (S2) is true, than α (β) represents a measure of certainty that S2 (S1) is true, assuming that no other information relevant to S2 (S1) is known. If S1 and S2 are term expressions, then α and β represent an average certainty with which the value of S1 determines the value S2, and conversely.

An obvious question arises as to the method for representing and computing merit parameters. We do not assume that they need to have a single representation. They could be numerical values representing a degree of belief, an estimate of the probability, ranges of probability, or a qualitative characterization of the strength of conclusions from using the implication in either direction. Here, we assume that they represent numerical degrees of dependency based on the contingency table (e.g., Goodman & Kruskal, 1979; Piatetsky-Shapiro, 1992), or estimated by an expert.

Another important problem is how to combine or propagate merit parameters when reasoning through a network of m-dependencies. Pearl (1988) discusses a number of ideas relevant to this problem. Since the certainty of a statement cannot be determined solely on the basis of the certainties of its constituents, regardless of its meaning, the ultimate solution of this open problem will require methods that take into consideration both the merit parameters and the meaning of the sentences.

A special case of m-dependency is *determination*, introduced by Russell (1989), and used for characterizing a class of analogical inferences. Determination is an m-dependency between term expressions in which α is 1, and β is unspecified, that is, a unidirectional functional m-dependency. If any of the parameters α or β takes value 1, then an m-dependency is called *conclusive*, otherwise is called *contingent*. The idea of an m-dependency stems from research on human plausible reasoning (Collins and Michalski, 1989).

Conclusions derived from inferences involving contingent dependencies (applied in either direction), and/or uncertain facts are thus uncertain. They are characterized by "degrees of belief" (probabilities, degrees of truth, likelihoods, etc.). For example, "If there is fire, there is smoke" is a bi-directional contingent dependency, because there could be a situation or a world in which it is false. It holds in both directions, but not conclusively in either direction. If one sees fire, then one may derive a plausible (deductive) conclusion that there is smoke. This conclusion, however, is not certain. Using reverse reasoning ("tracing backward" the above dependency), observing smoke, one may hypothesize, that there is fire. This is also an uncertain inference, called contingent abduction. It may thus appear that there is no principal difference between contingent deduction and contingent abduction.

These two types of inferences are different if one assumes that there is a causal dependency between fire and smoke, or, generally, between P and C in the context of BK (i.e., P can be viewed as a cause, and C as its consequence). Contingent deduction derives a plausible consequent, C, of the causes represented by P. Abduction derives plausible causes, P, of the consequent C. A problem arises when there is no causal dependency between P and C in the context of BK. In such a situation, the distinction between plausible deduction and abduction can be based on the relative strength of dependency between P and C in both directions (Michalski, 1992). Reasoning in the direction of stronger dependency is plausible deduction, and reasoning in the weaker direction is abduction. If a dependency is completely symmetrical, e.g., $P \Leftrightarrow C$, then the difference between deduction and abduction ceases to exist.

In sum, both contingent deduction and contingent induction are based on contingent, domain-dependent dependencies. Contingent deduction produces likely consequences of given causes, and contingent abduction produces likely causes of given consequences. Contingent deduction is truth-preserving, and contingent induction (or contingent abduction) is falsity-preserving only to the extent to which the contingent dependencies involved in reasoning are true. In contrast, conclusive deductive inference is strictly truth-preserving, and conclusive induction is strictly falsity-preserving (if C is not true, then the hypothesis P cannot be true either). A conclusive deduction thus produces a provably correct (valid) consequent from a given premise. A conclusive induction produces a hypothesis that logically entails the given consequent (though the hypothesis itself may be false).

The intersection of the deduction and induction, i.e., an inference that is both truth-preserving and falsity-preserving, represents an *equivalence-based* inference (or *reformulation*). Analogy can be viewed as an extension of such equivalence-based inference, namely, as a *similarity-based* inference. Every analogical inference can be characterized as a combination of deduction and induction. Induction is involved in hypothesizing an analogical match, i.e., the properties and/or relations that are assumed to be similar between the analogs, whereas deduction uses the analogical match to derive unknown properties of the target analog. Therefore, in Figure 4, analogy occupies the central area.

The above inference types underlie a variety of knowledge transmutations. We now turn to the discussion of various knowledge transmutations in learning processes.

TRANSMUTATIONS AS LEARNING OPERATORS

Inferential Theory of Learning views any learning process as a search through a *knowledge space*, defined as the space of admissible knowledge representations. Such a space represents all possible inputs, all learner's background knowledge, and all knowledge that the learner can potentially generate. In inductive learning, knowledge space is usually called a *description space.*

The theory assumes that search is conducted through an application of knowledge transmutations acting as operators. Such operators take some component of the current knowledge and some input, and generate a new knowledge component. A learning process is defined as follows:

Given	• Input knowledge	(I)
	• Goal	(G)
	• Background knowledge	(BK)
	• Transmutations	(T)

Determine

• Output knowledge O, satisfying goal G, by applying transmutations T to input I and background knowledge BK.

The input knowledge, I, is the information (facts or general knowledge) that the learner receives from the environment. The learner may receive the input all at once or incrementally, Goal, G, specifies criteria that need to be satisfied by the Output, O, in order that learning is accomplished. Background knowledge is a part of learner's prior knowledge that is "relevant" to a given learning process.

Transmutations are generic types of knowledge transformation for which one can make a simple mental model. They can be implemented using many different computational paradigms. They are classified into two general categories: *knowledge generation* transmutations, which change the content or meaning of the knowledge, and *knowledge manipulation* transmutations, which change its physical location or organization, but do not change its content.

Knowledge generation transmutations represent patterns of inference, and can be divided to *synthetic* and *analytic*. Synthetic transmutations are able to hypothesize *intrinsically* new knowledge, and thus are fundamental for knowledge creation (by "intrinsically new knowledge" we mean knowledge that cannot be conclusively deduced from the knowledge already possessed). Synthetic transmutations include inductive transmutations (those that employ some form of inductive inference), and analogical transmutations (those that employ some form of analogy). Analytic (or deductive) transmutations are those employing some form deduction.

This chapter concentrates on a few knowledge generation transmutations that are particularly important for the classification of learning processes described in the last section. A discussion of several other knowledge transmutations is in (Michalski, 1993).

In order to describe these transmutations, we need to introduce concepts of a *well-formed description*, the *reference set* of a description, and a *descriptor*. A set of statements is a well-formed description if and only if one can identify a specific set of entities such that this set of sentences describe. This set of entities (often a singleton) is called the reference set of the description. Well-formed descriptions have truth-status, that is, they can be characterized as true or false, or, generally, by some intermediate truth-value.

For the purpose of this presentation, we will make a simplifying assumption that descriptions can have one of only three truth-values: "true," "false," or "unknown." The "unknown" value is attached to hypotheses generated by contingent deduction, analogy, or inductive inference. The "unkown" value can be turned to true or false by subjecting the hypothesis to a validation procedure. A *descriptor* is an attribute, a function, or a relation, whose value or status is used to characterize the reference set.

Consider, for example, a statement: "Elizabeth is very strong, has Ph.D. in Astrophysics from the University of Warsaw, and likes soccer." This statement is a well-formed description because one can identify a reference set, {Elizabeth}, that this statement describes. This description uses three descriptors: a one-place attribute "degree-of-strength(person)," a binary relation "likes(person, activity)," and a four place relation, "degree-received(person, degree, topic, University). The truth-status of this description is true, if Elizabeth has the properties stated, false it she does not, unknown, if it is not known to be true, but there is no evidence that it is false.

Consider now a sentence: "Robert is a writer, and Barbara is a lawyer." This sentence is not a well-formed description. It could be split, however, to two sentences, each of which would be a well-formed description (one describing Robert, and another describing Barbara). Finally, consider a sentence "George, Jane and Susan like mango, political discussions, and social work." This is a well-formed description of the reference set {George, Jane, Susan}.

Knowledge generation transmutations apply only to well-formed descriptions. Knowledge manipulation transmutations apply to descriptions, as well as entities that are not descriptions (e.g., terms, or sets of terms). Below is a brief description of four major classes of knowledge generation transmutations. First two classes consists of a pair of opposite transmutations, and the third one contains a range of transmutations.

1. Generalization vs. specialization

A generalization transmutation extends the reference set of the input description. Typically, a generalization transmutation is inductive, because the extended set is inductively hypothesized. A generalization transmutation can also be deductive, when the more general assertion is a logical consequence of the more specific one, or is deduced from the background knowledge and/or the input. The opposite transmutation is *specialization* transmutations, which narrows the reference set. A specialization transmutation usually employs deductive inference, but, as shown in the next section, there are also inductive specialization transmutations.

2. Abstraction vs. concretion

Abstraction reduces the amount of detail in a description of a reference set, without changing the reference set. This can be done in a variety of ways. A simple way is by replacing one or more descriptor values by their parents in the generalization hierarchy of values. For example, suppose given is a statement "Susan found an apple." Replacing "apple" by "fruit" would be an abstraction transmutation (assuming that background knowledge contains a generalization hierarchy in which "fruit" is a parent node of "apple"). The underlying inference here is deduction. The opposite transmutation is *concretion*, which generates additional details about a reference set.

3. Similization vs. dissimilization

Similization derives new knowledge about some reference set on the basis of detected partial similarity between this set and some other reference set, of which the reasoner has more knowledge. The similization thus transfers knowledge from one reference set to another reference set, which is similar to the original one in some sense. The opposite transmutation is *dissimilization*, which derives new knowledge from the lack of similarity between the compared reference sets.

The similization and dissimilization are based on analogical inference. They can be viewed as a combination of deductive and inductive inference (Michalski, 1992). They represent patterns of inference described in the theory of plausible reasoning by Collins and Michalski (1989). For example, knowing that England grows roses, and that England and Holland have similar climates, a similization transmutation might hypothesize that Holland may also grow roses. An underlying background knowledge here is that there exists a dependency between climate of a place and the type of plants growing in that location. A dissimilization transmutation would be to hypothesize that bougainvillea, which is widespread on the Caribbean islands, probably does not grow in Scotland, because Scotland and Caribbean islands have very different climate.

4. Reformulation vs. randomization

A reformulation transmutation transforms a description into another description according to equivalence-based rules of transformation (i.e., truth- and falsity-preserving rules). For example, transforming a statement: "This set contains numbers 1,2,3,4 and 5" into "This set contains integers from 1 to 5" is a reformulation. An opposite transmutation is *randomization*, which transforms a description into another description by making random changes. For example, *mutation* in a genetic algorithm represents a randomization transmutation.

Reformulation and randomization are two extremes of a spectrum of intermediate transmutations, called *derivations*. Derivations employ different degrees or types of logical dependence between descriptions to derive one piece of knowledge from another. An intermediate transmutation between the two extremes above is *crossover*, which is also used in genetic algorithms. Such a transmutation derives new knowledge by exchanging parts of two related descriptions.

INDUCTIVE TRASMUTATIONS

Inductive transmutations, i.e., knowledge transformations employing inductive inference have fundamental importance to learning. This is due to their ability to generate intrinsically new knowledge. As discussed earlier, induction is an inference type opposite to deduction. The results of induction can be in the form of generalizations (theories, rules, laws, etc.), causal explanations, specializations, concretions and other. The usual aim of induction is not to produce just any premise ("explanation") that entails a given consequent ("observable"), but the one which is the most "justifiable." Finding such a "most justifiable" hypothesis is important, because induction is an under-constrained inference, and just "reversing" deduction would normally lead to an unlimited number of alternative hypotheses.

Taking into consideration the importance of determining the most justifiable hypothesis, the previously given characterization of inductive inference based on (2) can be further elaborated. Namely, an *admissible induction* is an inference which, given a consequent C, and BK, produces a hypothetical premise P, consistent with BK, such that

$$P \cup BK \models C \qquad (4)$$

and which satisfies the *hypothesis selection criterion.*

In different contexts, the selection criterion (which may be a combination of several elementary criteria) is called a *preference criterion* (Popper, 1972; Michalski, 1983), *bias* (e.g., Utgoff, 1986), a *comparator* (Poole, 1989). These criteria are necessary for any act of induction because for any given consequent and a non-trivial hypothesis description language there could be a very large number distinct hypotheses that can be expressed in that language, and which satisfy the relation (4).

The selection criteria specify how to choose among them. Ideally, these criteria should reflect the properties of a hypothesis that are desirable from the viewpoint of the reasoner's (or learner's) goals. Often, these criteria (or bias) are partially hidden in the description language used. For example, the description language may be limited to only conjunctive statements involving a given set of attributes, or determined by the mechanism performing induction (e.g., a method that generates decision trees is automatically limited to using only operations of conjunction and disjunction in the hypothesis representation). Generally, these criteria reflect three basic desirable characteristics of a hypothesis: *accuracy*, *utility*, and *generality*.

The accuracy expresses a desire to find a "true" hypothesis. Because the problem is logically under-constrained, the "truth" of a hypothesis can never be guaranteed. One can only satisfy (4), which is equivalent to making a hypothesis *complete* and *consistent* with regard to the input facts (Michalski, 1983). If the input is noisy, however, an inconsistent and/or incomplete hypothesis may give a better overall predictive performance than a complete and consistent one (e.g., Quinlan, 1989; Bergadano et al., 1992). The utility requires a hypothesis to be computationally and/or cognitively simple, and be applicable to performing an expected set of problems. The generality criterion expresses the desire to have a hypothesis that is useful for predicting new unknown cases. The more general the hypothesis, the wider scope of different new cases it will be able to predict. Form now on, when we talk about inductive transmutations, we mean transmutations that involve admissible inductive inference.

While the above described view of induction is by no means universally accepted, it is consistent with many long-standing discussions of this subject going back to Aristotle (e.g., Adler and Gorman, 1987; see also the reference under Aristotle). Aristotle, and many subsequent thinkers, e.g., Bacon (1620), Whewell (1857) and Cohen (1970), viewed induction as a fundamental inference type that underlies all processes of creating new knowledge. They did not assume that knowledge is created only from low-level observations and without use of prior knowledge.

Based on the role and amount of background knowledge involved, induction, can be divided into *empirical induction* and *constructive induction*. Empirical induction uses little background knowledge. Typically, an empirical hypothesis employs the descriptors (attributes, terms, relations, descriptive concepts, etc.) that are selected from among those that are used in describing the input instances or examples, and therefore such induction is sometimes called *selective* (Michalski, 1983).

In contrast, a constructive induction uses background knowledge and/or experiments to generate additional, more problem-oriented descriptors, and employs them in the formulation of the hypothesis. Thus, it changes the description space in which hypotheses are generated. Constructive induction can be divided into *constructive generalization*, which produces knowledge-based hypothetical generalizations, *abduction*, which produces hypothetical domain-knowledge-based explanations, and *theory formation*, which produces general theories explaining a given set of facts. The latter is usually developed by employing inductive generalization with abduction and deduction.

There is a number of knowledge transmutations that employ induction, such as *empirical inductive generalization*, *constructive inductive generalization*, *inductive specialization*, *inductive concretion*, *abductive derivation*, and other (Michalski, 1993). Among them, the empirical inductive generalization is the most known form. Perhaps for this reason, it is sometimes mistakenly viewed as the only form of inductive inference.

Constructive inductive generalization creates general statements that use other terms than those used for characterizing individual observations, and is also quite common in human reasoning. Inductive specialization is a relatively lesser known form of inductive inference. In contrast to inductive generalization, it decreases the reference set described in the input.

Concretion is related to inductive specialization. The difference is that it generates more specific information about a given reference set, rather than reduces the reference set. Concretion is a transmutation opposite to abstraction. Abductive explanation employees abductive inference to derive properties of a reference set that can serve as its explanation.

Figure 5 gives examples of the above inductive transmutations.

A. Empirical generalization (BK limited; "pure" generalization)

Input: "A girl's face" and "Lvow cathedral" are beautiful paintings.
BK: "A girl's face" and "Lvow cathedral" are paintings by Dawski.
Hypothesis: All paintings by Dawski are beautiful.

B. Constructive inductive generalization (generalization + deduction)

Input: "A girl's face" and "Lvow cathedral" are beautiful paintings.
BK: "A girl's face" and "Lvow cathedral" are paintings by Dawski.
Dawski is a known painter. Beautiful paintings by a known painter are expensive.
Hypothesis: All paintings by Dawski are expensive.

C. Inductive specialization

Input: There is high-tech industry in Northern Virginia.
BK: Fairfax is a town in Northern Virginia.
Hypothesis: There is high-tech industry in Fairfax.

D. Inductive Concretion

Input: John is an expert in some formal science.
BK: John is Polish. Many Poles like logic. Logic is a formal science.
Hypothesis: John is an expert in logic.

E. Abductive derivation

Input: There is smoke in the house.
BK: Fire usually causes smoke.
Hypothesis: There is a fire in the house.

F General constructive induction (generalization plus abductive derivation)

Input: Smoke is coming from John's apartment.
BK: Fire usually causes smoke. John's apt. is in the Hemingway building.
Hypothesis: The Hemingway building is on fire.

Figure 5. Examples of inductive transmutations.

In Figure 5, examples A, C and D illustrate conclusive inductive transmutations (in which the generated hypothesis conclusively implies the consequent), and examples B, E and F illustrate contingent inductive transmutations (the hypothesis only plausibly implies the consequent).In example B, the input is only a plausible consequence of the hypothesis and BK, because background knowledge states that "Beautiful paintings by a known painter are expensive." This does not imply that all paintings that are expensive are necessarily beautiful.

The difference between inductive specialization (Example C) and concretion (Example D) is that the former reduces the set being described (that is, the reference set), and the latter increases the information about the reference set. In example C, the reference set is reduced from Virginia to Fairfax. In example D, the reference set is John; the concretion increases the amount of information about it.

HOW ABSTRACTION DIFFERS FROM GENERALIZATION

Generalization is sometimes confused with abstraction, which is often employed as part of the process of creating generalizations. These two transmutations are quite different, however, and both are fundamental operations on knowledge. This section provides additional explanation of abstraction, and illustrates the differences between it and generalization. As mentioned earlier, abstraction creates a less detailed description of a given reference set from a more detailed description, without changing the reference set. The last condition is important, because reducing information about the reference set by describing only a part of it would not be abstraction. For example, reducing a description of a table to a description of one of its legs would not be an abstraction operation.

To illustrate an abstraction transmutation, consider a transformation of the statement "My workstation has a Motorola 25-MHz 68030 processor" to "My workstation is quite fast." To make such an operation, the system needs domain-dependent background knowledge that "a processor with the 25-MHz clock speed can be viewed as quite fast," and a rule "If a processor is fast then the computer with that processor can be viewed as fast." Note that the more abstract description is a logical consequence of the original description in the context of the given background knowledge, and carries less information.

The abstraction operation often involves a change in the representation language, from one that uses more specific terms to one that uses more general terms, with a proviso that the statements in the second language are logically implied by the statements in the first language. A very simple form of abstraction is to replace in a description of an entity a specific attribute value (e.g., the length in a centimeter) by a less specific value (e.g., the length stated in linguistic terms, such as short, medium and long). A more complex abstraction would involve a significant change of the description language, e.g., taking a description of a computer in terms of electronic circuits and connections, and changing it into a description in terms of the functions of the individual modules.

In contrast to abstraction, which reduces information about a reference set but does not change it, generalization extends the reference set. To illustrate simply the difference between generalization and abstraction, consider a statement d(S,v), which says that attribute (descriptor) d takes value v for the set of entities S. Let us write such a statement in the form:

$$d(S) = v \quad (5)$$

Changing (5) to the statement d(S) = v', in which v' represents a more general concept, e.g., a parent node in a generalization hierarchy of values of the attribute d, is an abstraction operation. By changing v to v' less information is being conveyed about the reference set S. Changing (5) to a statement d(S') = v, in which S' is a superset of S, is a generalization operation. The generated statement conveys more information than the original one, because the property d is not assigned to a larger set.

For example, transferring the statement "color(my-pencil) = light-blue" into "color(my-pencil)=blue" is an abstraction operation. Such an operation is deductive, if one knows that light-blue is a kind of blue.

Transforming the original statement into "color(all-my-pencils) = light-blue" is a generalization operation. Assuming that one does not have prior knowledge that all writing instruments that I posses are blue, this is an inductive operation. Finally, transferring the original statement into "color(all-my-pencils)=blue" is both generalization and abstraction. Thus, associating the same information with a larger set is a generalization operation; associating a smaller amount of information with the same set is an abstraction operation.

In summary, generalization transforms descriptions along the set-superset dimension, and abstraction transforms descriptions along the level-of-detail dimension. Generalization often uses the same description space (or language), abstraction often involves a change in the representation space (or language). An opposite transmutation to generalization is specialization. An opposite transmutation to abstraction is concretion. Generalization is typically an inductive operation, and abstraction a deductive operation.

As a parallel concept to constructive induction, which was discussed before, one may introduce the concept of *constructive deduction*. Similarly to constructive induction, constructive deduction is a process of deductively transforming a source description into a target description, which uses new, more goal-relevant terms and concepts than the source description. As in constructive induction, the process uses background knowledge for that purpose. Depending on the available background knowledge, constructive deduction may be conclusive or contingent.

Abstraction can be viewed as a form of constructive deduction that reduces the amount of information about a given inference set, without changing it. Such a reduction may involve using terms at "higher level of abstraction" that are derived from the "lower level" terms. Constructive deduction is a more general concept than abstraction, as it includes any type of deductive knowledge derivation, including transformations of a given knowledge to equivalent but different forms, plausible deductive derivations, such as those based on probabilistic inferences (e.g., Schum, 1986; Pearl, 1988), or plausible reasoning (e.g., Collins and Michalski, 1989). In such cases, the distinction between constructive induction and constructive deduction becomes a matter of degree to which different forms of reasoning play the primary role.

A CLASSIFICATION OF LEARNING PROCESSES

Learning processes can be classified according to many criteria, such as the type of the inferential learning strategy used (in our terminology, the type of primary transmutation employed), the type of knowledge representation (logical expressions, decision rules, frames, etc.), the way information is supplied to a learning system (batch vs. incremental), the application area in which it is applied, etc. Classifications based on such criteria have been discussed in Carbonell, Michalski and Mitchell (1983) and Michalski (1986).

The Inferential Theory of Learning outlined above offers a new way of looking at learning processes, and suggests some other classification criteria. The theory considers learning as a knowledge transformation process whose primary purpose may be either to increase the amount of the learner's knowledge, or to increase the effectiveness of the knowledge already possessed. Therefore, the primary learning purpose can be used as a major criterion for classifying learning processes.

Based on this criterion, learning processes are divided into two categories—*synthetic* and *analytic*. The main goal of synthetic learning is to acquire new knowledge that goes beyond the knowledge already possessed, i.e., beyond its deductive closure. Thus, such learning relies on synthetic knowledge transmutations. The primary inference types involved in such processes are induction and/or analogy. (The term "primary" is important, because every inductive or analogical inference also involves deductive inference. The latter form is used, for example, to test whether a generated hypothesis entails the observations, to perform an analogical knowledge transfer based the hypothesized analogical match, to generate new terms using background knowledge, etc.)

The main goal of analytic learning processes is to transform knowledge that the learner already possesses into the form that is most desirable and/or effective for achieving the given learning goal. Thus, such learning relies on analytic knowledge transmutations. The primary inference type used is therefore deduction. For example, one may have a complete knowledge of how an automobile works, and therefore can in principle diagnose the problems based on it. By analytic learning, one can derive simple tests and procedures for more efficient diagnosis.

Other important criteria for classification of learning processes include:

- The *type of input information*—whether it is in the form of (classified) examples, or in the form of (unclassified) facts or observations.
- The *type of primary inference type* employed in a learning process—induction, deduction or analogy.
- The *role of the learner's background knowledge* in the learning process—whether learning relies primarily on the input data, primarily on the background knowledge, or on some balanced combination of the two.

Figure 6 presents a classification of learning processes according to the above criteria. A combination of specific outcomes along each criterion determines a class of *learning methodologies*. Individual methodologies differ in terms of the knowledge representation employed, the underlying computational mechanism, or the specific learning goal (e.g., learning rules for recognizing unknown instances, learning classification structures, or learning equations). Such methodologies like empirical generalization, neural-net learning and genetic algorithm based learning all share a general goal (knowledge synthesis), have input in the form of examples of observed facts (rather than rules or other forms of general knowledge), perform induction as the primary form of inference, and involve relatively small amount of background knowledge. The differences among them are in the employed knowledge representation and the underlying computational mechanism.

If the input to a synthetic learning method are examples classified by some source of knowledge, e.g., a teacher, then we have *learning from examples*. Such learning can be divided in turn into "instance-to-class" and "part-to-whole" categories (not shown in the Figure).

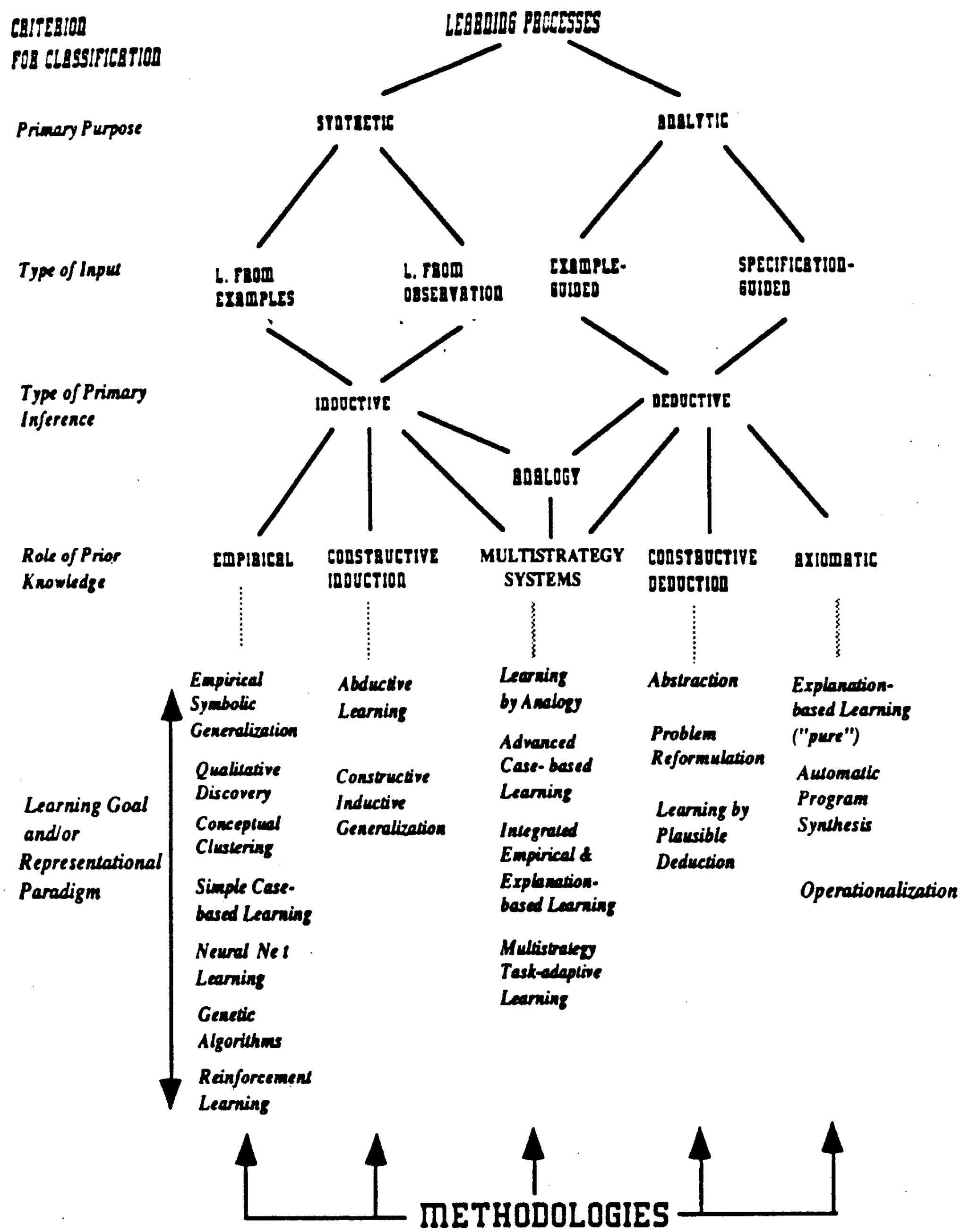


Figure 6. A general classification of learning processes.

In the "instance-to-class" category, examples are independent entities that represent a given class or concepts. For example, learning a general diagnostic rule for a given disease from characteristics of the patients with this disease is an "instance-to-class" generalization. Here each patient is an independent example of the disease.

In the "part-to-whole" category, examples are interdependent components that have to be investigated together in order to generate a concept description. For example, a "part-to-whole" inductive learning is to hypothesize a complete shape and look of a prehistoric animal from a collection of its bones.

When the input to a synthetic learning method includes facts that need to be described or organized into a knowledge structure, without the benefit of advise of a teacher, then we have *learning from observation*. The latter is exemplified by *learning by discovery*, *conceptual clustering* and *theory formation* categories.

The primary type of inference used in synthetic learning is induction. As described earlier, inductive inference can be empirical (background knowledge-limited) or constructive (background knowledge-intensive). Most work in empirical induction has been concerned with empirical generalization of concept examples using attributes selected from among those present in the descriptions of the examples. Another form of empirical learning includes *quantitative discovery*, in which learner constructs a set of equations characterizing given data.

Empirical inductive learning (both from examples, also called supervised learning, and from observation, also called unsupervised learning) can be done using several different methodologies, such as symbolic empirical generalization, neural net learning, genetic algorithm learning, reinforcement learning ("learning from feedback"), simple forms of conceptual clustering and case-based learning.

The above methods typically rely on (or need) relatively small amount of background knowledge, and all perform some form of induction. They differ from each other in the type of knowledge representation, computational paradigm, and/or the type of knowledge they aim to learn. Symbolic methods frequently use such representations as decision trees, decision rules, logic-style representations (e.g., Horn clauses or limited forms of predicate calculus), semantic networks or frames. Neural nets use networks of neuron-like units; genetic algorithms often use classifier systems. Conceptual clustering typically uses decision rules or structural logic-style descriptions, and aims at creating classifications of given entities together with descriptions of the created classes. Reinforcement learning acquires a mapping from situations to actions that optimizes some reward function, and may use a variety of representations, such a neural nets, sets of mathematical equations, or some domain-oriented languages (Sutton, 1992).

In contrast to empirical inductive learning, constructive inductive learning is knowledge-intensive. It uses background knowledge and/or search techniques to create new attributes, terms or predicates that are more relevant to the learning task, and use them to derive characterizations of the input. These characterizations can be generalizations, explanations or both.

As described before, abduction can be viewed as a form of knowledge-intensive (constructive) induction, which "traces backward" domain-dependent rules to create explanations of the given input. Many methods for constructive induction use decision rules for representing both background knowledge and acquired knowledge.

For completeness, we will mention also some other classifications of synthetic methods, not shown in this classification. One classification is based on the way facts or examples are presented to the learner. If examples (in *supervised* learning) or facts (in *unsupervised* learning) are presented all at once, then we have one-step or non-incremental inductive learning. If they are presented one by one, or in portions, so that the system has to modify the currently held hypothesis after each input, we have an incremental inductive learning.

Incremental learning may be with *no memory*, with *partial memory*, or with *complete memory* of the past facts or examples. Most incremental machine learning methods fall into the "no memory" category, in which all knowledge of past examples is incorporated in the currently held hypothesis. Human learning falls typically into a "partial memory" category, in which the learner remembers not only the currently held hypothesis, but also representative past examples supporting the hypothesis.

The second classification is based on whether the input facts or examples can be assumed to be totally correct, or can have errors and/or noise. Thus, we can have learning from a *perfect source*, or *imperfect* (*noisy*) source of information.

The third classification characterizes learning methods (or processes) based on the type of matching instances with concept descriptions. Such matching can be done in a *direct* way, which can be *complete* or *partial*, or an *indirect* way. The latter employs inference and a substantial amount of background knowledge. For example, rule-based learning may employ a direct match, in which any example has to exactly satisfy a condition part of some rule, or a partial match, in which a degree of match is computed, and the rule that gives the best match is fired Advanced *case-based* learning methods employ matching procedures that may conduct an extensive amount of inference to match a new example with past examples (e.g., Bareiss, Porter and Wier, 1990). Learning methods based on the *two-tiered concept representation* (Bergadano et al., 1992) also use inference procedures for matching an input with the stored knowledge. In both cases, the matching procedures perform a "virtual" generalization transmutation.

Analytic methods can be divided into those that are guided by an example in the process of knowledge reformulation (*example-guided*), and those that start with a knowledge specification (*specification-guided).* The former category includes ***explanation-based learning*** (e.g., DeJong et al., 1986), ***explanation-based-generalization*** (Mitchell et al., 1986), and ***explanation-based specialization*** (Minton et al., 1987; Minton, 1988). If deduction employed in the method is based on axioms, then it is called *axiomatic*. A "pure" explanation-based generalization is an example of an axiomatic method because it is based on a deductive process that utilizes a complete and consistent domain knowledge. This domain knowledge plays the role analogous to axioms in formal theories. Synthesizing a computer program from its formal specification is a specification-guided form of analytic learning.

Analytic methods that involve truth-preserving transformations of description spaces and/or plausible deduction are classified as methods of "constructive deduction." One important subclass of these methods are those utilizing abstraction as a knowledge transformation operation. Other subclasses include methods employing contingent deduction, e.g., plausible deduction, or probabilistic reasoning.

The type of knowledge representation employed in a learning system can be used as another dimension for classifying learning systems (also not shown in Figure 6). Learning systems can be classified according to this criterion into those that use a logic-style representation, decision tree, production rules, frames, semantic network, grammar, neural network, classifier system, PROLOG program, etc., or a combination of different representations. The knowledge representation used in a learning system is often dictated by the application domain. It also depends on the type of learning strategy employed, as not every knowledge representation is suitable for every type of learning strategy.

Multistrategy learning systems integrate two or more inferential strategies and/or computational paradigms. Currently, most multistrategy systems integrate some form of empirical inductive learning with explanation-based learning, e.g., Unimem (Lebowitz, 1986), Odysseus (Wilkins, Clancey, and Buchanan, 1986), Prodigy (Minton et al., 1987), GEMINI (Danyluk, 1987 and 1989), OCCAM (Pazzani, 1988), IOE (Dietterich and Flann, 1988) and ENIGMA (Bergadano et al., 1990). Some systems include also a form of analogy, e.g., DISCIPLE-1 (Kodratoff and Tecuci, 1987), or CLINT (Raedt and Bruynooghe, 1993). Systems applying analogy sometimes is viewed as multistrategy, because analogy is an inference combining induction and deduction. An advanced case-based reasoning system that uses different inference types to match an input with past cases can also be classified as multistrategy.

The Inferential Theory of Learning is a basis for the development of *multistrategy task-adaptive learning* (MTL), first proposed by Michalski (1990a). The aim of MTL is to synergistically integrate such strategies as empirical learning, analytic learning, constructive induction, analogy, abduction, abstraction, and ultimately also reinforcement strategies. An MTL system determines by itself which strategy or a combination thereof is most suitable for a given learning task.

In an MTL system, strategies may be integrated loosely, in which case they are represented as different modules, or tightly, in which case one underlying representational mechanism supports all strategies. Various aspects of research on MTL have been reported by Michalski (1990c) and by Tecuci and Michalski, (1991a,b). Related work was also reported by Tecuci (1991a,b; 1992).

Summarizing, the theory postulates that learning processes can be described in terms of generic patterns of inference, called transmutations. A few basic knowledge transmutations have been discussed, and characterized in terms of three dimensions:

A. The type of logical relationship between the input and the output: induction vs. deduction.
B. The direction of the change of the reference set: generalization vs. specialization.
C. The direction of the change in the level-of-detail of description: abstraction vs. concretion.

Each of the above dimensions corresponds to a different mechanism of knowledge transmutation that may occur in a learning process. The operations involved in the first two mechanisms, induction vs. deduction, and generalization vs. specialization, have been relatively well-explored in machine learning. The operations involved in the third mechanism, abstraction vs. concretion, have been relatively less studied. Because these three mechanisms are interdependent, not all combinations of operations can occur in a learning process . The problems of how to quantitatively and effectively measure the amount of change in the reference set and in the level-of-detail of descriptions are important topics for future research.

The presented classification of learning processes characterizes and relates to each other major subareas of machine learning. As any classification, it is useful only to the degree to which it illustrates important distinctions and relations among various categories. The ultimate goal of this classification effort is to show that diverse learning mechanisms and paradigms can be viewed as parts of one general structure, rather than as a collection of unrelated components.

SUMMARY

The goals of this research are to develop a theoretical framework and an effective methodology for characterizing and unifying diverse learning strategies and approaches. The proposed Inferential Theory looks at learning as a process of making goal-oriented knowledge transformations. Consequently, it proposes to analyze learning methods in terms of generic types of knowledge transformation, called transmutations, that occur in learning processes. Several transmutations have been discussed and characterized along three dimensions: the type of the logical relationship between an input and output (induction vs. deduction), the change in the reference set (generalization vs. specialization), and the change in the level-of-detail of a description (abstraction vs. concretion). Deduction and induction has been presented as two basic forms of inference. In addition to widely studied inductive generalization, other form of induction have been discussed, such as inductive specialization, concretion, and abduction. Is has been also shown that abduction can be viewed as a knowledge-based induction, and abstraction as a form of deduction.

The Inferential Theory can serve as a conceptual framework for the development of multistrategy learning systems that combine different inferential learning strategies. Research in this direction has led to the formulation of the multistrategy task-adaptive learning (MTL), that dynamically and synergistically adapts the learning strategy, or a combination of them, to the learning task.

Many of the ideas discussed are at a very early state of development, and many issues have not been resolved. Future research should develop more formal characterization of the presented transmutations, and develop effective methods for characterizing different knowledge transmutations, and measuring their "degrees." Other important research area is to determine how various learning algorithms and paradigms map into the described knowledge transmutations.

In conclusion, the ITL provides a new viewpoint for analyzing and characterizing learning processes. By addressing their logical capabilities and limitations, it strives to analyze and understand the competence aspects of learning processes. Among its major goals are to develop effective methods for determining what kind of knowledge a learner can acquire from what kind of inputs, to determine the areas of the most effective applicability of different learning methods, and to gain new insights into how to develop more advanced learning systems.

ACKNOWLEDGMENTS

The author expresses his gratitude to George Tecuci and Tom Arciszewski for useful and stimulating discussions of the material presented here. Thanks also go to many other people for their insightful comments and criticism of various aspects of this work, in particular, Susan Chipman, Hugo De Garis, Mike Hieb, Ken Kaufman, Yves Kodratoff, Elizabeth Marchut-Michalski, Alan Meyrowitz, David A. Schum, Brad Utz, Janusz Wnek, Jianping Zhang, and the students who took the author's Machine Learning class.

This research was done in the Artificial Intelligence Center of George Mason University. The research activities of the Center have been supported in part by the Office of Naval Research under grants No. N00014-88-K-0397, No. N00014-88-K-0226, No. N00014-90-J-4059, and No. N00014-91-J-1351, in part by the National Science Foundation under the grant No. IRI-9020266, and in part by the Defense Advanced Research Projects Agency under the grants administered by the Office of Naval Research, No. N00014-87-K-0874 and No. N00014-91-J-1854.

REFERENCES

Adler, M. J., Gorman, W. (Eds..) The Great Ideas: A Synopicon of Great Books of the Western World, Vol. 1, Ch. 39 (Induction), pp. 565-571, *Encyclopedia Britannica*, Inc., 1987.

Aristotle, Posterior Analytics, in *The Works of Aristotle*, Volume 1, R. M. Hutchins (Ed.), Encyclopedia Britannica, Inc., 1987.

Bacon, F., *Novum Organum*, 1620.

Bareiss, E. R., Porter, B. and Wier, C.C., PROTOS, An Exemplar-based Learning Apprentice, in *Machine Learning: An Artificial Intelligence Approach, Vol. III*, Carbonell, J.G., and Mitchell, T. M. (Eds.), Morgan Kaufmann, 1990.

Bergadano, F., Matwin, S., Michalski, R.S. and Zhang, J., Learning Two-tiered Descriptions of Flexible Concepts: The POSEIDON System, *Machine Learning Journal*, Vol. 8, No. 1, Januray 1992.

Carbonell, J. G., Michalski R.S. and Mitchell, T.M., An Overview of Machine Learning, in *Machine Learning: An Artificial Intelligence Approach*, Michalski, R.S., Carbonell, J.G., and Mitchell, T. M. (Eds.), Morgan Kaufmann Publishers, 1983.

Cohen, L.J., *The Implications of Induction*, London, 1970.

Collins, A. and Michalski, R.S., "The Logic of Plausible Reasoning: A Core Theory," *Cognitive Science*, Vol. 13, pp. 1-49, 1989.

Danyluk. A. P., "Recent Results in the Use of Context for Learning New Rules," *Technical Report No.* TR-98-066, Philips Laboratories, 1989.

DeJong, G. and Mooney, R., "Explanation-Based Learning: An Alternative View," *Machine Learning Journal*, Vol 1, No. 2, 1986.

Dietterich, T.G., and Flann, N.S., "An Inductive Approach to Solving the Imperfect Theory Problem," *Proceedings of 1988 Symposium on Explanation-Based Learning*, pp. 42-46, Stanford University, 1988.

Goodman, L.A. and Kruskal, W.H., *Measures of Association for Cross Classifications*, Springer-Verlag, New York, 1979.

Kodratoff, Y., and Tecuci, G., "DISCIPLE-1: Interactive Apprentice System in Weak Theory Fields," *Proceedings of IJCAI-87*, pp. 271-273, Milan, Italy, 1987.

Lebowitz, M., "Integrated Learning: Controlling Explanation," *Cognitive Science*, Vol. 10, No. 2, pp. 219-240, 1986.

Michalski, R.S., "A Planar Geometrical Model for Representing Multi-Dimensional Discrete Spaces and Multiple-Valued Logic Functions, " *Report No. 897*, Department of Computer Science, University of Illinois, Urbana, January 1978.

Michalski, R. S., "Theory and Methodology of Inductive Learning," *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (Eds.), Tioga Publishing Co., 1983.

Michalski, R.S., Understanding the Nature of Learning: Issues and Research Directions, in *Machine Learning: An Artificial Intelligence Approach Vol. II*, Michalski, R.S., Carbonell, J.G., and Mitchell , T. M. (Eds.), Morgan Kaufmann Publishers, 1986.

Michalski, R.S., Toward a Unified Theory of Learning: Multistrategy Task-adaptive Learning, *Reports of Machine Learning and Inference Laboratory MLI-90-1*, January 1990a.

Michalski, R.S.and Kodratoff, Y. "Research in Machine Learning: Recent Progress, Classification of Methods and Future Directions," in *Machine Learning: An Artificial Intelligence Approach, Vol. III*, Kodratoff, Y. and Michalski, R.S. (eds.), Morgan Kaufmann Publishers, Inc., 1990b.

Michalski, R.S., LEARNING FLEXIBLE CONCEPTS: Fundamental Ideas and a Method Based on Two-tiered Representation, in *Machine Learning: An Artificial Intelligence Approach, Vol. III*, Kodratoff, Y. and Michalski, R.S. (eds.), Morgan Kaufmann Publishers, Inc., 1990c.

Michalski, R.S., INFERENTIAL THEORY OF LEARNING: Developing Foundations for Multistrategy Learning, in *Machine Learning: A Multistrategy Approach, Vol. IV*, R.S. Michalski and G. Tecuci (Eds.), Morgan Kaufmann, 1993.

Minton, S., "Quantitative Results Concerning the Utility of Explanation-Based Learning," *Proceedings of AAAI-88*, pp. 564-569, Saint Paul, MN, 1988.

Minton, S., Carbonell, J.G., Etzioni, O., et al., "Acquiring Effective Search Control Rules: Explanation-Based Learning in the PRODIGY System," *Proceedings of the 4th International Machine Learning Workshop*, pp. 122-133, University of California, Irvine, 1987.

Mitchell, T.M., Keller,T., Kedar-Cabelli,S., "Explanation-Based Generalization: A Unifying View," *Machine Learning Journal*, Vol. 1, January 1986.

Pazzani, M.J., "Integrating Explanation-Based and Empirical Learning Methods in OCCAM," *Proceedings of EWSL-88*, pp. 147-166, Glasgow, Scotland, 1988.

Pearl J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

Piatetsky-Shapiro, G., "Probabilistic Data Dependencies," *Proceedings of the ML92 Workshop on Machine Discovery*, J.M. Zytkow (Ed.), Aberdeen, Scotland, July 4, 1992.

Popper, K. R., *Objective Knowledge: An Evolutionary Approach*, Oxford at the Clarendon Press, 1972.

Poole, D., Explanation and Prediction: An Architecture for Default and Abductive Reasoning, *Computational Intelligence*, No. 5, pp. 97-110, 1989.

Porter, B. W. and Mooney, R. J. (eds.), *Proceedings of the 7th International Machine Learning Conference*, Austin, TX, 1990.

De Raedt, L. and Bruynooghe, M. CLINT: A Multistrategy Interactive Concept Learner, in *Machine Learning: A Multistrategy Approach, Vol. IV*, R.S. Michalski and G. Tecuci (Eds.), Morgan Kaufmann, 1993 (to appear).

Rumelhart, D. E., McClelland and the PDP Research Group, Parallel Distributed Processing, Vol, 1 & 2, A Bradford Book, The MIT Press, Cambridge, Massachusetts, 1986.

Russell, S., The Use of Knowledge in Analogy and Induction, *Morgan Kaufman Publishers, Inc.*, San Mateo, CA, 1989.

Schafer, D., (Ed.), *Proceedings of the 3rd International Conference on Genetic Algorithms*, George Mason University, June 4-7, 1989.

Schum, D.,A.,"Probability and the Processes of Discovery, Proof, and Choice," *Boston University Law Review*, Vol. 66, No 3 and 4, May/July 1986.

Segre, A. M. (Ed.), *Proceedings of the Sixth International Workshop on Machine Learning*, Cornell University, Ithaca, New York, June 26-27, 1989.

Sutton, R. S. (Ed.), Special Issue on Reinforcement Learning, *Machine Learning Journal*, Vol. 8, No. 3/4, May 1992.

Tecuci G., "A Multistrategy Learning Approach to Domain Modeling and Knowledge Acquisition," in Kodratoff, Y., (ed.), *Proceedings of the European Conference on Machine Learning*, Porto, Springer-Verlag, 1991a.

Tecuci G., "Steps Toward Automating Knowledge Acquisition for Expert Systems," in Rappaport, A., Gaines, B., and Boose, J. (Eds.), *Proceedings of the AAAI-91 Workshop on Knowledge Acquisition "From Science to Technology to Tools"*, Anaheim, CA, July, 1991b.

Tecuci, G. and Michalski, R.S.,"A Method for Multistrategy Task-adaptive Learning Based on Plausible Justifications," in Birnbaum, L., and Collins, G. (eds.) *Machine Learning: Proceedings of the Eighth International Workshop*, San Mateo, CA, Morgan Kaufmann, 1991a.

Tecuci G., and Michalski R.S., Input "Understanding" as a Basis for Multistrategy Task-adaptive Learning, in Ras, Z., and Zemankova, M. (eds.), *Proceedings of the 6th International Symposium on Methodologies for Intelligent Systems*, Lecture Notes on Artificial Intelligence, Springer Verlag, 1991b.

Touretzky, D., Hinton, G., and Sejnowski, T. (Eds.), *Proceedings of the 1988 Connectionist Models*, Summer School, Carnegie Mellon University, June 17-26, 1988.

Utgoff, P. Shift of Bias for Inductive Concept Learning, in *Machine Learning: An Artificial Intelligence Approach Vol. II,* Michalski, R.S., Carbonell, J.G., and Mitchell, T. M. (Eds.), Morgan Kaufmann Publishers, 1986.

Warmuth, M. & Valiant, L. (Eds.) (1991). *Proceedings of the 4rd Annual Workshop on Computational Learning Theory*, Santa Cruz, CA: Morgan Kaufmann.

Whewell, W., *History of the Inductive Sciences*, 3 vols., Third edition, London, 1857.

Wilkins, D.C., Clancey, W.J., and Buchanan, B.G., *An Overview of the Odysseus Learning Apprentice*, Kluwer Academic Press, New York, NY, 1986.

Wnek, J., Sarma, J., Wahab, A. A. and Michalski,R.S., COMPARING LEARNING PARADIGMS VIA DIAGRAMMATIC VISUALIZATION: A Case Study in Concept Learning Using Symbolic, Neural Net and Genetic Algorithm Methods, *Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems*, University of Tennessee, Knoxville, TN, North-Holland, October 24-27, 1990.

Wnek, J. and Michalski, R.S., COMPARING SYMBOLIC AND SUBSYMBOLIC LEARNING: A Case Study, in *Machine Learing: A Multistrategy Approach, Volume IV*, R.S. Michalski and G. Tecuci (Eds.), Morgan Kaufmann, 1993.