Inferential Theory of Learning as a Conceptual Basis for Multistrategy Learning

RYSZARD S. MICHALSKI   MICHALSKI@AIC.GMU.EDU
Center for Artificial Intelligence, George Mason University, Fairfax, VA 22030

Abstract. In view of a great proliferation of machine learning methods and paradigms, there is a need for a general conceptual framework that would explain their interrelationships and provide a basis for their integration into multistrategy learning systems. This article presents initial results on the *Inferential Theory of Learning* that aims at developing such a framework, with the primary emphasis on explaining logical capabilities of learning systems, i.e., their *competence*. The theory views learning as a goal-oriented process of modifying the learner's knowledge by exploring the learner's experience. Such a process is described as a search through a *knowledge space*, conducted by applying knowledge transformation operators, called *knowledge transmutations*. Transmutations can be performed using any type of inference—deduction, induction, or analogy. Several fundamental pairs of transmutations are presented in a novel and very general way. These include generalization and specialization, explanation and prediction, abstraction and concretion, and similization and dissimilization. Generalization and specialization transmutations change the *reference set* of a description (the set of entities being described). Explanations and predictions derive additional knowledge about the reference set (explanatory or predictive). Abstractions and concretions change the level of detail in describing a reference set. Similizations and dissimilizations hypothesize knowledge about a reference set based on its similarity or dissimilarity with another reference set. The theory provides a basis for *multistrategy task-adaptive learning* (MTL), which is outlined and illustrated by an example. MTL dynamically adapts strategies to the *learning task*, defined by the input information, the learner's background knowledge, and the learning goal. It aims at synergistically integrating a wide range of inferential learning strategies, such as empirical and constructive inductive generalization, deductive generalization, abductive derivation, abstraction, similization, and others.

Keywords. Learning theory, multistrategy learning, inference, classification of inference, deduction, induction, abduction, generalization, abstraction, analogy, transmutation.

> *For every belief comes either through syllogism or from induction.*
> Aristotle, Prior Analytics, Book II, Chapter 23 (p. 90)
> ca 330 BC.

1. Introduction

Most research in machine learning has been oriented toward the development of *monostrategy* methods that employ one type of inference and a single computational mechanism. Such methods include, for example, inductive learning of decision rules or decision trees, explanation-based generalization, empirical discovery, neural net learning from examples, genetic algorithm-based learning, conceptual clustering, and others. The research progress on these and related topics has been reported by many authors, among them Laird (1988), Touretzky, Hinton, and Sejnowski (1988), Goldberg (1989), Schafer (1989), Segre (1989),

Rivest, Haussler, and Warmuth (1989), Fulk and Case (1990), Porter and Mooney (1990), Kodratoff and Michalski (1990), Birnbaum and Collins (1991), Warmuth and Valiant (1991), and Sleeman and Edwards (1992).

With the growing understanding of capabilities and limitations of monostrategy methods, there has been an increasing interest in *multistrategy learning* systems that employ two or more inference types and/or computational mechanisms. Multistrategy systems can have potentially for much greater *competence*, that is, the ability to solve a much wider range of learning problems than monostrategy systems, because they take advantage of the complementarity of individual learning strategies. On the other hand, they are also potentially significantly more complex, and thus their implementation presents a much greater challenge. Therefore, a decision on their application to a given range of problems should reflect the above trade-off. Since human learning is clearly multistrategy, research on multistrategy systems is of significant relevance to understanding human learning, and thus has a great importance regardless of the practical applications of the work.

Among early well-known multistrategy systems (sometimes called "integrated learning systems") are UNIMEM (Lebowitz, 1986), Odysseus (Wilkins, Clancey, & Buchanan, 1986), Prodigy (Minton et al., 1987) DISCIPLE (Kodratoff & Tecuci, 1987), Gemini (Danyluk, 1987, 1989, 1993), OCCAM (Pazzani, 1988), IOE (Dietterich & Flann, 1988), and KBL (Whitehall, 1990). Most of these systems are concerned with integrating symbolic empirical induction with explanation-based learning. Some, like DISCIPLE, also include a simple method for analogical learning. The integration of the strategies is often done in a predefined, problem-independent way, and without clear theoretical justification. Some recent multistrategy systems are described by De Raedt and Bruynooghe (1993) and Mooney and Ourston (1993).

An open and challenging problem is how to integrate a whole spectrum of learning strategies in a problem-dependent way, and on the basis of sound and general theoretical foundations. By a problem-dependent integration we mean an integration in which a learning strategy, or a combination of them, is automatically adapted to different learning situations.

The articles in this special issue represent novel and diverse efforts in this general direction. The present article investigates principles characterizing diverse inferential learning strategies and proposes a general conceptual framework for developing multistrategy systems. Its central part is devoted to *Inferential Theory of Learning (ITL)*, which analyzes learning processes in terms of generic operators for knowledge transformation, called *knowledge transmutations* or *knowledge transforms*.

A knowledge transmutation is a conceptually simple, high-level knowledge transformation pattern that derives a piece of knowledge from a given input and background knowledge. A transmutation is an operator on knowledge that can employ any type of inference. Among major research topics studied in the theory are an analysis of the properties of different knowledge transmutations, their interrelationships, and their role in different learning algorithms, methods, and paradigms.

The main objective of the theory is to characterize the logical capabilities of learning methods and processes, that is, their *competence*. To this end, the theory addresses such questions as what types of knowledge transformation occur in learning processes; what is the "truth" status of knowledge generated by them; what knowledge transmutations occur in different types of learning; how prior knowledge is used; what types of logical relationships

exist between the learned knowledge, the input information, and the learner's prior knowledge; how learning goals and their structure influence learning processes; how learning processes can be classified and evaluated from the logical viewpoint; what is the validity of the learned knowledge; etc. The theory stresses the use of multitype inferences in learning processes, the role of the learner's prior knowledge, and the importance of learning goals. Although the primary goal is not to develop a cognitive learning theory, the conceptual framework presented does try to formally capture the intuitive perceptions of different forms of human inference and learning, and suggests solutions that could be used as a basis for developing cognitive models. A number of ideas presented here stem from research on the core theory of human plausible inference (Collins & Michalski, 1989).

The above goals distinguish the Inferential Theory of Learning (ITL) from the Computational Learning Theory (COLT), which focuses on the *computational complexity* and *convergence* of learning algorithms, particularly those for empirical inductive learning. COLT has not yet been much concerned with multistrategy learning, the role of the learner's prior knowledge, or the learning goals (e.g., Fulk & Chase; 1990; Warmuth & Valiant, 1991). The above should not be taken to mean that the issues studied in COLT are unimportant, but only that they are different. A "unified" theory of learning should take into consideration both competence and complexity of learning processes.

This article outlines basic ideas of ITL and presents a novel and more general view of several fundamental knowledge transmutations, such as generalization, abduction, abstraction, similization, and their opposites. The last section describes briefly an application of the theory to the development of a methodology for *multistrategy task-adaptive learning*. Many ideas are presented in prefatory and illustrative fashion, with the emphasis on developing a general perspective of the whole subject. Various details and a better formalization of various concepts await further research. To make the article easily accessible to the AI and Cognitive Science communities, as well as to readers who are not regularly using predicate logic, expressions in predicate logic are usually accompanied by a natural language interpretation. The presented work is a significant extension or refinement of ideas described in earlier publications (Michalski, 1983, 1990a,b, 1991).

2. Basic tenets of the inferential theory of learning

Learning has been traditionally characterized as an improvement of a system's behavior or knowledge due to its experience. While this view is appealing due to its simplicity, it does not provide many clues about how to actually implement a learning system. To build a learning system, one needs to understand, in computation terms, what types of behavior or knowledge changes occur in learning, and how they are accomplished in response to different types of experience. (Here "experience" means the totality of information generated in the course of performing some actions, not a physical process.)

To provide answers to such questions, the Inferential Theory of Learning assumes that learning is a goal-guided process of modifying the learner's knowledge by exploring the learner's experience. Such a process can be viewed as a search through a *knowledge space*, defined by the knowledge representation used. The search can employ any type of inference—any form of deduction, induction, or analogy. It involves "background knowledge,"
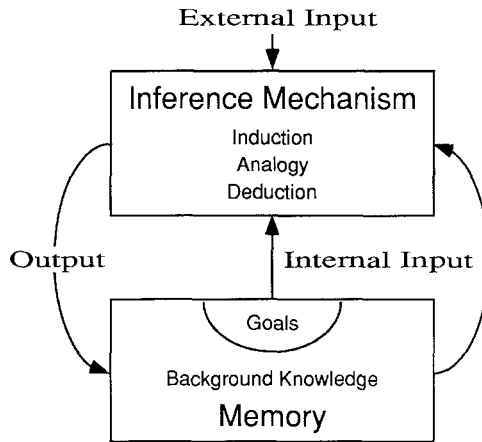
5

External Input

Inference Mechanism

Induction
Analogy
Deduction

Output   Internal Input

Goals

Background Knowledge
Memory

Figure 1. An illustration of a general learning process.

that is, the relevant parts of the learner's prior knowledge. Consequently, the information flow in a learning process can be characterized by a general schema shown in figure 1. In each learning cycle, the learner analyses the input information in terms of its background knowledge and its goals, and performs various inferences to generate new knowledge and/or a better form of knowledge. The "new knowledge" may be provided from the outside, or may be generated by induction, deduction, or analogy. It can be additional knowledge, or a change in the belief in the knowledge already possessed. (For further discussion of this topic, and a distinction between different types of new knowledge—"intrinsic" vs. "derived"—see section 4.) If the results of a given learning act are satsifactory from the viewpoint of the learning goal(s), they are assimilated within the learner's memory and become available for use in subsequent learning processes.

The basic premise of the Inferential Learning Theory is that in order to learn, an agent has to be able to perform inference and to have memory that both stores the background knowledge (BK) needed for performing the inference and records "useful" results of inference. Without either of the two components—the ability to reason and the ability to memorize and retrieve information from memory—no learning can be accomplished. Thus, one can write an "equation":

Learning = Inferencing + Memorizing

It should be noted that the term "inferencing" is used here in a very general sense, meaning any type of reasoning or knowledge transformation. The double role of memory, as a supplier of background knowledge and as a depository of results, is often reflected in the organization of a learning system. For example, in a neural net, background knowledge resides in the structure of the network, i.e., in the type and the number of units used, in the way they are interconnected, and in the initial weights of the connections. The learned knowledge usually resides only in the new values of the weights. In decision tree learning, the BK includes the set of attributes used to describe objects, the domains of the attributes,

and an attribute evaluation method. The knowledge created is in the form of a decision tree. In a "self-contained" rule-learning system, all background knowledge and learned knowledge would be in the form of rules. A learning process would involve modifying prior rules and/or creating new ones. The ultimate learning capabilities of a learning system are determined by what it can or cannot change in its knowledge base, and by what kinds of inference and knowledge transformations it is capable of performing.

The Inferential Learning Theory postulates that a learning process depends on the input information (input), background knowledge (BK), and learning goal. These three components constitute a *learning task*. The learning task determines what type of learning strategy or strategies need to be employed (by "learning strategy" is roughly meant a combination of the type of inference employed and the computational and the representational mechanism used to implement it). An input to any step of learning can be sensory observations, knowledge communicated by a source (e.g., a teacher), or knowledge generated by the previous learning step. Although most learning systems assume that input is in the form of facts or concept examples, a learning system should be able to learn from any type of knowledge, including previously formed generalizations, conceptual hierarchies, mathematical equations, knowledge at any level of abstraction, estimates of certainty of given knowledge, or any combination of various types.

The key idea of the theory is that processes involved in accomplishing a learning goal can be characterized in terms of generic patterns of inference, called knowledge transmutations. A knowledge transmutation takes an input and background knowledge as arguments and generates a new piece of knowledge. It represents conceptually simple and comprehensible units of knowledge transformation, and it can employ any type of inference. Specifically, the theory views a learning process as a search through a knowledge space, conducted by applying knowledge transmutations as search operators:

Given:
• Input knowledge   (I)
• Goal   (G)
• Background knowledge   (BK)
• Transmutations   (T)
Determine:
• Output knowledge O that satisfies goal G, by applying transmutations from the set T to input I and/or background knowledge BK.

By the input knowledge (I) is meant here any information (facts, examples, general knowledge, etc.) that the learner receives from the environment or as results of previous steps of learning. Goal (G) specifies criteria to be satisfied by the Output Knowledge (O) in order that learning is accomplished. Background knowledge (BK) is a part of learner's prior knowledge that is relevant to a given learning process. While complete and formal definition of "relevant" knowledge goes beyond the scope of this article, as a working definition the reader may assume that it is a part of prior knowledge that can be useful at any stage of a given learning process.

Transmutations are operators that make knowledge changes in the knowledge space. The knowledge space is a space of knowledge representations that can represent all possible

inputs, all of the learner's background knowledge, and all knowledge that the learner can potentially generate. In the context of empirical inductive learning, the knowledge space is usually called a *description space*.

For illustration, here are a few examples of transmutations. An *inductive generalization* takes one or more concept examples and creates a general description of them. An explanation-based generalization is a form of deductive generalization that takes an example from an "operational" description space, a concept description from an "abstract" description space, and relevant domain knowledge, and derives from them a concept description in the operational description space (see, e.g., Mitchell, Keller, & Kedar-Cabelli, 1986). Sections 4 to 7 define and discuss several fundamental knowledge transmutations, such as generalization, abstraction, and similization, and their counterparts, specialization, concretion, and dissimilization. Other types of transmutations, due to space limitations, are discussed only very briefly.

The analysis and explanation of diverse learning processes in terms of underlying knowledge transmutations is a major topic of the Inferential Theory of Learning. The transmutations represent transformations of various aspects of knowledge, and they can be implemented in many different ways. Depending on the knowledge representation and the computation mechanism, knowledge transmutations can be performed explicitly or implicitly. In symbolic learning systems, transmutations are usually (but not always) implemented in a more or less explicit way and executed in steps that are conceptually comprehensible. For example, the INDUCE learning system performs inductive generalization according to certain generalization rules—selective or constructive—where each rule represents a conceptually simple transformation (Michalski, 1983; Bloedorn & Michalski, 1991; Wnek & Michalski, 1991a).

In neural networks, transmutations are performed implicitly, in steps dictated by the underlying computational mechanism. These steps may not correspond to any conceptually simple operations or rules of inference. For example, a neural network may generalize an input example by performing a sequence of small modifications of weights of internode connections. Individual weight modifications may be difficult to explain in terms of explicit inference rules. Nevertheless, they can produce a global effect equivalent to generalizing a set of examples.

Such an effect can be easily demonstrated by a method for *diagrammatic visualization* (DIAV). In DIAV, concepts are mapped into sets of cells in a planar diagram, which represents a multidimensional space spanned over multivalued attributes. Operations on concepts are visualized by changes in the configurations of the corresponding sets of cells. Examples of diagrammatic visualization of inductive generalizations performed by a neural network, a genetic algorithm, and symbolic learning systems are presented by Wnek and Michalski (1991b, 1993).

A learning goal is a necessary component of any learning process. Given an input, and some nontrivial background knowledge, a learner could potentially generate an unbounded number of inferences. To limit the proliferation of choices, a learning process is constrained and/or guided by the learning goal. A learning goal determines what parts of prior knowledge are relevant, what knowledge is to be acquired, in which form, and how the learned knowledge is to be evaluated. There can be many different types of learning goals, and they can be expressed explicitly or implicitly. Goals can be classified roughly as domain independent or domain dependent.

Domain-independent goals call for a certain type of learning activity, independent of the topic of discourse, e.g., to acquire a general rule for classifying given facts, to confirm a given piece of knowledge, to derive from it some other knowledge, to concisely describe given observations, to discover a regularlity in a collection of data, to find a causal explanation of a found regularity, to acquire control knowledge, to reformulate given knowledge to a more effective form, to solve a problem of a given type, to plan what to learn, etc. Domain-dependent goals call for acquiring a problem of a given type, to plan what to learn, etc. Doman-dependent goals call for acquiring a specific piece or type of domain knowledge. A learner may pursue more than one goal, and the goals may be conflicting. When they are conflicting, their relative importance controls the amount of effort that is extended to pursue any of them. The relative importance of specific goals depends on the importance of higher-level goals. Thus, learning processes are controlled by a hierarchy of goals and the estimated degrees of their importance.

Most machine learning research has so far given relatively little consideration to the problem of learning goals and how they affect learning processes. As a result, many developed systems are method oriented rather than problem oriented. There have been, however, several investigations of the role and the use of goals in learning and inference (e.g., Stepp & Michalski, 1983; Hunter, 1990; Ram, 1991; Ram & Hunter, 1992). Among the important research problems related to this topic are the development of methods for goal representation, for using goals to guide a learning process, and to understand the interaction and conflict resolution among domain-independent and domain-specific goals. These issues are of significant importance to the understanding of learning in general, and interest in them will likely increase in the future.

In sum, Inferential Learning Theory states that learning is a goal-guided process of deriving desired knowledge by using input information and background knowledge. Such a process involves a search through a knowledge space, using knowledge transmutations as operators. Knowledge transmutations can involve any type of inference.

3. Types of inference

The central property of any knowledge transmutation is the type of underlying inference. The inference type characterizes the transmutation along the truth–falsity dimension, and thus determines the validity of the knowledge derived by it. For this reason, before discussing different knowledge transmutations, we will first analyze basic types of inference that can be involved in them.

Any type of inference can produce some useful knowledge worth remembering for future use. Consequently, a complete learning theory has to include a complete theory of inference. Such a theory of inference has to account for all possible types of inference. To this end, figure 2 presents an attempt to illustrate schematically all major types of inference. The first classification divides inferences into two fundamental types: deductive and inductive. In defining these types, many conventional approaches do not distinguish between the input information and the reasoner's prior knowledge. Such a distinction, however, is important from the learning theory viewpoint, because in a given learning process, there is clearly a difference between the knowledge already possessed by the learner and the information communicated from the outside.
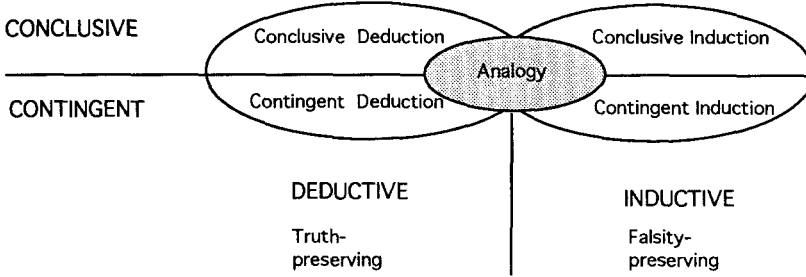
Figure 2. A classification of major types of inference.

To characterize basic types of inference in a general, language-independent way, consider the entailment

$$P \cup PK \models C \tag{1}$$

where P stands for a set of statements, called the *premise*, BK stands for a set of statements representing the reasoner's background knowledge, and C stands for a set of statements, called the *consequent*. P is assumed to be consistent with BK.

Deductive inference is deriving consequent C, given P and BK. Inductive inference is hypothesizing premise P, given C and BK. Thus, deduction can be viewed as "tracing forward" the relationship (1), and induction as "tracing backward" this relationship. Because (1) succinctly explains the relationship between two fundamental forms of inference, it is called the *fundamental equation* for inference.

Deduction is truth-preserving, i.e., C must be true if P and BK are true, and induction is falsity-preserving, i.e., if C is false, then P must be false also, if BK is true. (The latter property applies to every type of induction such as inductive generalization, abduction, inductive specialization, concretion, and others—see section 5).

In a general view of deduction and induction that also captures their approximate or common sense forms, the "strong" entailment $\models$ in (1) may be replaced by a "weak" entailment. A weak entailment includes cases in which C is only a *plausible, probabilistic, or partial* consequence of P and BK. The difference between a "strong" (valid) and "weak" (plausible) entailment leads to another major classfication of types of inference.

Specifically, inferences can be *conclusive* or *contingent*. Conclusive inferences assume "strong" entailment; they are true in every possible situation. Contingent inferences assume only "weak" entailment; they may be true in some situations and not true in others. Conclusive deductive inferences are "strongly" truth-preserving, and conclusive inductive inferences are "strongly" falsity-preserving. Contingent deductive inferences are "weakly" truth-preserving, and contingent inductive inferences are "weakly" falsity-preserving.

If inductive inference produces a statement that characterizes a larger set of entities than the input statement (C), then it is called an *inductive generalization*. As shown below, such an inference is based on tracing backward a domain-independent (tautological) implication, known as the rule of *universal specialization*. If it hypothesizes a premise that explains the input, then it is called *abduction*. Abduction is based on tracing backward domain-dependent implications. These distinctions are illustrated by the examples below.

10

A conclusive deductive inference is illustrated by the following transmutation:

Input $\alpha \in X$ (α is an element of X.)
BK $\forall x \in X, q(x)$ (All elements of X have property q.)
          $(\forall x \in X, q(x)) \Rightarrow (\alpha \in X \Rightarrow q(\alpha))$ (If all elements of X have property q,
                        then any element of X, e.g., α, must have property q.

Output $q(\alpha)$ (α has property q.)

If Input is the premise P, and Output is the consequent C, then the fundamental equation (1) is clearly satisfied. In constrast, the following transmutation illustrates conclusive induction:

Input $q(\alpha)$ (α has property q.)
BK $\alpha \in X$ (α is an element of X.)
          $(\forall x \in X, q(x)) \Rightarrow (\alpha \in X \Rightarrow q(\alpha))$ (If all elements of X have property q,
                        then any element of X, e.g., α, must have property q.

Output $\forall x \in X, q(x)$ (*Maybe* all elements of X have property q.)

The Output is obtained by tracing backward a tautological implication (listed as part of BK), known in logic as the rule of universal specialization. If Input is the consequent C and the Output is the premise P, then the fundamental equation (1) is satisfied, because the union of sentences in Output and BK entails the Input. The inference is falsity-preserving, because if the Input were not true (it did not have the property q), then the hypothetical premise (Output) would have to be false. This form of induction is called *generalization* because it hypothesizes a statement in which the property that characterized only one element (α) now characterizes a larger set (X). The output from induction is uncertain, which here, and henceforth is indicated by the qualifier "*Maybe.*"

To proceed, we will introduce two important concepts, a *reference set* and a *descriptor*. A reference set of a statement (or set of statements) is an entity or a set of entities that this statement(s) describes or refers to. A descriptor is an attribute, a relation, or a transformation whose instantiation (value) is used to characterize the reference set or the individual entities in it. For example, consider a statement: "Nicholas is of medium height, has a Ph.D. in astronomy from the Jagiellonian University, and likes travel." The reference set here is the singleton "Nicholas." The sentence uses three descriptors: a one-place attribute, "height(person)"; a binary relation, "likes(person,activity)"; and a four-place relation, "degree-received(person, degree, topic, university)."

Consider another statement: "Most people on Barbados and Dominica have beautiful dark skin." Here the reference set is "Most people on Barbados and Dominica," and the descriptors are "skin-color(person)" and "skin-attractiveness(person)." What is the reference set and what are descriptors in a statement or set of statements may be a matter of interpretation and/or context. However, once the interpretation is decided, other concepts below can be consistently applied.

Using the above concepts, a generalization is characterized as a transmutation that extends the reference set of the input statement(s). Abduction (also called *explanation*) is characterized as a transmutation that hypothesizes explanations of the properties of the reference set, but does not change the set. An opposite of abduction is *prediction*, which derives consequences of the properties of the reference set. Here is an illustration of an abduction:

Input	$q(\alpha)$	(α has property q.)
BK	$\forall x \in X, q(x)$	(All elements of X have property q.)
	$(\forall x \in X, q(x)) \Rightarrow (\alpha \in X \Rightarrow q(\alpha))$	

Output	$\alpha \in X$	(*Maybe* α is an element of X.)

The *Input* states that the reference set α has the property q. The abductive transmutation hypothesizes that α is an element of X, which is the context of BK can be viewed as an explanation of the Input: $q(\alpha)$ *because* $\alpha \in X$. The fundamental equation (1) holds, because, if Output is true, then Input must also be true in the context of BK. Again, if Input were not true, then Output could not be true; thus the inference preserves falsity. As in generalization, Output was obtained by "tracing backward" a rule in BK ($\alpha \in X \Rightarrow q(\alpha)$).

In both cases above, generalization and abduction, Output plus BK (strongly) entails Input. These forms are therefore called *conclusive induction* and *conclusive abduction*, respectively (as opposed to *contingent induction and contingent abduction*, described below).

In the literature, some authors restrict abduction to processes of creating the "best" explanation of a given fact, i.e., to inferences involving "tracing backwards" the "strongest" implication, whose right-hand side states the given fact. A difficulty with this view is that it is not always easy to determine which explanation among the alternative ones is the "best." If producing an alternative but not the "best" explanation is not abduction, then what is and what is not abduction depends on the measure of "goodness" of explanation, rather than on logical properties of inference. Some authors also require that the explanation is "causal." Since the rule "All elements of X have property q" (or, "If an entity belongs to X, then it has property q") does not express a causal relationship, then the reasoning illustrated in the example above would not be classified as abduction.

This article views the concept of abduction in a more general sense. Specifically, abduction is viewed as an inference that generates explanations of a given reference set, and is done by tracing backward certain domain-dependent implications. If these implications represent causal relationships, then abduction produces causal explanations. Peirce (1965), who originally introduced the concept of abduction, did not restrict it to reasoning producing only the "best" or only "casual" explanations (for a discussion on the relationship between abduction and deduction, see Console, Theseider, & Torasso, 1991). Some theoretical views on abduction are found in Zadrozny (1991). For an analysis and development of casual reasoning in humans, see Shultz and Kestenbaum (1985).

Conventional definitions of abduction describe it as "tracing backwards" an implicative rule. We will now show that such a view makes a tacit assumption that, if violated, allows abduction to produce completely implausible inferences. Consider, for example, the following inference:

Input	Color(My-Pencil, Green)	(My pencil is green.)
BK	Type (object, Grass) $\Rightarrow$ Color(object, Green)	(If an object is grass then it is green.)

Output	Type(My-Pencil, Grass)	(*Maybe* my pencil is grass.)

Although the implication in BK is quite strong, the abductive derivation based on it is implausible. The reason for this is that the implication

Color(object, Green) $\Rightarrow$ Type(object, grass)   (If an object is green then it is grass.)

normally holds only with an infinitesimal likelihood. Thus, abduction can produce an implausible hypothesis if the reverse implication has insufficient "strength." This simply means that abductive derivation makes an assumption about the "reverse strength" of an implication employed in it. To make this issue explicit, we introduce a more general form of implication as a basis for abductive derivation.

Definition. A *mutual implication* or, for short, an *m-implication*, describes a logical dependency between statements (well-formed predicate logic expressions) in both directions:

$$A \Leftrightarrow B: \alpha, \beta \tag{2}$$

where α and β, called *merit parameters,* express the *forward strength* and the *backward strength* of the m-implication, respectively.

An m-implication can be used for reasoning by tracing it in either direction. Tracing it forward (from the left to the right) means that if A is known to be true, then B can be asserted as true with the degree of belief α, if no other information relevant to B is known that affects this conclusion. Tracing an m-implication backward means that if B is known to be true, then A can be asserted as true with the degree of belief β, if no other information relevant to A is known that affects this conclusion. The m-implication reduces to a logical implication if α is 1 and β is unknown (in which case it is written as A $\Rightarrow$ B).

If any of the parameters α or β takes value 1 (which represents a complete belief), then the m-implication is *conclusive* (or *demonstrative*) in the direction for which the merit parameter equals 1; otherwise it is called *mutually-contingent* (or *m-contingent.*) In many situations, it is convenient to express an m-implication that has merit parameters (or only one) sufficiently high to merit their consideration without stating their precise values. For this purpose, we use symbols $\leftrightarrow$ (or $\rightarrow$), without listing α and β. Thus, an m-implication A $\Leftrightarrow$ B: α, β, in which α and β are unspecified but are above some "threshold of acceptability," is alternatively written A $\leftrightarrow$ B, or A $\rightarrow$ B. The concept of mutual implication was originally postulated in the theory of plausible reasoning (Collins & Michalski, 1989), which was developed by analyzing protocols recording examples of human reasoning.

Based on the above definition, one can say that abduction produces a plausible conclusion, if it traces backward a mutual implication in which β is sufficiently high. Thus, if

abduction is based on a standard implication (in which β is unknown), then it can be quite haphazard reasoning. Section 7 shows that a generalized form of mutual implication provides a formal basis for analogical inference.

The concept of an m-implication raises two basic problems: how merit parameters are determined, and how they are combined and propagated in reasoning through a network of m-implications. Regarding the first problem, the simplest interpretation of them is to assume that $\alpha = p(B \mid A)$ and $\beta = p(A \mid B)$. However, to make the concept of m-implication applicable for expressing many kinds of dependencies (including those occuring in human plausible reasoning), it is assumed that merit parameters do not have only one interpretation or representation. In a general view of m-implication, they can be precise values or only estimates of conditional probability, ranges of probabilities, degrees of dependency based on a contingency table (e.g., Goodman & Kruskal, 1979; Piatetsky-Shapiro, 1992), characterizations of the "strength" of dependency provided by an expert, or some other measures of dependency.

As to the second problem (how to combine merit parameters in reasoning with multiple m-implications), a comprehensive study of ideas and methods for the case of the probabilistic interpretation of merit parameters is presented by Pearl (1988). He uses "Bayesian networks" for updating and propagating beliefs based on a probabilistic model.

The fundamental difficulty in solving the second problem generally is that all logics of uncertainty, such as multiple-valued logic, probabilistic logic, fuzzy logic, etc., are not *truth-functional*, which means that there is no definite function for combining uncertainties. The reason is that the certainty of a conclusion from uncertain premises does not depend solely on the certainty (or probability) of the premises, but also on their meaning and their semantic interrelationship. The ultimate solution of this open problem will require methods that take into consideration both merit parameters and the meaning of the sentences. The results of research on human plausible reasoning conducted by Collins and Michalski (1989) show that people derive a combined certainty of a conclusion from uncertain premises by taking into consideration structural (or semantic) relations among the premises, based on a hierarchical knowledge representation, and involve also other types of merit parameters, such as typicality, frequency, dominance, etc.

Conclusive inferences are those that involve tracing mutual implications in which one merit parameter equals 1 (assuming that the input statement is true and perfectly matches the premise). Tracing mutual implications in which both merit parameters are below 1 (or when the above assumptions are not true) produces contingent inferences. In natural language, the contingency of a conclusion is expressed by a qualitative or quantitative degree of "strength" or "truth" (e.g., "maybe," "probably," "likely," "with probability of .7," with a "degree of belief," "certainty," or "confidence," etc.), or by a contingent quantifier (e.g., "most," "frequently," "usually," "90% of . . .," etc.). Suppose, for example, that BK contains a statement "Most elements of X have property q" and input is "x is a member of X." Deriving the statement "x likely has the property q" is a contingent deduction.

Let us consider more examples of contingent inferences. Consider a statement: "Fire usually produces smoke." This statement can be viewed as a weak mutual implication (both merit parameters are below 1). If one sees fire, then by contingent deduction one may derive a conclusion that there may be smoke. If one observes smoke, then by contingent induction

("tracing backward" the m-implication), one may hypothesize that there may be fire. Since the m-implication represents domain knowledge, this form of induction is contingent abduction. It explains "fire" by hypothesizing "smoke." Notice that if the input is false (there is no smoke), the uncertainty degree (falsity) of the hypothesis depends on the forward strength of the mutual implication.

In the above example, reasoning in both directions of the m-implication is uncertain. This might suggest that there is no intrinsic difference between contingent deduction and contingent induction (in the case above, abduction). A usual way of distinguishing between the two types of inference is to check if the entailment $|=$ in a given instantiation of (1) represents as a causal ordering, i.e., if P can viewed as a cause and C as an effect. In such a situation, contingent deduction (or contingent prediction) derives a plausible consequent, C, of the causes represented by P. Abduction derives plausible causes, P, of the consequent C. Since we can say that "fire causes smoke," and not the converse, then the above rule allows us to make a qualitiative distinction between inferences that trace this implication in one or another direction. Contingent deduction (prediction) can thus be viewed as "tracing forward" and abduction as "tracing backward" contingent, causally ordered dependencies.

This distinction, however, is not generally sufficient. The problem is that there are mutual implications that do not represent causal dependencies. For example, consider the statement "Prices at Tiffany's tend to be high." This statement expresses a non-causal mutual implication:

$$\text{Purchased-at(item, Tiffany's)} \Leftrightarrow \text{Price(item, High)}: \alpha, \beta \qquad (3)$$

If one is told that an item, e.g., a crystal vase, was purchased at Tiffany's, then one may conclude, with confidence α, that the price of it was high (if no other information about the price of the vase is known). The conclusion is uncertain if $\alpha < 1$ (which reflects, e.g., the possibility of a sale). If one is told that the price of an item was high, then one might hypothesize, with confidence β (usually low) that perhaps the item was purchased at Tiffany's. The confidence β depends on our knowledge about how many expensive shops are in the area where the item was purchased. Both inferences above are uncertain (assuming α, $\beta < 1$), and there is no clear causal ordering underlying the m-implication. Which inference is then contingent deduction, and which is contingent induction (or abduction)?

The way we propose to resolve this problem is based on the observation that a conclusive deduction traces an implication in the "strong" direction (with the degree of strength 1), and abduction traces such an implication in a "weaker" direction. Generalizing this property to reasoning with weak mutual implications that are not causal dependencies, we propose the rule: *reasoning in the direction of the greater strength of an m-implication is deduction, and reasoning in the direction of the weaker strength is induction (abduction, if the m-implication represents a domain knowledge). If reasoning in both directions has the same strength, there is no distinction between deduction and induction*. For example, reasoning based on tracing logical equivalence is deduction or induction both ways.

Going back to our example with Tiffany's, one may observe that α is usually significantly higher than β (unless Tiffany's is the only expensive store in the area under consideration). Thus, the forward reasoning based on (3) can be viewed as contingent deduction, and the

backward reasoning as contingent abduction. The distinction between contingent deduction and contingent induction (or abduction) is in this case a matter of degree, and it depends on the assumptions underlying the m-implication used for inference.

Summarizing, conclusive deductive inference is strictly truth-preserving, and conclusive induction is strictly falsity-preserving. A conclusive deduction produces a provable (valid) consequent from a given premise in the context of BK. A conclusive induction produces a "valid" hypothesis, which logically entails the given consequent in the context of BK (though the hypothesis itself may be false). Contingent deduction is truth-preserving and contingent induction is falsity-preserving to the degree defined by the forward strength of mutual implications involved in reasoning, respectively. Contingent induction is truth-preserving to the degree defined by the backward strength of the mutual implication.

The intersection of deduction and induction, that is a truth- and falsity-preserving inference, represents an equivalence-based inference, or *reformulation* transmutation. Such an inference transforms given knowledge into logically equivalent knowledge. For example, if A is logically equivalent to A', then the rule A $\Rightarrow$ B can be transformed to A' $\Rightarrow$ B. Analogy can be viewed as an extension of equivalence-based inference, namely, as a "similarity-based" inference. For example, if A is similar to $\mathcal{A}$ in terms relevant to B (for an explanation, see section 7), then from A $\Rightarrow$ B one can plausibly derive $\mathcal{A} \Rightarrow$ B. Analogy occupies the central area in the diagram in figure 2 because deriving new knowledge by analogy can be viewed formally as a combination of induction and deduction. Details on this idea are in section 7.

4. Knowledge transmutations

As stated earlier, transmutations are generic patterns of knowledge change, and their instantiations can be viewed as operators in knowledge spaces. A transmutation may change some aspect of the input knowledge, derive new knowledge, or perform certain manipulations on knowledge that do not change its content. Formally, a transmutation can be viewed as a transformation that takes as arguments a set of sentences (S), a set of entities (E), and background knowledge (BK), and generates a new set of sentences (S'), and/or new set of entities (E'), and/or new background knowledge (BK'):

$$\text{T: S, E, BK } \rightarrow \text{ S', E', BK'} \tag{4}$$

Transmutations can be classified into two categories. In the first category are *knowledge-generation* transmutations that change informational content of the input knowledge. Such transmutations represent patterns of inference. For example, they may derive consequences from given knowledge, suggest new hypothetical knowledge, determine relationships among knowledge components, confirm or disconfirm given knowledge, perform mathematical operations on quantitative knowledge, organize knowledge into certain structures, etc. Knowledge-generation transmutations are performed on statements that have a truth status.

In the second category are *knowledge-manipulation* transmutations that view input knowledge as data or objects to be manipulated. They can be performed on statements (well-formed logical expressions) or on terms (sets). They include inserting (deleting) knowledge

components into (from) give knowledge structures, physically transmitting or copying knowledge to/from other knowledge bases, or ordering knowledge components according to some syntactic criteria.

Transmutations are typically bidirectional operations, that is, they can be grouped into pairs of opposite operators, except for derivations that span a range of transmutations; the endpoints of this range are opposites. Below is a summary of knowledge transmutations that have been identified in the theory as frequently occuring in human reasoning or machine learning algorithms. This is not an exhaustive list; further research will likely identify other transmutations. The first eight groups represent knowledge-generation transmutations, and the remaining ones represent knowledge-manipulation transmutations. It should be noted that these transmutations can be applied to all kinds of knowledge expressed in a declarative way—specific facts, general statements, metaknowledge, control knowledge, or goals.

1. *Generalization/specialization*. The generalization transmutation extends the reference set of the input, that is, it generates a description that characterizes a larger reference set than the input. Typically, the underlying inference is inductive, that is, the extended set is inductively hypothesized. Generalization can also be deductive, when the more general description is deductively derived from the more specific one using background knowledge. It can also be analogical, when the more general description is hypothesized through analogy to a generalization performed on a similar reference set. The opposite transmutation is *specialization*, which narrows the reference set. Specialization usually employs deductive inference, but there can also be an inductive or analogical specialization.

2. *Abstraction/concretion*. Abstraction reduces the amount of detail in a description of the given reference set. It may change the description language to one that uses more abstract concepts or operators, which ignore details irrelevant to the reasoner's goal. The underlying inference is typically deduction. An opposite transmutation is *concretion*, which generates additional details about the reference set.

3. *Similization/dissimilization*. *Similization* derives new knowledge about a reference set on the basis of the similarity between this set and another reference set about which the learner has more knowledge. The similization is based on analogical inference. The opposite operation is *dissimilization*, which derives new knowledge on the basis of the lack of similarity between the compared reference sets. These transmutations are based on the patterns of inference presented 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 is to hypothesize that Holland may also grow roses. An underlying background knowledge here is that there exists a dependency between the climate of a place and the type of plants growing in that location. A dissimilization transmutation is to infer that bougainvilleas probably do not grow in Holland, because Holland has very different climate from the Caribbean Islands where they are very popular. These transmutations are based on analogical inference, which can be characterized as a combination of inductive and deductive inference (see section 7).

4. *Association/disassociation*. The association transmutation determines a dependency between given entities or descriptions based on the observed facts and/or background knowledge. The dependency may be logical, causal, statistical, temporal, etc. Associating a concept instance with a concept name is an example of an association transmutation.

The opposite transmutation is *disassociation*, which asserts a lack of dependency. For example, determining that a given instance is not an example of some concept is a disassociation transmutation.

5. *Selection/generation*. The selection transmutation selects a subset from a set of entities (e.g., a set of knowledge components) that satisfies some criteria. For example, choosing a subset of relevant attributes from a set of candidates, or determining the most plausible hypothesis among a set of candidate hypotheses, is a selection transmutation. The opposite transmutation is *generation*, which generates entities of a given type. For example, generating an attribute to characterize a given entity, or creating an alternative hypothesis to the one already generated, is a form of generation transmutation.

6. *Agglomeration/decomposition*. The agglomeration transmutation groups entities into larger units according to some goal criterion. If it also hypotheses that the larger units represent general patterns in data, then it is called *clustering*. The grouping can be done according to a variety of principles, e.g., to maximize some mathematical notion of similarity, as in conventional clustering, or to maximize "conceptual cohesiveness," as in conceptual clustering (e.g., Stepp & Michalski, 1983). The opposite transmutation is a *decomposition*, which splits a group (or a structure) of entities into subgroups, according to some goal criterion.

7. *Characterization/discrimination*. A *characterization* transmutation determines a *characteristic* description of a given set of entities, which differentiates these entities from any other entities. A simple form of such a description is a list (or a conjunction) of all properties shared by the entities of the given set. The opposite transmutation is *discrimination*, which determines a description that discriminates the given set of entities from another set of entities (Michalski, 1983).

8. *Derivations: Reformulation/intermediate transmutations/randomization*. *Derivations* are transmutations that derive one piece of knowledge from another piece of knowledge (based on some dependency between them), but do not fall into the special categories described above. Because the dependency between knowledge components can range from logical equivalence to random relationship, derivations can be classified on the basis of the strength of dependency into a wide range of forms. The extreme points of this range are *reformulation* and *randomization*. Reformulation transforms a segment of knowledge (a set of conceptually related sentences) into a logically equivalent segment of knowledge. For example, mapping a geometrical object represented in a right-angled coordinate system into a radial coordinate system is a reformulation. In contrast, randomization transforms one knowledge segment to another one by making random changes. For example, the *mutation* operation in a genetic algorithm represents a randomization. Deductive derivation, abductive explanation, and prediction can be viewed as intermediate derivations. Mathematical or logical transformations of knowledge also represent forms of derivations. A weak intermediate derivation is the *crossover* operator used in genetic algorithms, which derives new knowledge by exchanging two segments of related knowledge components.

9. *Insertion/deletion*. The insertion transmutation inserts a given knowledge component (e.g., a component generated by some other transmutation) into a given knowledge structure. The opposite transmutation is *deletion*, which removes some knowledge component from a given structure. An example of deletion is forgetting.

18

10. *Replication/destruction*. Replication reproduces a knowledge structure residing in some knowledge base in another knowledge base. Replication is used, e.g., in *rote learning*. There is no change of the contents of the knowledge structure. The opposite transmutation is *destruction*, which removes a knowledge structure from a given knowledge base. The difference between destruction and deletion is that destruction removes a copy of a knowledge structure that resides in some knowledge base, while deletion removes a component of a knowledge structure residing in the given knowledge base.

11. *Sorting/unsorting*. The sorting transmutation changes the organization of knowledge according to some criterion. For example, ordering decision rules in a rule base from the shortest (having the smallest number of conditions) to the longest is a sorting transmutation. An opposite operation is *unsorting*, which is returning to the previous organization.

Figure 3 provides a summary of the above transmutations together with the underlying types of inference. It is postulated that, depending on the amount of available background knowledge and the way that the input and the background knowledge are employed, any knowledge-generation transmutation can be, in principle, accomplished by any type of inference, i.e., deduction, induction, or analogy. Figure 3 illustrates this by linking these transmutations with all three forms of inference. Exceptions to this rule are similization and dissimilization, which are based on analogy (analogy is viewed as deduction and induction combined). A vertical link between lines stemming from the nodes denoting similarity/dissimilarity transmutations signifies that these transmutations combine deduction with induction (for an explanation, see section 7).

Transmutations that employ induction, analogy, or contingent deduction increase the amount of *intrinsic* knowledge in the system. By intrinsic knowledge we mean knowledge that cannot be conclusively deduced from other knowledge in the system. Learning that changes the amount of the system's intrinsic knowledge is called *synthetic* (it has been also called "learning at the knowledge level" (Newell, 1981; Dietterich, 1986)). Transmutations that employ only conclusive deduction increase the amount of *derived* knowledge in the system. Such knowledge is a logical consequence of what the learner already knows. Learning that changes only the amount of derived knowledge in the system is called *analytic* (Michalski & Kodratoff, 1990).

In actual use, different transmutations are typically performed using only one type of inference. For example, generalization and agglomeration are typically done through induction, and specialization and abstraction through deduction.

Generalization can also be deductive (as, e.g., in explanation-based generalization), or analogical (when a more general description is derived by an analogy to some other generalization transformation). Specialization is typically deductive, but it can also be inductive or analogical.

Thus, the theory views transmutations as different types of change in knowledge, and inferences as different ways in which these changes can be accomplished. This is a radical departure from the traditional view of these issues. The traditional view blurs the proposed distinctions; for example, it typically equates generalization with induction, and specialization with deduction.

The proposed view stems from our efforts to provide an explanation of different operations on knowledge observed in people's reasoning and to relate this explanation to formal types of inference in a consistent way. Experiments performed with human subjects have

Knowledge Generation Transmutations

Inference Type   *Transmutation*

DEDUCTION

ANALOGY

INDUCTION

Generalization
Specialization

Abstraction
Concretion

Association
Disassociation

Similization
Dissimilization

Selection
Generation

Agglomeration
Decomposition

Characterization
Discrimination

Derivations

EQUIVALENCE   Reformulation
DEDUCTION
ANALOGY   Intermediate
INDUCTION   derivations
NO INFERENCE   Randomization

Knowledge Manipulation Transmutations

DEDUCTION

Insertion
Deletion

Replication
Destruction

Sorting
Unsorting

Figure 3. A summary of transmutations and the underlying types of inference.

shown that the proposed ideas agree well with typical intuitions people have about different types of transmutations. Further research is needed to formalize these ideas precisely.

The theory views learning as a sequence of goal-oriented knowledge transmutations. For example, a generation transmutation may generate a set of attributes to characterize given entities. Another generation transmutation may create examples expressed in terms of these attributes. A general description of these examples is created by a generalization transmutation. By repeating different variants of a generalization transmutation, a set of alternative general descriptions of these examples can be determined. A selection transmutation would

choose the "best" candidate description according to a criterion specified by the given learning goal. If a new example contradicts the description, a specialization transmutation would produce a new description that takes care of the inconsistency. The description obtained may be added to the knowledge base by an insertion transmutation. A replication transmutation may then copy this description into another knowledge base.

The next section analyzes those transmutations that employ inductive inference, which are fundamental for the creation of instrinsically new knowledge.

5. Admissible induction and inductive transmutations

As described in section 3, induction produces a premise (a fact, a rule, a set of sentences, a theory, etc.) that together with BK entails a given consequent. In the case of inductive generalization, the entailment is guaranteed by the set–superset relationship between the input and output. In the case of abduction, the entailment is due to the implicative domain-dependent relationship that is being traced backward. For any given consequent C and non-trivial BK, one can generate a potentially infinite number of hypotheses that together with BK entail C, but only a few of them may be of any interest. One is usually interested only in "simple," "justifiable," and/or "plausible" hypotheses. Therefore, to limit the choices, an *admissible induction* is defined by adding constraints to the definition of induction in section 3.

Definition. Given a consequent C and background knowledge BK, an admissible induction hypothesizes a premise P, consistent with BK, such that

$$P \cup BK \models C \tag{5}$$

and P satisfies the *hypothesis selection criterion*.

The selection criterion specifies how to choose a hypothesis among all candidates satisfying (5), and may be a combination of several elementary criteria. In different contexts, or for different forms of induction, the selection criterion has been called a *preference criterion* (Popper, 1972; Michalski, 1983), a *bias* (Utgoff, 1986; Grosof & Russell, 1989) and a *comparator* (Poole 1989). Ideally, the selection criterion should reflect the properties of a hypothesis that are desirable from the viewpoint of the learner's goals, rather than be influenced by the properties of a particular induction method.

In some learning programs, a part of the selection criterion may be hidden in the description language used (a "description language bias"). For example, an inductive program may use a description language that is limited to only conjunctive statements involving attributes from a predefined set. Therefore any hypothesis generated will have to be in this form. Often, the selection criterion relates to the knowledge representation used. For example, when learning decision trees, the selection criterion may seek trees with a small number of nodes (by choosing at each step the locally most informative attribute); when learning DNF descriptions, it may seek the shortest DNF descriptions.

There are three generally desirable characteristics of a hypothesis: *plausibility, utility,* and *generality.* The plausibility expresses a desire to find a "true" hypothesis. Because the problem is logically underconstrained, the "truth" of a hypothesis cannot be guaranteed in principle. To satisfy equation (5) a hypothesis has to be *complete* and *consistent* with regard to the input facts (Michalski, 1983). Experiments have shown, however, that in situations where the input contains errors or noise, an inconsistent and/or incomplete hypothesis (with regard to the input) will often lead to a better overall predictive performance than a complete and consistent one (e.g., Bergadano et al., 1992). The utility criterion requires a hypothesis to be simple to express and easy to apply to the expected set of problems. The generality criterion seeks a hypothesis that can predict a large range of new cases.

The view of induction described above is more general than the one often expressed in machine learning literature. It is also consistent with many long-standing thoughts on this subject going back to Aristotle (e.g., Adler & Gorman, 1987; Aristotle, 1987). Aristotle, and many subsequent thinkers, e.g., Bacon (1620), Whewell (1857), Cohen (1970), Popper (1972), and others, viewed induction as a fundamental inference for all processes of creating new knowledge. They did not limit it—as is sometimes done—to only inductive empirical generalization.

Induction underlies a number of knowledge transmutations, such as *inductive generalization, inductive specialization, abductive derivation,* and *concretion* (not a complete list). The first two were illustrated in section 3. The other two are illustrated below.

In principle, all knowledge-generation transmutations described in section 4 can have a form involving inductive inference. Inductive generalization is central to many learning processes. As mentioned earlier, it extends the reference set described, e.g., it extends a set of training concept examples to the set of all concept instances. Such an operation can be done with or without changing the description space.

If generalization is done without change of description space (i.e., without modifying descriptors or introducing new descriptors), then generalization is called *empirical* or *selective.* If generalization is done by changing the description space into a more problem-oriented one (e.g., some initially given descriptors are modified or removed, and new descriptors added), then generalization is called *constructive* (Michalski, 1983; Wnek & Michalski, 1991a). (See figure 4 for an example.)

Another form of inductive transmutation is inductive specialization, which decreases the reference set described in the input. Typically, a generalization is inductive and a specialization is deductive. However, depending on the way the input is transformed, a generalization transmutation may also be deductive, and a specialization transmutation may also be inductive (as shown in figure 4). Both generalization and specialization transmutations may also be analogical when the description of an extended or contracted reference set is hypothesized by analogy. Abductive derivation derives an explanation of properties of a given reference set. Concretion is the opposite of abstraction (see the next section). It hypothesizes more specific information about a given reference set.

Figure 4 presents examples of the above transmutations. In the examples, to indicate that some m-implications are not conclusive (not logical implications) but are sufficiently strong to warrant consideration (characterize a tendency), the symbol $\leftrightarrow$ is used. The parameter α stands here for "maybe." Given an input and BK, there are usually many possible inductive transmutations of them; here we list one of each type—the one that is normally perceived as the most "natural." The first, third, and fourth example in figure 4 represent conclusive

• *Empirical   inductive   generalization*
(Background knowledge-limited)

Input:	Pntng(GF, Dwski) ⇒ Btfl(GF)	(Dawski's paintings, "A girl's face" and
	Pntng(LC, Dwski) ⇒ Btfl(LC)	"Lvov's cathedral," are beautiful)
BK:	∀x,P(x) ⇒ P(ɑ)	(The universal specialization rule; short form)
Output:	∀x,Pntng(x, Dwski) ⇒ Btfl(x): α	(*Maybe* all Dawski's paintings are beautiful)

• *Constructive   inductive   generalization   (generalization + deductive derivation)*
(Background knowledge-intensive)

Input:	Pntng(GF, Dwski) ⇒ Btfl(GF)	(Dawski's paintings, "A girl's face" and
	Pntng(LC, Dwski) ⇒ Btfl(LC)	"Lvov's cathedral," are beautiful)
BK:	∀x,y, Pntng(x,y)&Btfl(x) <—> Exp(x)	(Btfl pntngs tend to be expensive & opposite)
Output:	∀x,Pntng(x, Dwski) ⇒ Exp(x): α	(*Maybe* all Dawski's paintings are expensive)

• *Inductive   specialization*

Input:	Lives(John, Virginia)	(John lives in Virginia)
BK:	Fairfax ⊂ Virginia	(Fairfax is a "subset" of Virginia)
	∀x,y,z, y ⊆ z&Lives(x,y)⇒Lives(x,z)	(Living in x implies living in superset of x)
Output:	Lives(John, Fairfax): α	(*Maybe* John lives in Fairfax)

• *Concretion*

Input:	Going-to(John, New York)	(John is going to New York)
BK:	Likes(John, driving)	(John likes driving)
	∀x,y, Driving(x,y) ⇒ Going-to(x,y)	("Driving to" is a special case of "going to.")
	∀x,y, Likes(x,driving)⇒Driving(x,y)	(Liking to drive m-implies driving to places)
Output:	Driving(John, New York): α	(*Maybe* John is driving to New York)

• *Abductive   derivation*

Input:	In(House, Smoke)	(There is smoke in the house)
BK:	In(x, Smoke) <--> In(x, Fire)	(Smoke usually indicates fire &conversely)
Output:	In(House, Fire): α	(*Maybe* there is fire in the house)

• *Constructive   inductive   generalization   (generalization   plus   abduction)*

Input:	In(John'sApt, Smoke)	(Smoke is in John's apartment)
BK:	In(x, Smoke) <--> In(x, Fire)	(Smoke usually indicates fire &conversely)
	John'sApt ⊂ GKBld	(John's apt. is in the Golden Key building)
Output:	In(GKBld, Fire): α	(*Maybe* there is fire in the Golden Key bld)

Figure 4. Examples of inductive transmutations.

induction (in which the hypothesis with BK strongly implies the input); the second and the last two examples represent contingent induction. The second example would be a conclusive induction, if the rule in BK were

$$\forall x, y \ (Pntng(x, y) \ \& \ Btfl(x) \ \Leftrightarrow \ Exp(x): \alpha \ = \ usually, \ \beta \ = \ 1$$

("All beautiful paintings are usually expensive, but expensive paintings are always beautiful"), which does not reflect the facts in real life. In the examples, the subset symbol " ⊂ " is used under the assumption that cities, states, apartments, and buildings can be viewed as sets of space parcels.

6. Generalization vs. abstraction

This section analyzes two fundamental knowledge-generation transmutations, namely, generalization and abstraction, and their opposites, specialization and concretion, respectively. Generalization and abstraction are sometimes confused with each other; therefore we provide an analysis of the differences between them. We start with generalization and specialization.

6.1. Generalization and specialization

As stated earlier, our view of generalization is that it is a knowledge transmutation that extends the reference set of a given description. Depending on the background knowledge and the way it is used, generalization can be inductive, deductive, or analogical. Such a view of generalization is more general than the one traditionally expressed in the machine learning literature, which recognizes only one form of generalization, namely, inductive generalization.

Based on experiments with human subjects, we claim that the view presented here more adequately captures the common intuitions and the natural language usage of the term "generalization." To express the proposed view rigorously, let us provide a more precise definition of the reference set.

Suppose S is a set of statements in predicate logic calculus. Suppose further that an argument of one or more predicates in statements in S stands for a set of entities, and that S is interpreted as a description of this set.

Under this interpretation, the set of entities described by S is called the *reference set* for S. If the reference set is replaced by a set-valued variable, then the resulting expression is called a *descriptive schema*, and denoted D[R], where R stands for the reference set. For example, suppose given is a statement:

S: In(John'sApt, Smoke)   (Smoke is in John's apartment.)

This statement can be interpreted as a description of the set {John'sApt}. Thus we have

D[R]:   In (R, Smoke)
R: John'sApt.

For a given statement, if one ignores the context in which it is used, there could be more than one reference set and the corresponding descriptive schema. For example, consider the statement: "George Mason lived at Gunston Hall." It can be interpreted as a description of "George Mason" (a singleton set), which specifies the place where he lived. It can also be interpreted as a description of "Gunston Hall," which specifies a property of this place, namely, that George Mason lived there. The appropriate interpretation of a statement depends on the context in which it is used. For example, in the context of a discussion about George Mason, the first interpretation would apply; but if Gunston Hall is the object of a discussion, the second interpretation would apply.

Suppose two sets of statements, S1 and S2, are given that can be interpreted as having reference sets R1 and R2 and descriptive schemes D1 and D2, respectively, i.e., S1 = D1[R1] and S2 = D2[R2]

Definition. The statement set S2 is more *general* than statement set S1 if and only if

$R2 \supset R1$ and

$$D2[R2] \cup BK \Rightarrow D1[R1] \qquad\qquad (5')$$

$$D1[R1] \cup BK \Rightarrow D2[R2] \qquad\qquad (5'')$$

If condition (5′) holds, S2 is an *inductive* generalization of S1; if condition (5″) holds, S2 is a *deductive* generalization of S1. By requiring that the compared statements satisfy an implicative relation in the context of given background knowledge, the definition allows one to compare the generality of statements that use different descriptive concepts or languages.

Let us illustrate the above definition using examples from section 5.

Example 1. (Empirical inductive generalization)

S1:	Pntng(GF, Dwski) & Btfl(GF)	(Dawski's painting, "A girl's face," is beautiful.)
D1[R1]:	Pntng(R1, Dwski) & Btfl(R1)	
R1:	GF	(GF is a singleton, {Girl's face})

S2:	∀x, Pntng(x, Dwski) ⇒ Btfl(x)	(All Dawski's paintings are beautiful.)
	Alternatively: Btfl(All_DPs)	(All_DPs denotes the set of all Dawski's paintings.)
D2[R2]:	Btfl(R2)	(Paintings from the set R2 are beautiful.)
R2:	All_DPs	(All Dawski's paintings.)
BK:	GF ⊂ All_DPs	

The interpretation of the predicate Btfl(R) is that the property Btfl applies to every element of the set R. Since R2 ⊃ R1, and D2[R2] ⇒ D1[R1], then S2 is more general than S1.

Example 2. (Deductive generalization)

S1:	Lives(John, Fairfax)	(John lives in Fairfax.)
D1[R1]:	Lives(John, R1)	
R1:	Fairfax	

S2:	Lives(John, Virginia)	(John lives in Virginia.)
D2[R2]:	Lives(John, R2)	
R2:	Virginia	
BK:	Fairfax ⊂ Virginia	

S2 is more general than S1 because R2 ⊃ R1, and D1[R1] ∪ BK ⇒ D2[R2].

In human reasoning, generalization is frequently combined with other types of transmutations producing various composite transmutations. Here is an example of such a composite transmutation.

Example 3. (Inductive generalization and abduction)

S1:	In(John'sApt, Smoke)	(There is smoke in John's apartment.)
D1[R1]:	In (R1, Smoke)	
R1:	John'sApt	
BK:	In(x, Smoke) ↔ In(x, Fire)	
	John'sApt ⊂ GKBldng	

S2:	In(GKBldng, Fire)	(There is fire in Golden Key Building.)
D2[R2]:	In(R2, Fire)	
R2:	GKBldng	

In this example, a generalization transmutation of the input produces a statement "Smoke is in the Golden Key building." An abductive derivation (also called abductive explanation) applied to the same input would produce a statement "There is fire in John's apartment." By applying abductive derivation to the outptut from generalization, one obtains a statement "There is fire in Golden Key building."

The above definition defined a generalization relation only between two sets of statements. Let us now extend this definition to the case in which the input may be a collection of sets of statements. Such a case occurs in learning rules that generalize a set of examples (each example may be described by one or more statements.).

Definition. The statement set, S, is a *generalization of a collection of statement sets* $\{S_i\}$, i = 1, 2, .. k, if and only if S is more general than each S_i.

Summarizing, a generalization transmutation is a mapping from one description (input) to another description (output) that extends the reference set of the input. Depending on the background knowledge, such an operation can be inductive or deductive.

A transmutation opposite to generalization is *specialization*, which reduces the reference set of a given set of statements. A typical form of specialization is deductive, but there can also be an inductive specialization. For example, a reverse of the inductive specialization in figure 4 is a deductive generalization:

Input:	Lives(John, Fairfax)	(John lives in Fairfax.)
BK:	Fairfax $\subset$ Virginia	(Fairfax is a "subset" of Virginia.)
	$\forall$x, y, z, y $\subset$ z & Lives(x,y) $\Rightarrow$ Lives(x, z)	(Living in y implies living in a superset of y.)

Output:	Lives(John, Virginia)	(John lives in Virginia.)

In the above example, Fairfax and Virginia are interpreted as reference sets (sets of land parcels). The Input states that a property of Fairfax is that "John lives there." The property "Living in a set of land parcels" means occupying some elements of this set. This is *an existential property* of a set, which is defined as a property that applies only to some unspecified elements of the set. If a set has such property, then so do its supersets. This is why the above inference is deductive.

In contrast, a *universal property* of a set applies to all elements of the set. If a set has such a property, so does its every subset, but not every superset. Thus, if in the above example a "universal property" was used, e.g., "Soil(good, Fairfax)," a generalization transmutation to "Soil(good, Virginia)" would be inductive.

Generalization/specialization transmutations are related to another type of transmutations, namely, abstraction/concretion. Transmutations of these two types often co-occur in commonsense reasoning; therefore they are easy to confuse with each other. By changing the interpretation of an input statement (i.e., by differently assigning the reference set and descriptive schema in a statement), deductive generalization can often be *reinterpreted* as abstraction. Abstraction and concretion transmutations are analyzed below.

6.2. Abstraction and concretion

Abstraction reduces the amount of information conveyed by a description of a set of entities (the reference set). The purpose of abstraction is to reduce the amount of information about the reference set in such a way that the information relevant to the learner's goal is preserved and the irrelevant information is discarded. For example, abstraction may transfer a description from one language to another language in which the properties of the reference set relevant to the reasoner's goal are preserved, but other properties are not. An opposite operation to abstraction is *concretion*, which generates additional details about a given reference set.

A simple form of abstraction is to replace a specific attribute value (e.g., the length in centimeters) in the description of an entity by a less specific value (e.g., the length stated in linguistic terms, such as short, medium, or long). A complex abstraction would be, e.g., to take a description of a computer in terms of electronic circuits and connections, and, based on background knowledge, change it into a description in terms of the functions of major components. Typically, abstraction is a form of deductive transmutation, because it preserves the important information in the input and does not hypothesize any information (this latter possibility may occur when the input or BK contain uncertain information).

Let us express this view of abstraction more formally. An early formal definition of abstraction was proposed by Plaisted (1981), who considered it as a mapping between languages that preserves instances and negation. A related but somewhat different view was presented by Giordana, Saitta, and Roverso (1991), who consider abstraction as a mapping between abstract models. In the view presented here, abstraction is a mapping between descriptions based on background knowledge. Specifically, it is a knowledge transmutation that creates a less detailed description from a more detailed description of the same set of entities (the reference set), using the same or other terms. Unlike generalization, it does not change the reference set, but only changes the description of it.

Suppose given are two sets of expressions, S1 and S2, that can be interpreted as having descriptive schemes D1 and D2, respectively, and the same reference set, R.

Definition. S2 is more *abstract* than S1 *in the context* of background knowledge BK, and with *the degree of strength* α, if and only if

$$D1[R] \cup BK \Rightarrow D2[R]: \alpha, \text{ where } \alpha \geq Th \tag{6}$$

and there is a homomorphic mapping between the set of properties specified in D1, and the set of properties specified in D2. The threshold Th denotes a limit of acceptability of transformation as abstraction.

The last condition is needed to exclude arbitrary deductive derivations. The most common form of abstraction is when (6) is a standard (conclusive) implication ($\alpha = 1$). In this case, the set of strong inferences (deductive closure) that can be derived from the output (abstract) description and BK is a proper subset of strong inferences that can be derived from the input description and BK. This case can be called a *strong* abstraction, in contrast to *weak* abstraction, which occurs when $\alpha < 1$.

An example of weak absraction is when a picture of a table seen from one side (without seeing all legs) is transformed to a sketch of this table from a somewhat different side, showing four legs. When inference goals are defined, a "good" abstraction should preserve the inferences that are important to the goals and ignore those that are not. Comparing (5) and (6), one can see that an abstraction transmutation can be a part of an inductive generalization transmutation. For that reason, these two transmutations are sometimes confused with each other.

6.3. An illustration of the difference between abstraction and generalization

Let us illustrate the difference between abstraction and generalization by a simple example. Consider a statement $d(\{r_i\}, v)$, saying that descriptor d takes value v for entities from the set $\{r_i\}$. Thus, the reference set of this statement is $R = \{r_i\}$, $i = 1, 2, \ldots$, and a descriptive schema is $D[R] = d(R, v)$. Let us write the above statement in the form that illustrates syntactically the difference between the reference set and a descriptor value:

$$d(R) = v \tag{7}$$

Changing (7) to $d(R) = v'$, where v' represents a more general concept, e.g., a parent node in a generalization hierarchy of values of the attribute d, is an abstraction transmutation. Changing (7) to a statement $d(R') = v$, in which R' is a superset of R, is a generalization operation.

For example, transferring the statement "color(my-pencil) = light-blue" into "color(my-pencil) = blue" is an abstraction operation. To see this, notice that {[color(my-pencil) = light-blue]} & (light-blue $\subset$ blue) $\Rightarrow$ [color(my-pencil) = blue]. Transforming the original statement into "color(all-my-pencils) = light-blue" is a generalization operation. Finally, transferring the original statement into "color(all-my-pencils) = blue" is both generalization and abstraction. In other words, associating the same property with a larger set is a generalization; associating less information with the same set is an abstraction operation. Combining the two is a composite transmutation.

An opposite transmutation to abstraction is concretion, which increases the amount of information that is conveyed by a statement about the given set of entities (reference set).

The two pairs of mutually opposite transmutations, i.e., {generalization, specialization} and {abstraction, concretion}, differ by the aspects of knowledge they change. If a transmutation changes the size of the reference set of a description, then it is *generalization* or *specialization*. If a transmutation changes the amount of information (detail) conveyed by a description of a reference set, then it is *abstraction* or *concretion*. In other words, generalization (specialization) transforms descriptions along the set-superset (set-subset) direction, and is typically falsity-preserving (truth-preserving). In contrast, abstraction (concretion) transforms descriptions along the more-to-less-detail (less-to-more-detail) direction, and is typically truth-preserving (falsity-preserving). Generalization often uses the same description space (or language) for input and output statements, whereas abstraction often involves a change in the description space (or language).

7. Similization vs. dissimilization

The similization transmutation uses analogical inference to derive new knowledge. A dissimilization transmutation derives new knowledge based on a lack of analogy. As mentioned in section 2, analogical reasoning can be considered as a combination of inductive and deductive inference. Before we demonstrate this claim, let us observe that an important part of our knowledge is dependencies among various entities in the world. These dependencies can be of different strengths or types, such as functional, monotonic, correlational, general trend, etc. For example, we know that the dimensions of a rectangle exactly determine its area (this is a unidirectional functional dependency), that smoking causes lung cancer (this is a weakly causal dependency), or that improving education of citizens is good for the country (this is an unquantified belief).

Such dependencies are often bidirectional, but the "strength" of the dependency in different directions may vary considerably. For example, from the fact that Martha is a heavy smoker one may develop an expectation that she will likely get lung cancer later in her life; from learning that Betty has lung cancer, one may hypothesize that perhaps she was a smoker. The "strength" of these conclusions, however, may be quite different. Betty may have lung cancer for other reasons, or perhaps she was only married to a smoker. The dependencies can be known at different levels of specificity. In the past, the dependency between smoking and lung cancer was only a general hypothesis; now we have a much more precise knowledge of this dependency.

Section 3 introduced the notion of mutual implication (equation (2)) to express a wide class of such relationships. In order to describe a similization transmutation, we will extend the notion of mutual implication into a more general *mutual dependency*. As defined earlier, mutual implication expresses a relationship between two predicate logic statements (well-formed formulas; closed predicate logic sentences with no free variables). A mutual dependency expresses a relationship between two *sentences* that are both either predicate logic statements or term expressions (open predicate logic sentences, in which some of the arguments are free variables).

To state that there is a mutual dependency (*m-dependency*) between two sentences S1 and S2, we write

$$S1 \Leftrightarrow S2: \alpha, \beta \tag{8}$$

where merit parameters α and β represent an overall *forward strength* and *backward strength* of the dependency, respectively. α and β represent the average certainty with which a value of S1 determines a value of S2, and conversely.

If S1 and S2 are statements (well-formed formulas), then m-dependency is an m-implication. If S1 and S2 are term expressions, then mutual dependency expresses a relationship between functions (since term expressions can be interpreted as functions). If terms expressions in a mutual dependency are discrete functions, then mutual dependency is logically equivalent to a set of mutual implications. 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.

The concept of m-dependency allows us to describe similization and dissimilization trans- mutations. These transmutations involve determining a *similarity* or *dissimilarity* between entities, and then given some knowledge about one entity, hypothesizing some new knowl- edge about another entity. The concept of similarity is sometimes misunderstood and viewed as an objective, context-independent property of objects. In reality, the similarity between any two entities is highly context dependent. Any two entities (objects, sets of objects, be- haviors, etc.) can be viewed as boundlessly similar or boundlessly dissimilar, depending on what descriptors are used to characterize them, or, in other words, what properties are used to compare the entities. Therefore to talk meaningfully about a similarity between entities, one needs to indicate, explicitly or implicitly, the relevant descriptors. To capture this idea, we employ the concept of the *similarity in the context* introduced by Collins and Michalski (1989). To say that entities *El E2* are similar in context CTX (which defines relevant descriptors D), we write

$$\text{E1 SIM E2 in CTX(D)} \tag{9}$$

This statement says that values of the descriptors from the set D for the entity El and for the entity E2 differ no more than by some assumed tolerance threshold. For numerical descriptors, the theshold "Th" can be expressed as a percentage, relative to the larger value. For example, if Th = 10%, the values of the descriptor cannot differ more than 10%, rela- tive to the larger value. Descriptors in D can be attributes, relations, functions, or any transformations applicable to the entities under consideration. The threshold expresses the required degree of similarity for triggering the inference.

The similization transmutation is a form of analogical inference, and is defined by the following schema:

Input: E1 $\Rightarrow$ A
BK: E1 SIM E2 in CTX(D)
           D $\Rightarrow$ A: α > RT
_____
Output: E2 $\Rightarrow$ A   (10)

where α > RT states that the strength of the forward term dependency D $\Rightarrow$ A should be above a *relevance threshold*, RT, in order to trigger the inference. RT is a control parameter for the inference.

Given that entity El has property A, and knowing that there is a similarity between El and E2 in terms of descriptors defined by D, the rule hypothesizes that entity E2 may also have property A. This inference is allowed, however, only if there is a dependency between the property A and descriptors defined by D. The reason for the latter condition can be illustrated by the following example. Suppose we know that some person who is handsome and has received their Ph.D. from MIT. It would not be reasonable to hypothesize that another person who we find handsome also received their Ph.D. from MIT. The reason is that we do not expect any dependency between looks of a person and the university from which that person gets a Ph.D. degree.

A dissimilization transmutation draws an inference from the knowledge that two entities are very different in the context of some descriptors. A dissimilization transmutation follows the schema

Input: E1 ⇒ A
BK: E1 DIS E2 in CTX(D)
            D ⇒ A: α > RT
            _____

Output: E2 ⇒ ~A   (11)

where DIS denotes a relation of dissimilarity, and other parameters are as in (10).

Given that some entity E1 has property A, and knowing that entities E1 and E2 are very different in terms of descriptors that are in mutual dependency relation to A, the transmutation hypothesizes that maybe E2 does not have the property A.

The following simple example illustrates dissimilarity transmutation. Suppose we are told that apples grow in Poland. Knowing that apples are different from oranges in a number of ways, including the climate they require in order to grow, and that the climate of the area is m-dependent on the type of fruit grown there, one may hypothesize that perhaps oranges do not grow in Poland. Other examples of dissimilization are discussed by Collins and Michalski (1989).

Let us now illustrate the similization transmutation by a real-world example, and then demonstrate that it can be viewed as a combination of inductive and deductive inference. To argue for a national, ultra-speed electronic communication network for linking industrial, governmental, and academic organizations in the U.S., its advocates used an analogy that "Building this network is an information equivalent of building national highways in the '50s and '60s," There is little physical similarity between building highways and electronic networks, but there is an end-effect similarity in that they both improve communication. Since building highways helped the country, and thus was a good decision, then by analogy, building the national network will help the country, and is also a good decision to make. Using the schema (10), we have:

Input: Decision(Bld, NH) SIM Decision(Bld, NN) in CTX (FutCom)
BK: Decision(Bld, NH) ⇒ Effect-on(U.S., good)
            FutCom (US, x) ⇒ Effect-on(US, x): α > RT
            _____

Output: Decision(Bld, NN) ⇒ Effect-on(U.S., good)   (12)

where

NH—stands for National Highways
NN—stands for National Network
Decision(Bld, x)—a statement expressing the decision to build x
FutCom(area, state)—a descriptor expressing an evaluation of the future state of communication in the "area" that can take values: "will improve" or "will not improve"
Effect-on(US, x)—a descriptor stating that "the effect on the US is x."

We will now show how the general schema (10) can split into an inductive and deductive step.

An inductive step:

Input: E1 SIM E2 in CTX(D)
BK: D ⇔ A: α > RT
_____
Output: E1 SIM E2 in CTX(D, A)   (13)

From the similarity between two entities in terms of descriptor D, and a mutual dependency between the descriptor and some new term (descriptor) A, the schema hypothesizes a similarity between the entities in terms of D and A. The deductive step uses the hypothesized relationship of similarity to derive new knowledge.

A deductive step:

Input: E1 SIM E2 in CTX(D, A)
BK: E1 ⇒ A(α)
_____
Output: E2 ⇒ A(α')   (14)

where A(α) states that descriptor A takes value α, and α is equal or sufficiently close (for the learner's goals) to α'.

Using the above schemes, we can now describe the previous example of similization in terms of an inductive and deductive step.

An inductive step:

Input: Decision(Bld, NH) SIM Decision(Bld, NN) in CTX (FutCom)
BK: FutCom(US, x) ⇒ Effect-on(US, x): α > RT
_____
Output: Decision(Bld, NH) SIM Decision(Bld, NN) in CTX (FutCom, Effect-on)
                                                                                            (15)

A deductive step:

Input: Decision(Bld, NH) SIM Decision(Bld, NN) in CTX (FutCom, Effect-on)
BK: Decision(Bld, NH) ⇒ Effect-on(US, good)
_____
Output: Decision(Bld, NN) ⇒ Effect-on(US, good)   (16)

From the knowledge that the decision to build national highways is similar to the decision to build national networks from the viewpoint of communication in the U.S., and that communication in the U.S. has an effect on the U.S., the inductive step hypothesizes that there may be a similarity between two decisions also in terms of their effect on the U.S. The deductive step uses this similarity to derive a conclusion that building NN will have a good effect on the U.S., because building highways had a good effect. The validity of the deductive step rests on the strength of the hypothesis generated in the inductive step.

Summarizing, a similization (dissimilization) transmutation, given knowledge about one entity, hypothesizes knowledge about another entity, if the other entity is similar (dissimilar) to the first entity. The similarity is measured in terms of properties or transformations characterizing the relationship among the entities. The above rule holds, however, only if the hypothesized properties are sufficiently related, by an m-dependency, to the properties used for defining the similarity.

8. Multistrategy task-adaptive learning

The ideas presented in previous sections provide a conceptual framework for *multistrategy task-adaptive learning* (MTL), which aims at integrating a whole range of learning strategies. A general underlying idea of MTL is that a learning system should by itself determine the learning strategy, i.e., the types of inference to be employed and/or the representational paradigm that is most suitable for the given learning task (Michalski, 1990; Tecuci & Michalski 1991a, b). As introduced in the Inferential Learning Theory, a learning task is defined by three components: what information is provided to the learner (i.e., *input* to the learning process), what the learner already knows that is relevant to the input (i.e., *background knowledge* (BK)), and what the learner wants to learn (i.e., the goal or goals of learning). Given an input, an MTL system analyzes its relationship to BK and the learning goals and on that basis determines a learning strategy or a combination of them. If an impasse occurs, a new learning task is assumed, and the learning strategy is determined accordingly.

The above characterization of MTL covers a wide range of systems, from "loosely coupled" systems that use the same representational paradigm and employ different inferential strategies as separate modules, to "tightly coupled" (or "deeply integrated") systems in which individual strategies represent instantiations of one general knowledge and inference mechanism, to *multirepresentational* multistrategy systems that can synergistically combine and adapt both the knowledge representation and inferential strategies to the learning task.

A general schema for Multistrategy Learning is presented in figure 5. The input to a learning process is supplied either by the External World through Sensors, or from a previous learning step.

The Control module directs all processes. The Actuators perform actions on the External World that are requested by the Control module, e.g., an action to get additional information. The input is filtered by the Selection module, which estimates the relevance of the input to the learning goal. Only information that is sufficiently relevant to the goal is passed through. The current learning goal is decided by the Control Module according to the information received from an external "master'" system, e.g., teacher, or from the analysis of goals residing in the learner's knowledge base. The knowledge base is called a Multitype Knowledge Base (MKB) to emphasize the fact that it may contain, in the general case, different types of knowlege (various forms of symbolic and/or numeric and/or iconic knowledge), which can be specified at different levels of abstraction.

Learning goals are organized into a *goal dependency network* (GDN), which captures the dependency between different goals. Goals are represented as nodes, and the dependency among goals by labeled links. The labels denote the type and the strength of dependency.
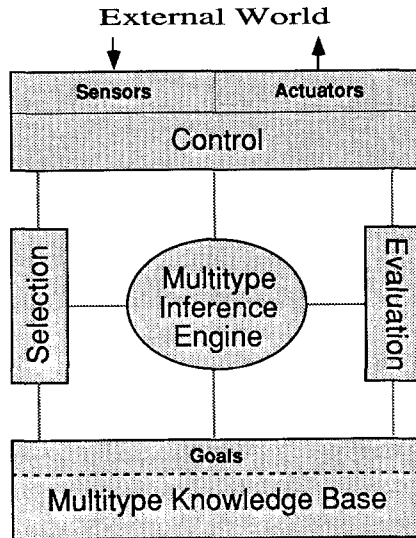
Figure 5. A general schema of a multistrategy task-adaptive learning (MTL) system.

If a goal G1 subsumes goal G2, then node G1 has an arrow pointing to node G2. For example, the goal "Learn rules characterizing concept examples" subsumes the goal "Find concept examples," and is subsumed by the goal "Use rules for recognizing unknown concept instances." The idea of a GDN network was introduced by Stepp and Michalski (1983) and was originally used for conceptual clustering.

In a general GDN for learning processes, the most general and domain-independent goal (represented by a node and no input links) is to store any given input and any plausible information that can be derived from it. More specific, though also domain-independent, goals are to learn certain types of knowledge.

For example, domain-independent goals may be to learn a general rule that characterizes facts supplied by the input, to reformulate a part of the learner's knowledge into a more efficient form, to determine knowledge needed for accomplishing some task, to develop a conceptual classification of given facts, to validate given knowledge, etc. Each of these goals is linked to some more specific subgoals. Some subgoals are domain dependent and call for determining some specific piece of knowledge, e.g., "learn basic facts about the Washington monument." Such a goal in turn subsumes a more specific goal, e.g., "learn the height of the Washington monument."

Any learning step starts with the goal either directly defined by an external source (e.g., a teacher, a failure to accomplish something, etc.) or determined by the analysis of the current learning situation. The control module dynamically activates new goals in the GDN as the learning process proceeds. The Multitype Inference Engine performs various types of inferences/transmutations required by the Control module in search for the knowledge specified by the current goal. Any knowledge generated is evaluated and critiqued by the Evaluation module from the viewpoint of the learning goal. If the knowledge satisfies the Evaluation module, it is assimilated into the MKB. It can then be used in subsequent learning processes.

Developing a learning system that would have all the features described above is a very complex problem, and thus a long-term goal. Current research explores more limited approaches to multistrategy-task adaptive learning. One such approach is based on building *plausible justification trees* (see the article by Tecuci in this issue).

Another approach, called *dynamic task analysis*, is outlined below. The learning system analyzes the dynamically changing relationship between the input, the background knowledge, and the current goal, and based on this analysis controls the learning process. The approach uses a knowledge representation that is specifically designed to facilitate all basic forms of inference. The representation consists of collections of type (or generalization) hierarchies and part hierarchies (representing part-of relationships). The nodes of the hierarchies are interconnected by "traces" that represent observed or inferred knowledge. This form of knowledge representation, called DIH ("Dynamically Interlaced Hierarchies"), allows the system to conduct different types of inference by modifying the location of the nodes connected by traces. This representation stems from the theory of human plausible reasoning proposed by Collins and Michalski (1989). Details are described by Hieb and Michalski (1993).

To give a very simple illustration of the underlying idea, consider a statement "Roses grow in the Summer." Such a statement would be represented in DIH as a "trace" linking the node *Roses*, in the type hierarchy of *Plants*, with the node *grow*, in the type hierarchy of *Actions*, and with the node *Summer*, in the hierarchy of *Seasons*. By "moving" different nodes linked by the trace in different direction, different transmutations are performed. For example, moving the node *Roses* downward to Yellow *roses* would be a specialization transmutation; moving it upward to *Garden flowers* would be a generalization transmutation. Moving the node *Summer* horizontally to *Autumn* would be a similization transmutation.

In the dynamic task analysis approach, a learning step is activated when system receives some input information. The input is classified into an appropriate category. Depending on the category and the current goal, relevant segments of MKB are evoked and designated as BK. The next step determines the type of relationship that exists between the input information and BK. The method distinguishes among five basic types of relationship. The classification presented below of the types of input is only conceptual. It does not imply that a learning system needs to process each type by a separate module. To the contrary, due to the underlying knowledge representation (DIH), all these functions are integrated into one seamless system, in which they are processed in a synergistic fashion. Here are the basic types of the relationship between the input and the background knowledge.

1. *The input represents programatically new information.* An input is pragmatically new to the learner if no entailment relationship can be determined between it and BK, i.e., if it cannot be determined if it subsumes, is subsumed by, or contradicts BK, within goal-dependent time constraints. The learner tries to identify parts of BK that are siblings of the input under the same node in some hierarchy (e.g., other examples of the concept represented by the input). If this effort succeeds, the related knowledge components are generalized, so that they account now for the input and possibly for other information stored previously. The resulting generalizations and the input facts are evaluated for "importance" (to the goal) by the Evaluation module, and those that pass an *importance criterion* are stored. If the above effort does not succeed, the input is stored, and the control is passed to case 4. Generally, case 1 involves some form of inductive learning (empirical learning, constructive induction), and/or learning by instruction.

2. *The input is implied by or implies BK.* This case represents a situation when BK accounts for the input or is a special case of it. The learner creates a derivational explanatory structure that links the input with the involved part of BK. Depending on the learning task, this structure can be used to create new knowledge that is more adequate ("operational," more efficient, etc.) for future handling of such cases. If the new knowledge passes an "importance criterion," it is stored for future use. This mechanism is related to ideas about the utility of explanation-based learning (Minton, 1988). If the input represents a "useful" result of a problem-solving activity, e.g., "given state x, it was found that a useful action is y, and if such a rule is sufficiently general to be evoked sufficiently often, then storing it is cost-effective. Such a mechanism is related to chunking used in SOAR (Laird, Rosenbloom, & Newell, 1986). If the input information (e.g., a rule supplied by a teacher) implies some part of BK, then an "importance criterion" is applied to it. If the criterion is satisfied, the input is stored, and an appropriate link is made to the part of BK that is implied by it. In general, this case handles situations requiring some form of analytic learning.

3. *The input contradicts BK.* The system identifies the part of BK that is contradicted by the input information, and then attempts to specialize this part. If the specialization involves too much restructuring or the confidence in the input is low, no change to this part of BK is made, but the input is stored. When some part of BK has been restructured to accommodate the input, the input also is stored, but only if it passes an "importance criterion." If contradicted knowledge is a specific fact, this is noted, and any knowledge that was generated on the basis of the contradicted fact is to be revised. In general, this case handles situations requiring a revision of BK through some form of synthetic learning or managing inconsistency.

4. *The input evokes an analogy to a part of BK.* This case represents a situation when the input does not match any background fact or rule exactly, nor is related to any part of BK in the sense of case 1, but there is similarity between the fact and some part of BK at some level of abstraction. In this case, matching is done at this level of abstraction, using generalized attributes or relations. If the fact passes an "importance criterion," it is stored with an indication of a similarity (analogy) to a background knowledge component, and with a specification of the aspects (abstract attributes or relations) defining the analogy. For example, an input describing a lamp may evoke an analogy to the part of BK describing the sun, because both lamp and sun match in terms of an abstract attribute "produces light."

5. *The input is already known to the learner.* This case occurs when the input matches exactly some part of BK (a stored fact, a rule or a segment). In such a situation, a measure of confidence associated with this part is updated.

Summarizing, an MTL learner may employ any type of inference and transmutation during learning. A deductive inference is employed when an input fact is consistent with, implies, or is implied by the background knowledge; analogical inference is employed when the input is similar to some part of past knowledge at some level of abstraction; and inductive inference is employed when there is a need to hypothesize new and/or more general knowledge. The above cases have been distinguished for the sake of theory. By using a proper knowledge representation (such as DIH), they all can be performed in a seamless way by one integrated mechanism.
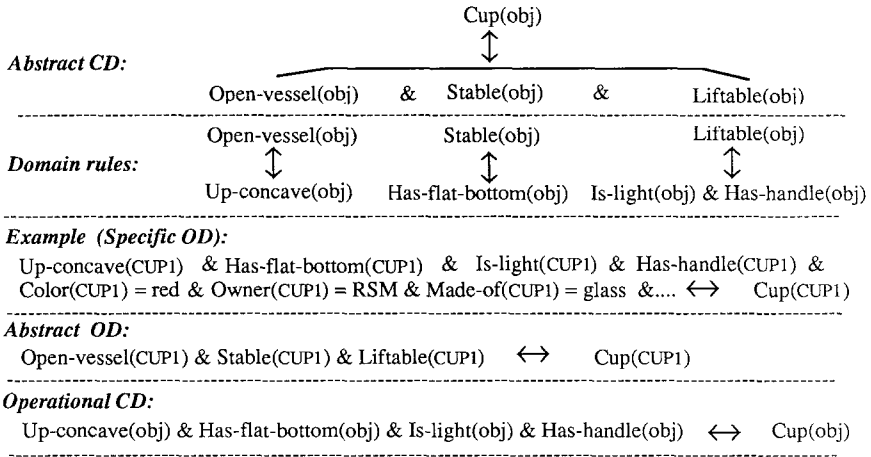
9. An illustration of MTL

To illustrate the ideas of MTL outlined above in terms of the Inferential Theory of Learning, let us use a well-known example of learning the concept of a "cup" (Mitchell, Keller & Kedar-Cabelli, 1986). The example is deliberately oversimplified, so that the ideas can be presented in a very simple way.

Figure 6 presents several inferential learning strategies as applicable to different learning tasks (as defined by a combination of the input, BK, and the desired output). For each strategy, the figure shows the input and the background knowledge required by a given learning strategy, and the output knowledge produced. The strategies are presented as independent processes only in a conceptual sense. In the actual implementation of MTL, all strategies are to be performed within one integrated inference sytem. The system specializes to any specific strategy using one general computational mechanism, based on Dynamic Interlaced Hierarchies (Hieb & Michalski, 1993). In figure 6, the name "obj" (in small letters) denotes a variable; the name "CUP1" (in capital letters) denotes a specific object. The top part of the figure presents the following:

* An *abstract concept description* (*Abstract CD*) for the concept "cup."
  Such a description characterizes a concept (or a set of entities that constitute the concept) in absract terms, i.e., in terms that are assumed not be directly observable or measurable. Here, it states that a cup is an open vessel that is stable and liftable. The relationship between individual conditions and the concept name is viewed as mutual implication.
* The *domain rules*.
  These rules (formally, m-implications) relate abstract terms to observable or measurable properties ("operational" properties). These rules can be used to derive abstract properties from operational properties or operational properties from abstract ones. For example, the abstract property "open vessel" can be derived from the observed property that the object is "up-concave," and that the object is "stable," if it has "flat bottom."
* A *specific object description* (*Specific OD*) of an example of a cup.
  Such a description characterizes a specific object (here, a cup) in terms of operational properties. By an example of a concept is meant a specific OD that is associated with the concept name.
* An *abstract object description* (*Abstract OD*).
  Such a description characterizes a specific object in abstract terms. It is not a generalization of an object, since its reference set is still the same object. Here, this description characterizes the specific cup, CUP1, in terms of abstract properties.
* An *operational concept description* (*Operational CD*).
  This description characterizes the concept in observable or measurable terms ("operational" terms). Such a description is used for recognizing the object from observable properties of the object. Notice that the argument of the predicates here is not some specific cup, but the variable "obj."

The bottom part of the figure illustrates several inferential learning strategies (corresponding to the primary knowledge transmutation involved) using the above concepts. For each strategy, the input to the process, the background knowledge (BK), and the goal description are specified.

Cup(obj)
↕

Abstract CD: ───

Open-vessel(obj)   &   Stable(obj)   &   Liftable(obj)

- -

Open-vessel(obj)   Stable(obj)   Liftable(obj)

Domain rules: ↕   ↕   ↕

Up-concave(obj)   Has-flat-bottom(obj)   Is-light(obj) & Has-handle(obj)

- -

Example (Specific OD):

Up-concave(CUP1) & Has-flat-bottom(CUP1) & Is-light(CUP1) & Has-handle(CUP1) &
Color(CUP1) = red & Owner(CUP1) = RSM & Made-of(CUP1) = glass &.... ↔   Cup(CUP1)

- -

Abstract OD:

Open-vessel(CUP1) & Stable(CUP1) & Liftable(CUP1)   ↔   Cup(CUP1)

- -

Operational CD:

Up-concave(obj) & Has-flat-bottom(obj) & Is-light(obj) & Has-handle(obj)   ↔   Cup(obj)

- -

Transmutation	_Input + BK:_		_Learning Goal:_
Abstraction	Example Domain rules	▷	Abstract OD
Deductive Generalization	Example Abstract CD Domain rules	▷	Operational CD
Empirical Induction	Examples BK'	▷	Operational CD
Constructive Induction (Case of Generalization)	Example(s) Domain rules	◁	Abstract CD
Constructive Induction (Case of Abduction)	Example(s) Abstract CD	◁	Domain rules
Multistrategy Task-adaptive Learning	Applies any of the above transmutation, or their combination, depending on the learning task, defined by the input, BK and the learning goal.		

Explanation: OD and CD stand for object description and concept description, respectively. CUP1 stands for a specific cup; obj denotes a variable. BK' denotes some limited background knowledge, e.g., a specification of the value sets of the attributes and their types. Symbol ↔ stands for mutual implication in which the merit parameters (the backward and the forward strength) are unspecified. Symbols |> and |< indicate deduction and induction, respectively.

Figure 6. An illustration of inferential strategies.

The input and BK are related to the goal description by a symbol indicating the type of the underlying inference: |> for deduction and |< for induction. A description of an object or of a concept is associated with a concept name by a mutual dependency relation ↔ (without defining the merit parameters). The use of mutual dependencies emphasizes a bidirectionality of the inference. If an unknown entity matches the left-hand side of the dependency, then it can be assigned the concept on the right-hand side. Conversely, if one

knows that an entity represents a concept on the right-hand side, then one can derive properties stated on the left-hand side of the dependency. The sign ↔ of mutual implication also expresses the fact that the general concept description is a hypothesis rather than a proven generalization. The mutual implication used here can be viewed as a generalization of the *concept assignment operator* "::>" which is sometimes used in the machine learning literature for denoting the link between a concept description and the corresponding concept name.

10. Summary

The Inferential Theory of Learning provides a unifying theoretical framework for characterizing logical capabilities (or competence) of learning processes. It analyzes learning processes in terms of generic patterns of knowledge transformation, called transmutations. Transmutations take input information and background knowledge, and generate some new knowledge. They represent either different patterns of inference ("knowledge-generation transmutations") or different patterns of knowledge manipulation ("knowledge-manipulation transmutations"). Knowledge-generation transmutations change the logical content of input knowledge, while knowledge-manipulation transmutations perform managerial operations that do not change the knowledge content. Transmutations can be performed using any kind of inference—deduction, induction, or analogy.

Several fundamental knowledge-generation transmutations have been analyzed and illustrated by examples: generalization, abstraction, and similization. These were shown to differ in terms of the aspects of knowledge that they change. Specifically, generalization and specialization change the reference set of a description; abstraction and concretion change the level of detail of a description of the reference set; and similization and dissimilization hypothesize new knowledge about a reference set based on the similarity or lack of similarity between the source and the target reference sets.

By analyzing diverse learning strategies and methods in terms of abstract, implementation-independent transmutations, the Inferential Theory of Learning offers a very general view of learning processes. Such a view provides a clear understanding of the roles and the applicability conditions of diverse inferential learning strategies and facilitates the development of a theoretically well-founded methodology for building multistrategy learning systems.

The theory was used to outline a methodology for multistrategy task-adaptive learning (MTL). An MTL system determines by itself which strategy, or combination of strategies, is most suitable for a given learning task. A learning task is defined by the input, background knowledge, and the learning goal. MTL aims at integrating strategies such as empirical and constructive generalization, abductive derivation, deductive generalization, abstraction, and analogy.

Many ideas presented here are at a very early stage of development, and a number of topics need to be explored in future research. Much more work is needed on the formalization of the proposed transmutations, on a clarification of their interrelationships, and on the identification and analysis of other types of knowledge transmutations. Future research needs to address also the problem of the role of goal structures, their representation, and the methods for their use for guiding learning processes.

Open problems also include the development of an effective method for measuring the amount of knowledge change resulting from different transmutations, and the amount of knowledge contained in various knowledge structures in the context of a given BK. Other important research topics are to systematically analyze existing learning algorithms and paradigms using concepts of the theory, that is, to describe them in terms of knowledge transmutations employed. A research problem of great practical value is to use the theory for determining clear criteria for the most effective applicability of different learning strategies in diverse learning situations.

The proposed approach to multistrategy task-adaptive learning was only briefly sketched. It needs much more work and a proof of concept. Future research should also investigate different approaches to the implementation of multistrategy task-adaptive learning, investigate their relationships, and implement experimental systems that synergistically integrate all major learning strategies. It is hoped that the presented research, despite its early state, provides a good insight into the complexities of research in multistrategy learning and that it will stimulate the reader to undertake some of the indicated research topics.

Acknowledgments

References

Adler, M.J., & Gorman, W. (Eds.). (1987). The great ideas: A synopicon of great books of the western world. *Encyclopedia Britannica*, Vol. 1, Ch. 39 (Induction), pp 565–571.

Aristotle (1987). Posterior analytics. In R.M. Hutchins (Ed.), *The works of Aristotle*, Vol. 1. *Encyclopedia Britannica*. Inc.

Bacon, F. (1620). *Novum Organum*. In R.M. Hutchins (Ed.), *Great Books of the Western World*, Vol. 30. *Encyclopedia Britannica*, 1987.

Bergadano, F., Matwin, S., Michalski, R.S., & Zhang, J. (1992). Learning two-tiered descriptions of flexible concepts: The POSEIDON system. *Machine Learning*, 8, 5–43. (Originally published in *Machine Learning and Inference Reports, No. MLI-3*, Center for Artificial Intelligence, George Mason University, Fairfax, VA, September 1990).

Birnbaum, L., & Collins, G. (1991). *Proceedings of the 8th International Conference on Machine Learning*. Chicago.

Bloedorn, E., & Michalski, R.S. (1991). Data-driven constructive induction. *Proceedings of the Tools for Artificial Intelligence Conference*. San Jose, CA.

Cohen, L.J. (1970). *The implications of induction*. London:

Collins, A., & Michalski, R.S. (1989). The logic of plausible-reasoning: A core theory, *Cognitive Science, 13*, 1-49.

Console, L., Theseider, D., & Torasso, P. (1991). On the relationship between abduction and deduction. *Journal of Logic and Computation, 1* (5).

Danyluk, A.P. (1987). The use of explanations for similarity-based learning. *Proceedings of IJCAI-87* (pp. 274-276). Milan, Italy.

Danyluk, A.P. (1989). Recent results in the use of context for learning new rules. Technical Report No. TR-98-066. Philips Laboratories.

Danyluk, A.p. (1993). Gemini: An integration of analytical and empirical learning. In R.S. Michalski & G. Tecuci (Eds.) *Machine learning: A multistrategy approach*. Vol. *IV*.

De Raedt, L., & Bruynooghe, M. (1993). Interactive theory revision. In R.S. Michalski & G. Tecuci (Eds.), *Machine learning: A multistrategy approach*. Vol. *IV*. Morgan Kaufmann.

Dietterich, T.G., & Flann, N.S. (1988). An inductive approach to solving the imperfect theory problem. *Proceedings of 1988 Symposium on Explanation-Based Learning*, (pp. 42-46). Stanford University, 1988.

Dietterich, T.G. (1986). Learning at the knowledge level. *Machine Learning, 1* (3), 287-316. (Reprinted in J.W. Shavlik & T.G. Dietterich (Eds.), (1990). *Readings in machine learning*. San Mateo, CA: Morgan Kaufmann.)

Fulk, M., & Case, J. (1990). *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*. University of Rochester, NY, August 6-8.

Giordana, A., Saitta, L., & Roverso, D. (1991). Abstracting concepts with inverse resolution. *Proceedings of the 8th International Workshop on Machine Learning* (pp. 142-146). Evanston, IL:

Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.

Goodman, L.A., & Kruskal, W.H. (1979). *Measures of association for cross classifications*. New York: Springer-Verlag.

Grosof, B.N., & Russell (1989). Declarative bias for structural domains. *Proceedings of the Sixth International Workshop on Machine Learning*, Cornell Univesity. Ithaca, NY: Morgan Kaufmann.

Hieb, M., & Michalski, R.S. (1993). A knowledge representation system based on dynamically interlaced hierarchies: Basic ideas and examples. *Reports of Machine Learning and Inference Laboratory MLI 93-5*, Fairfax, VA: Center for Artificial Intelligence, George Mason University.

Hunter, L. (1990). Planning to learn. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*. (pp. 26-34). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kodratoff, Y., & Michalski, R.S. (Eds.) (1990). *Machine learning: An artificial intelligence approach* (Vol. 3). Morgan Kaufmann.

Kodratoff, Y., & Tecuci, G. (1987). DISCIPLE-1: Interactive apprentice system in weak theory fields. *Proceedings of IJCAI-87* (pp. 271-273). Milan, Italy.

Laird, J.E. (Ed.). (1988). *Proceedings of the Fifth International Conference on Machine Learning*, University of Michigan, Ann Arbor, June 12-14.

Laird, J.E., Rosenbloom, P.S., & Newell, A. (1986). Chunking in SOAR: the anatomy of a general learning mechanism. *Machine Learning, 1(1)*, 11-46.

Lebowitz, M. (1986). Integrated learning: Controlling explanation. *Cognitive Science, 10* (2), 219-240.

Michalski, R.S. (1983). Theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach*. Tioga.

Michalski, R.S. (1990). Learning flexible concepts: Fundamental ideas and a method based on two-tiered representation. In Y. Kodratoff and R.S. Michalski (Eds.), *Machine learning: An artificial intelligence approach*, vol. III. Morgan Kaufmann.

Michalski, R.S. (1990a). Toward unified theory of learning: Multistrategy task-adaptive learning. *Reports of Machine Learning and Inference Laboratory MLI-90-1*. (Subsequently published in B.G. Buchanan and D.C. Wilkins (Eds.), *Readings in knowledge acquisition and learning*. Morgan Kaufmann.

Michalski, R.S. (1991b). A methodological framework for multistrategy task-adaptive learning. *Proceedings of the Fifth International Symposium on Methodologies for intelligent Systems*. Knoxville, TN: Elsevier.

Michalski, R.S. (1991). Toward unified theory of learning: An outline of basic ideas. In M. De Glas & D. Gabbay (Eds.), *Proceedings of the First World Conference on the Fundamentals of Artificial Intelligence*, Paris, France, July 1-5.

Michalski, R.S., & Kodratoff, Y. (1990). Research in machine learning: Recent progress, classification of methods and future directions. In Y. Kodratoff and R.S. Michalski (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 3). Morgan Kaufmann.

Minton, S. (1988). Quantitative results concerning the utility of explanation-based learning. *Proceedings of AAAI-88* (pp. 564–569). Saint Paul, MN.

Minton, S., Carbonell, J.G., Etzioni, O., Knoblock, C.A., & Kuokka, D.R. (1987). 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.

Mitchell, T.M., Keller, T., & Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view. *Machine learning. 1* (1), 47–80.

Mooney, R.J., & Ourston, D. (1993). A multistrategy approach to theory refinement. In R.S. Michalski & G. Tecuci (Eds.), *Machine learning: A multistrategy approach*, Vol. *IV.* Morgan Kaufmann.

Newell, (1981). The Knowledge Level. *AI Magazine*, 2, 1–20.

Muggleton, S. (1988). A strategy for constructing new predicates in first-order logic. *Proceedings of EWSL-88* (pp. 123–130). Glasgow, Scotland.

Pazzani, M.J. (1988). Integrating explanation-based and empirical learning methods in OCCAM. *Proceedings of EWSL-88* (pp. 147–166). Glasgow, Scotland.

Pierce, C.S. (1965). Elements of logic. In Ch. Hartshorne & P. Weiss (Eds.), *Collected papers of Charles Sanders Peirce* (1839–1914). Cambridge, MA: The Belknap Press, Harvard University Press.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* Morgan Kaufmann.

Piatetsky-Shapiro, G. (1992). Probabilistic data dependencies. In I.M. Zytkow (Ed.), *Proceedings of the ML92 Workshop on Machine Discovery*, Aberdeen, Scotland, July 4.

Plaisted, D. (1981). Theorem proving with abstraction. *Artificial Intelligence*, 16, 47–108.

Polya, G. (1968). *Mathematics and plausible reasoning*, Vol. I and II. Princeton, NJ: Princeton University Press.

Poole, D. (1989). Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5, 97–110.

Popper, K.R. (1972). *Objective knowledge: An evolutionary approach.* Oxford: Clarendon Press.

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

Ram, A. (1991). A theory of questions and question asking. *The Journal of the Learning Sciences, 1(3&4)*, 273–318.

Ram, A., & Hunter, L. (in press). The use of explicit goals for knowledge to guide inference and learning. *Applied Intelligence.* (Also available as Technical Report GIT-CC-92/04, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, January, 1992.)

Rivest, R., Haussler D., & Warmuth, M. (1989). *Proceedings of the Second Annual Workshop on Computational Learning Theory*, University of Santa Cruz, July 31–August 2.

Russell, S. (1989). *The use of knowledge in analogy and induction.* San Mateo, CA: Morgan Kaufmann.

Schafer, D. (Ed.). (1989). *Proceedings of the 3rd International Conference on Genertic Algorithms.* Fairfax, VA: George Mason University.

Schultz, T.R., & Kestenbaum, N.R. (1985). Causal reasoning in children. In G.J. Whitehurst (Ed.), *Annals of Child development*, vol. 2, pp. 195–249. JAI Press.

Segre, A.M. (Ed.). (1989). *Proceedings of the Sixth International Workshop on Machine Learning*, Cornell University. Ithaca, NY: Morgan Kaufmann.

Sleeman, D., & Edwards, P. (1992). *Proceedings of the Ninth International Workshop.* Great Britain: Aberdeen, Morgan Kaufmann.

Stepp, R.S., & Michalski, R.S. (1993). How to structure structured objects. *Proceedings of the International Machine Learning Workshop.* (pp. 156–160). Urbana, IL: University of Illinois Allerton House.

Tecuci, G., & Michalski, R.S. (1991a). A method for multistrategy task-adaptive learning based on plausible justifications. In L. Birnbaum & G. Collins (Eds.). *Machine Learning: Proceedings of the eighth international workshop.* San Matteo, CA: Morgan Kaufmann.

Tecuci, G., & Michalski, R.S. (1991b). Input 'understanding' as a basis for multistrategy task-adaptive learning. In Z. Ras & M. Zemankova (Eds.), *Proceedings of the 6th International Symposium on Methodologies for Intelligent Systems.* Lecture Notes on Artificial Intelligence. Berlin: Springer Verlag.

Tecuci, G. (in press). Plausible justification trees: A framework for deep and dynamic integration of learning strategies. *Machine Learning Journal* (Special Issue on Multistrategy Learning).

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

Utgoff, P. (1986). Shift of bias for inductive concept learning. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2). Morgan Kaufmann.

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

Whewell, W. (1857). *History of the inductive sciences*, 3rd edition. London.

Whitehall, B.L. (1990). Knowledge-based learning: Integration and deductive and inductive learning for knowledge base completion. Ph.D. Thesis, Computer Science Department, University of Illinois.

Whitehall, B.L., & Lu, S. C-Y. Theory Completion using Knowledge-Based Learning, In R.S. Michalski & G. Tecuci (Eds.), *Machine learning: A multistrategy approach*, Vol. IV. Morgan Kaufmann.

Wilkins, D.C., Clancey, W.J., & Buchanan, B.G. (1986). *An overview of the odysseus learning apprentice*. New York: Kluwer Academic.

Wnek, J., & Michalski, R.S. (1991a). Hypothesis-driven constructive induction in AQ17: A method and experiments. In K. Morik, F. Bergadano, and W. Buntine (Eds.), In *Proceedings of the IJCAI-91 Workshop on Evaluating and Changing Representation in Machine Learning*, (pp. 13–22). Sydney, Australia, August 24–30.

Wnek, J., & Michalski, R.S. (1991b). An experimental comparison of symbolic and subsymbolic learning paradigms: Phase I–Learning logic-style concepts. In R.S. Michalski & G. Tecuci (Eds.), *Proceedings of the First International Workshop on Multistrategy Learning*. GMU Center for Artificial Intelligence, Harpers Ferry, Nov. 7–9.

Wnek, J., & Michalski, R.S. (1993). Comparing Symbolic and Subsymbolic Learning: Three Studies. In R.S. Michalski & G. Tecuci (Eds.), *Machine learning: A multistrategy approach*, Vol. IV. Los Altos, CA: Morgan Kaufmann.

Zadrozny, W. (1990). The logic of abduction (preliminary report). *First International Workshop on Principles of Diagnosis*, Stanford, CA.