In *Machine Learning: A Multistrategy Approach, Volume 4*
R.S. Michalski & G. Tecuci (Eds.), Morgan Kaufmann Publishers, 1993.
(A slighly updated version, 2/18/99)

THE INFERENTIAL THEORY OF LEARNING:
Developing Foundations for Multistrategy Learning

Ryszard S. Michalski
Machine Learning and Inference Laboratory
George Mason University

and

Institute of Computer Science
Polish Acadecy of Sciences

Abstract

The development of multistrategy learning systems requires a clear understanding of the roles and the applicability conditions of different learning strategies. To this end, this chapter introduces the *Inferential Theory of Learning* that provides a conceptual framework for explaining logical capabilities of learning strategies, i.e., their *competence*. Viewing learning as a process of modifying the learner's knowledge by exploring the learner's experience, the theory postulates that any such process can be described as a search in a *knowledge space*, which involves the learner's experience, pior knowledge and the *learning goal*. The search operators are instantiations of *knowledge transmutations,* which are generic patterns of knowledge change. Transmutations may employ any basic type of inference — deduction, induction or analogy. Several fundamental knowledge transmutations are described in a novel and general way, such as generalization, abstraction, explanation and similization, and their counterparts, specialization, concretion, prediction and dissimilization, respectively. Generalization enlarges the *reference set* of a description (the set of entities that are being described). Abstraction reduces the amount of the detail about the reference set. Explanation generates premises that explain (or imply) the given properties of the reference set. Similization transfers knowledge from one reference set to a similar reference set. Using concepts of the theory, a *multistrategy task-adaptive learning* (MTL) methodology is outlined, and illustrated by an example. MTL dynamically adapts strategies to the *learning task*, defined by the input information, learner's background knowledge, and the learning goal. The goal of MTL research is to synergistically integrate a wide range of inferential learning strategies, such as empirical generalization, constructive induction, deductive generalization, explanation, prediction, abstraction, and similization.

Key words: learning theory, inference theory, multistrategy learning, deduction, induction, abduction, generalization, abstraction, prediction, analogy, knowledge 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

The last several years have marked a period of great expansion and diversification of methods and approaches to machine learning. Most of this research has been concerned with single learning strategy methods that employ one primary type of inference, within a specific representational or computational paradigm. Such *monostrategy* methods include, for example, inductive learning of decision rules or decision trees, explanation-based generalization, quantitative empirical discovery, neural net learning from examples, genetic algorithm based learning, conceptual clustering, reinforcement learning, and others. The research progress on these methods have been reported by many authors, for example, by 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).

Monostrategy systems are intrinsically limited to solving only certain classes of learning problems, defined by the type of input information they can learn from, the type of operations they can perform on a given knowledge representation, and the type of output knowledge they can produce. With the growing understanding of the capabilities and limitations of such *monostrategy* systems, there has been an increasing interest in *multistrategy learning* systems, which integrate two or more inference types and/or representational or computational paradigms. Multistrategy systems have a potentially greater competence, i.e., a greater ability to solve diverse learning problems, than monostrategy systems, which is due to a complementary nature of various learning strategies. On the other hand, their implementation presents a greater challenge, due to their greater complexity. Therefore, the effectiveness of their applicability to a given domain depends on the resolution of the above trade-off. Human learning is intrinsically multistrategy, and research on multistrategy systems is of significant relevance to its understanding, and thus is important regardless of their practical applications.

Among early well-known multistrategy systems (often called "integrated learning systems") are UNIMEM (Lebowitz, 1986), Odysseus (Wilkins, Clancey, and Buchanan, 1986), Prodigy (Minton et al., 1987), DISCIPLE (Kodratoff and Tecuci, 1987), Gemini (Danyluk, 1987, 1989; also 1993—chapter 7 in this book), OCCAM (Pazzani, 1988), IOE (Dietterich and Flann, 1988), and KBL (Whitehall, 1990; Whitehall and Lu, 1993—chapter 6). 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 usually done in a predefined, problem-independent way, and without clear theoretical foundations.

This book presents some of the most recent multistrategy learning systems. These include the system EITHER, for revising incorrect propositional Horn-clause domain theories using deduction, abduction or empirical induction (Mooney and Ourston, 1993—chapter 5); the system CLINT, for interactive theory revision represented as a set of Horn clauses (De Raedt and Bruynooghe, 1993—chapter 9); and the system WHY that learns using both causal models and examples (Baroglio, Botta and Saitta, 1993—chapter 12).

A remarkable aspect of human learners is that they are able to apply a great variety of learning strategies in a flexible and multigoal-oriented fashion, and to dynamically accommodate the demands of changing learning situations. Developing an adequate and general computational model of these abilities emerges as a fundamental long-term objective for machine learning research. To this end, it is necessary to investigate the principles and trade-offs characterizing diverse learning strategies, to understand their function and interrelationships, to determine conditions for their most effective applicability, and ultimately to develop a general theory of multistrategy learning. The theory should provide conceptual foundations for constructing learning systems that integrate a whole spectrum of learning strategies in a domain-dependent way. Such multistrategy systems would adapt the learning strategy or a combination of strategies to any given learning situation.

This chapter reports early results toward these goals, and presents a novel characterization of basic types of inference, and a variety of knowledge operators employing them. Specifically, it describes the *Inferential Theory of Learning* that views learning as a search through a *knowledge space*, guided by learning goals. The search operators are instantiations of certain generic types of knowledge change, called knowledge *transmutation*s (or knowledge *transforms*). Transmutations change various aspects of knowledge; some of them generate new knowledge, others only manipulate knowledge. They may employ different types of inference for this purpose. The first part of the chapter analyzes several fundamental knowledge transmutations, and the second part illustrates how the theory can be applied to the development of a methodology for *multistrategy task-adaptive learning* (MTL).

Inferential Theory of Learning strives to characterize logical capabilities of learning systems, that is their *competence*. To this end, it addresses such questions as what types of knowledge transformations occur in different learning processes; what is the validity of knowledge obtained through different types of learning, how prior knowledge is used; what knowledge can be derived from the given input and the prior knowledge; how learning goals and their structure influence learning processes; how learning processes can be classified and evaluated from the viewpoint of their logical capabilities, etc. The theory stresses the use of multitype inferences, the role of learner's prior knowledge, and the importance of learning goals.

The above aims 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 and Case, 1990; Warmuth and 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 the competence and the complexity of learning processes.

This chapter presents a novel and general analysis of several fundamental knowledge transmutations, such as generalization, abstraction, explanation, similization, and their counterparts, specialization, concretion, prediction, and dissimilization, respectively. Learning processes are analyzed at the level of abstraction that makes the theory relevant to characterizing machine learning algorithms, as well as to developing insights into the conceptual principles of learning in biological systems. The presented framework tries to formally capture many intuitive perceptions of various forms of human inference and learning, and suggests solutions that could be used as a basis for developing cognitive models. In a number of cases, the presented ideas resolve several popular misconceptions, such as that induction is the same as generalization, that generalization and abstraction are similar forms of inference, that induction must be data-intensive, that abduction is fundamentally different from induction, etc. They also suggest some new types of transmutations, e.g., inductive specialization, analogical generalization, similization, and others. A number of ideas in the theory stem from the research on the core theory of human plausible reasoning (Collins and Michalski, 1989).

To provide an easy introduction and a general perspective on the subject, many results are presented in an informal fashion, using conceptual explanations and examples, rather formal definitions and proofs. Various details and a formalization of many ideas await further research. To make the chapter easily accessible to both AI and Cognitive Science communities, as well as to readers who do not have much practice with predicate logic, expressions in predicate logic are usually accompanied by a natural language interpretation. Also, to help the reader keep track with a large number of symbols and abbreviations, they were compiled into a list included in the Appendix. The chapter is a modified version of the paper (Michalski, 1993), and represents a significant extension or refinement of ideas described in earlier publications (Michalski, 1983, 1990a,b & 1991).

2. BASIC TENETS OF THE THEORY

Learning has been traditionally characterized as an improvement of the learner's behavior due to 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 computational terms, what behavior changes need to be performed in response to what type of experience, how to efficiently implement them, how to evaluate them, how to employ the prior knowledge of the learner, etc. (By the "experience" is meant here the total information that a learner obtains from the outside in the course of learning.)

To provide answers to such questions, the Inferential Theory of Learning (ITL) assumes that learning is a goal-guided process of modifying the learner's knowledge by exploring the learner's experience. It attributes behavior change, e.g., a better performance in problem solving, to improvements of the learner's knowledge. The learner's knowledge includes both conceptual knowledge that represents the learner's understanding of the world, as well as control knowledge that is responsible for performing any skills.

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 —deduction, induction, or analogy. It involves "background knowledge," 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.

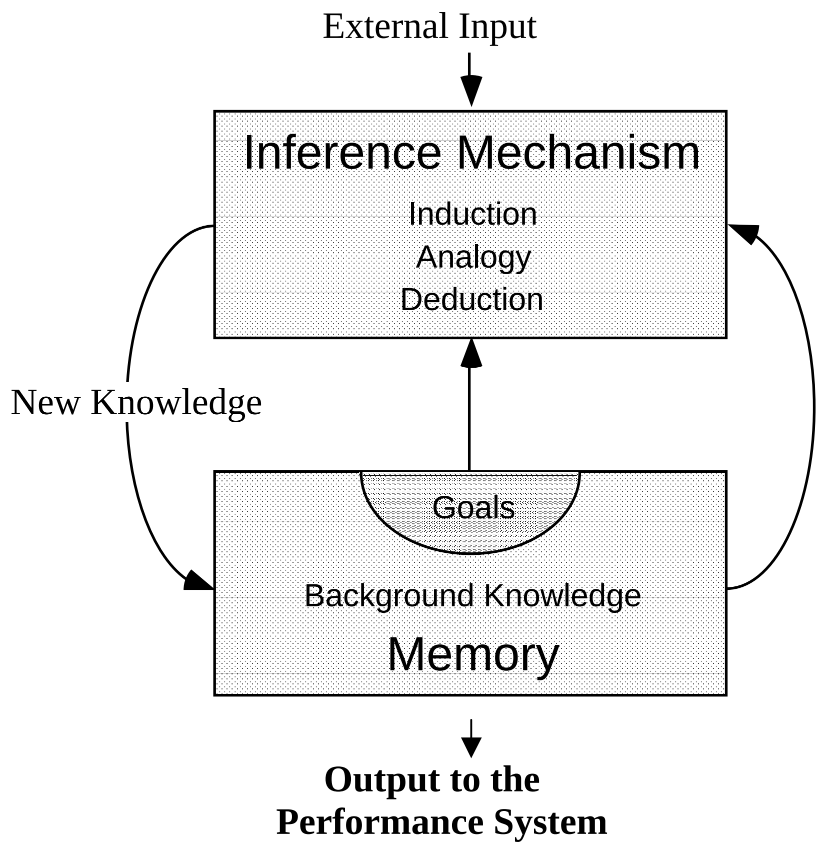

Figure 1: An illustration of a general learning process.

In each learning cycle, the learner analyzes the external input information using background knowledge and the given learning goal, and performs knowledge tranformations (inferences) that leand to knowledge satisfying the learning goal. Learning terminates if new knowledge satisfies the learning goal. A default learning goal is to increase the "total" knowledge of the system.

The term new knowledge is understood here very generally. The new knowledge can consist of *derived* knowledge, *intrinsic* (or *intrinsically new*) knowledge, or both. The new knowledge is called derived, if it is generated by deduction from prior learner's knowledge (it is a part of the "deductive closure" of the learner's knowledge that has actually been generated and stored in the learner's memory).

The new knowledge is called intrinsic (or intrinsically new) if it cannot be obtained by deduction from the learner's prior knowledge (that is, by truth-preserving "conclusive" deduction, according to the terminology proposed in Section 3). Such intrinsically new knowledge can be provided by an external source (a teacher or observation), or generated by induction, analogy, or contingent deduction. (A related concept is *pragmatically* new knowledge, which is knowledge that cannot be obtained by deduction from prior knowledge *using available computational resources*—time and/or space. Thus, pragmatically new knowledge includes both intrinsically new knowledge and knowledge that is theoretically deducible, but doing this is infeasible.)

The truth-status of derived knowledge depends on the validity of the background knowledge. The derived knowledge is true, if the premises for deduction are true. The truth-status of intrinsically new knowledge is typically uncertain (it is certain only if the knowledge is obtained not by inference, but communicated by a source that the learner trusts completely). Therefore, intrinsically new knowledge often needs to be validated by an interaction with an external information source, e.g., through an experiment.

A question arises as to whether learning occurs in the case where the only change in the learner's knowledge is a change in the knowledge organization or in the learner's confidence in the prior knowledge. The answer is yes to both parts of the question, and is based on the following arguments.

The theory assumes that any independent segment of the learner's knowledge (e.g., a sentence in predicate calculus or a rule) has three aspects: its content, its organization, and its certainty. The content is what is conveyed by a declarative knowledge representation (e.g., by a logical expression that represents this knowledge segment). The knowledge organization is reflected by the structure of the knowledge representation and determines the way in which the knowledge segment is used (e.g., the order in which components of a logical expression are evaluated).

To illustrate the above distinction, consider the following example. The knowledge content of a telephone book ordered alphabetically by the subscriber's name is the same as that of a book in which phone numbers are ordered numerically. The difference is only in the knowledge organization. Since change in the knowledge organization does not change the truth-status of knowledge (is truth-preserving), the result of such a change constitutes a special case of derived knowledge, and as such is new knowledge. Looking at this issue from another viewpoint, observe that different knowledge organizations facilitate different tasks. If a change in the knowledge organization improves the learner's performance of some tasks, and this improvement is required by the learning goal, then such a change is viewed as learning.

The *certainty* of a segment of a learner's knowledge reflects the degree to which the learner believes that this particular segment is true. It is a subjective measure of knowledge validity, in contrast to the objective validity determined by an objective measure, such as an experiment. Being a subjective measure, the learner's certainty may or may not agree with the objective validity.

The total change of a learner's knowledge in the process of learning consists collectively of changes in all of the three aspects—the knowledge content, its organization, and its certainty. The theory states that learning occurs if there is an increase of the total knowledge of a learner, or more precisely, if the learner's total knowledge changes in the direction determined by the learning goal. Even if the only change is in the certainty of some part of a learner's knowledge (as a result of obtaining some input or performing some inference), then the learner's total knowledge still increases, and thus the theory views this as learning.

If the results of a given learning step ("Output") satisfy the learning goal, they are assimilated within the learner's background knowledge and become available for use in subsequent learning processes. A learning system that is able to take the learned knowledge as an input to another learning process is called a *closed-loop* system; otherwise, is it called an *open-loop* system. It is interesting to note that human learning is universally closed-loop, while many machine learning programs are open-loop.

The basic premise of the Inferential Theory of Learning is that, in order to learn, an agent has to be able to perform *inference*, and has to possess the ability to *memorize* knowledge. The ability to memorize knowledge serves two purposes: to supply the background knowledge (BK) needed for performing the inference, and to record "useful" results of inference. Without both components—the ability to reason and the ability to store and retrieve information from memory—no learning can be accomplished.

Thus, one can write an "equation":

$$Learning = Inference + Memory$$

It should be noted that the term “inference” means here any possible type of reasoning, including any knowledge manipulation, formal and plausible reasoning, as well as random search for an abstractly specified knowledge target, etc. 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 is determined by the structure of the network (the number and the type of units used, and their interconnection), and by the initial weights of the connections. The learned knowledge resides in the new values of the weights. In a decision tree learning system, the BK includes the set of available attributes, their legal values, and an attribute evaluation procedure. 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 key idea of ITL is to characterize any learning process as a goal-guided search through a *knowledge space*, defined by the knowledge representation language and the available search operators. The search operators are specific applications of knowledge transmutations that a learner is capable of performing. Transmutations change various aspects of knowledge; some of them generate new knowledge, others only manipulate knowledge. Transmutations can employ any type of inference. Each transmutation takes some input information and/or background knowledge, and generates some new knowledge.

A learning process is then viewed as a sequence of knowledge transmutations that transform the initial learner’s knowledge to knowledge satisfying the learning goal (or goals).Thus, formally, ITL characterizes any learning process as a transformation:

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 the background knowledge BK.

The input knowledge, I, is the information (observations, facts, general descriptions, hypotheses) that the learner receives in the process of learning. The goal, G, specifies criteria to be satisfied by the output knowledge, O, in order to accomplish learning. The background knowledge, BK, is a part of the learner’s total prior knowledge that is relevant to a given learning process. (While a formal definition of “relevant” knowledge goes beyond the scope of this chapter, as a working definition the reader may assume that it is prior knowledge that is found useful at any stage of a learning process.)

Transmutations are generic classes of knowledge operators that a learner performs in the *knowledge space*. They are classes of knowledge transformations that correspond to some cognitively comprehensible and meaningful types of knowledge change. Thus, a change in knowledge that does not represent some identifiable and comprehensible knowledge transformation would not be called a transmutation. The knowledge space is a space of representations of all possible inputs, the learner’s background knowledge, and all the knowledge that the learner can potentially generate. In the context of empirical inductive learning, the knowledge space is usually called a *description space.*

Let us consider a few examples of transmutations. An *inductive generalization* takes descriptions of a subset of objects (e.g., concept examples), and hypothesizes a description of a superset. As shown in (Michalski, 1983), such a process can be characterized as an application of "inductive generalization rules." A *deductive generalization* derives a description of a superset of a given fact by employing background domain knowledge and deductive inference rules. A form of deductive generalization is *explanation-based generalization* (Mitchell, Keller and Kedar-Cabelli, 1986) that takes a concept example from an "operational" description space, a concept description from an "abstract" description space, and deduces a generalized concept description by employing domain knowledge linking the "abstract" and "operational" description spaces. Given some facts and background knowledge characterizing similar facts, an *analogical generalization* hypothesizes a general description of the given facts by drawing analogical inferences from the background knowledge. An *abstraction* takes a description of some set of entities, and transforms it to a description that conveys less information about the set, but preserving information relevant to the learner's goals. An *explanation transmutation*, given some facts, generates an explanation of them, by employing background knowledge that asserts that certain premises imply the given facts.

In general, a learning process can be a complex sequence of knowledge transmutations. Given some input and prior knowledge, a new piece of knowledge may be determined in a number of ways, e.g., through a deductive derivation, inductive generalization, or a *similization* transmutation (a form of analogy; see Section 8). An abstraction transmutation may re-express the derived piece of knowledge in a more abstract from. If the derived knowledge is hypothetical, a *generation* transmutation may generate additional facts, which are then used by a deductive transmutation to confirm or disconfirm the derived knowledge. If the knowledge is confirmed, it may be added to the original knowledge base by an *insertion* transmutation. The modified knowledge structure can be replicated in another knowledge base by a *replication transmutation*. The ultimate learning capabilities of a given learning system are determined by the types and the complexity of transmutations the system is capable of performing, and by what components of its knowledge it can or cannot change.

Another important tenet of the theory is that knowledge transmutations can be analyzed and described *independently* of the computational mechanism that performs them. This is analogous to the analysis of an information content of an information source independently of the ways information is represented or transmitted. Thus, ITL characterizes learning processes in an abstract way that does not depend on how transmutations are physically implemented. Transmutations can be physically implemented in a great variety of ways, using different knowledge representations and/or different computational mechanisms. In symbolic learning systems, knowledge 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 well-defined generalization rules, which represent conceptually understandable units of knowledge transformation (e.g., Michalski, 1983).

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

The above effect can be demonstrated by a *diagrammatic visualization* (DIAV) of concepts. In DIAV, concepts are mapped into sets of cells in a planar diagram representing 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 a diagrammatic visualization of inductive generalizations performed by a neural network, genetic algorithm, and two different symbolic learning systems are presented by Wnek and Michalski (1991b, Wnek and Michalski, 1993-chapter 18).

As indicated above, a learning process depends on the input information (input), background knowledge (BK), and the learning goal. These three components constitute a *learning task.* An input can be sensory measurements or knowledge from a source (e.g., a teacher), or the previous learning step. The input can be in the form of stated facts, concept instances, previously formed generalizations, conceptual hierarchies, certainty measures, or any combinations of such types.

A learning goal is a necessary component of any learning process, although it may not be expressed explicitly. Given an input, and a non-trivial background knowledge, a learner could potentially generate an unbounded number of inferences. To limit the proliferation of choices, a learning process has to be constrained and/or guided by the learning goal or goals. In human learning, there is usually a whole structure of interdependent goals. Learning goals determine 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. Goals can be classified into domain-independent and domain-dependent. Domain-independent goals call for a certain generic type of learning activity, independent of the topic of discourse, e.g., to derive knowledge of given type (e.g., justification) of the given knowledge, to concisely describe and/or generalize given observations, to discover a regularity in a collection of facts, to find a causal explanation of a given regularity, to acquire control knowledge to perform some activity, to reformulate given knowledge into a more effective form, to confirm a given piece of knowledge, etc. If a learning goal is complex, a learner needs to develop a plan specifying knowledge components to learn, and the order in which they should be learned (e.g., Hunter, 1990). A domain-dependent goal calls for acquiring a specific piece of knowledge about the domain. A learner may pursue several goals simultaneously, 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 importance of specific goals depends on the importance of higher-level goals. Thus, learning processes may be controlled by a hierarchy of goals, and the estimated degrees of their importance.

Most machine learning research has so far given relatively little attention 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 and Michalski, 1983; Hunter, 1990; Ram, 1991; Ram and Hunter, 1992). Among important research problems related to this topic are to develop 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 understanding learning in general, and interest in them will likely increase in the future.

In sum, the Inferential Theory of Learning states that learning is a goal-guided process of deriving desired knowledge by using input information and background knowledge. Such a process can be viewed as a search through a knowledge space, using transmutations as search operators. When a learning process produces knowledge satisfying a learning goal, it is stored, and made available for subsequent learning processes.

Transmutations represent generic patterns of knowledge change (knowledge generation, transformation, manipulation, etc.), and can employ any type of inference. To clearly explain their function, one needs to analyze different types of inference, and their interrelationships. To this end, Sections 3 to 5 discuss fundamental forms of inference, and give examples of transmutations based on them. Section 6 summarizes different types of transmutations currently recognized in the theory. Subsequently, Sections 7 and 8 analyze in detail several basic transmutations, such as generalization, abstraction, similization, and their counterparts, specialization, concretion, and dissimiliz ation. Sections 9 and 10 briefly discuss the application of the theory to the development of a methodology for multistrategy task-adaptive learning.

3. TYPES OF INFERENCE

Any type of inference may generate a piece of knowledge that can be useful for some purpose, and thus worth learning. Therefore, a complete theory of learning must include a complete theory of inference.

An attempt to schematically illustrate all basic types of inference is presented in Figure 2. The first classification is to divide inferences into two fundamental types: deductive and inductive.

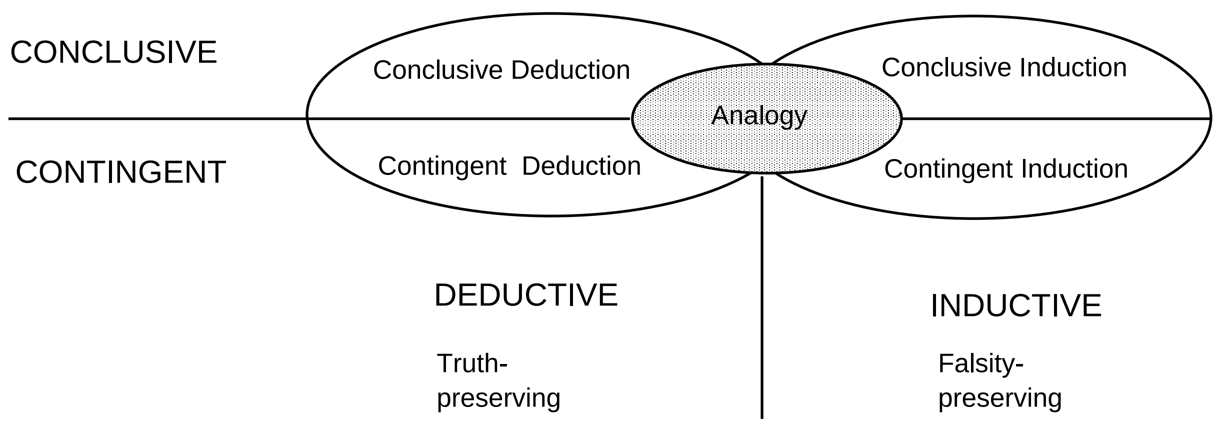

Figure 2: A classification of basic types of inference.

In defining these types, conventional approaches (like those in formal logic) do not distinguish between the input information and the reasoner's background knowledge. Such a distinction is important, however, for characterizing learning processes. Clearly, from the viewpoint of a learner, there is a difference between the information received from the senses, and the information that already resides in the learner's memory. Thus, making such a distinction better reflects cognitive aspects of reasoning and learning, and leads to a more adequate description of learning processes. To define basic types of inference in a general and language-independent way, let us consider an entailment:

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

where P stands for a set of statements, called the *premise*, BK stands for the reasoner's *background knowledge*, |= denotes semantic entailment, and C stands for a set of statements, called the *consequent*. It is assumed that P is logically consistent with BK.

Statement (1) can be interpreted: P and BK logically entails C; or, alternatively, C is a logical consequence of P and BK. Deductive inference is deriving consequent C, given P and BK. Inductive inference is hypothesizing premise P, given C and BK. Deduction can thus be viewed as tracing forward the relationship (1), and induction as tracing backward this relationship. Deduction is finding a logical consequence of given knowledge, and its basic form is truth-preserving (C must be true if P and BK are true). In contrast, induction is hypothesizing a premise that together with BK implies the input, and its basic form is falsity-preserving (if C is not true, then P cannot be true). Because (1) succinctly captures the relationship between two fundamental types of inference, we call it the *fundamental equation* for inference.

Inductive inference underlies several major knowledge generation transmutations, among them *inductive generalization* and *abductive derivation*. These two differ in the type of premise P they generate, and in the type of BK they employ. To put it simply, the differences between the two types of inference are as follows (a more precise characterization is given in Sections 4 and 5; see also examples below). Inductive generalization produces a premise P that is a generalization of C, i.e., P characterizes a larger set of entities than the set described by C. As shown later, inductive generalization can be viewed as tracing backward a tautological implication (specifically, the rule of *universal specialization*: $\forall x, P(x) \Rightarrow P(a)$). In contrast, abductive derivation produces a description that characterizes "reasons" for C. This is done by tracing backward an implication that represents some domain knowledge. If the domain knowledge represents a causal dependency, then such abductive derivation is called *causal explanation.* Other less known types of inductive transmutations include *inductive specialization* and *inductive concretion* (see Sections 5 and 6).

In a general view of deduction and induction that also captures their approximate or common sense forms, the standard logical entailment |= is replaced by a *contingent* or *weak* entailment |≈ (in contrast to contingent entailment, the standard entailment is called *conclusive* or *strong*). A contingent entailment means that C is only a *plausible, probabilistic,* or *partial* consequence of P and BK. The difference between these two types of entailments leads to another major classification of types of inference.

Specifically, inferences can be *conclusive* (or *strong*) or *contingent* (or *weak*). Conclusive inferences assume strong entailment in (1), and contingent inferences assume weak entailment in (1). Conclusive deductive inferences (also called *formal* or *demonstrative*) produce true consequences from true premises. Conclusive inductive inferences produce hypotheses that conclusively entail premises. Contingent deductive inferences produce consequents that may be true in some situations and not true in other situations; they are weakly truth-preserving. Contingent inductive inferences produce hypotheses that weakly entail premises; they are weakly falsity-preserving.

The intersection of deduction and induction, that is a truth- and falsity-preserving inference, represents an equivalence-based inference (or a *reformulation* transmutation, see section 6). Such an inference transforms a given statement (or set of statements) into a logically equivalent one. For example, if A is logically equivalent to A', then the rule A $\Rightarrow$ B can be transformed to rule A' $\Rightarrow$ B. Analogy can be viewed as an extension of such an equivalence-based inference, namely as a "similarity-based" inference. It occupies the central area in the diagram because deriving new knowledge by analogy can be viewed as a combination of induction and deduction.

The inductive step consists of hypothesizing that a similarity between two entities in terms of certain descriptors extends to their similarity in terms of some other descriptors. Based on this similarity and the knowledge of the values of the additional descriptors for the source entity, a *deductive* step derives their values for the target entity. An important knowledge transmutation based on analogical inference is *similization*. For example, if A is similar to A, then from A $\Rightarrow$ B one can plausibly derive A $\Rightarrow$ B. In order that such an inference can work, there is a tacit assumption that the similarity between A and A is *relevant* to B. This idea is explained and illustrated by examples in Section 9.

Let us now illustrate various knowledge transmutations based on the above basic forms of inference. A conclusive deductive inference is illustrated by the following transmutation:

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

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

Input	$q(a)$	(a has property q.)
BK	$a \in X$	(a is an element of X.)
	$(\forall x \in X, q(x)) \Rightarrow (a \in X \Rightarrow q(a))$	(If all elements of X have property q, then any element of X, e.g., a, must have property q.)
Output	$\forall x \in X, q(x)$	(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 (a) now characterizes a larger set (X). The output from induction is uncertain, which henceforth will be 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 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 can be consistently applied.

Using the above concepts, different inductive transmutations can be briefly characterized as follows:

- *inductive generalization* inductively extends the reference set of the input statement(s);
- *inductive specialization* inductively contracts the reference set,
- *abductive derivation* hypothesizes a premise (or an explanation) that implies the given input description according to some domain rule;
- *inductive concretion* hypothesizes additional details about the reference set described in the input statement (e.g., by hypothesizing values of more specific descriptors, or hypothesizing more precise values of the original descriptors; see Section 7).

Let us illustrate these transmutations by simple examples. The following is an example of inductive generalization:

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

In this example, the property q initially assigned only to element a has been hypothetically reassigned to characterize a larger set of elements (all elements in X). 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 (a did not have the property q), then the hypothetical premise (Output) would have to be false. The output was produced by tracing backward an implicative rule in BK.

Let us now turn to an example of inductive specialization:
Input $\exists x \in X, q(x)$ (There is an element in X that has property q)
BK $a \in X$ (a is an element of X)
$(a \in X \Rightarrow q(a)) \Rightarrow (\exists x \in X, q(x))$ (If some element a from X has property q, then there exists an element in X with property q)
Output $q(a)$ (*Maybe* a has property q)

The input statement can be restated as "One or more elements of X have property q." The reference set here is one or more unidentified elements in X. The inductive specialization hypothesizes that a specific element, a, from X has property q. Clearly, if this hypothesis and BK were true, the consequent would also have to be true. Again, the hypothesis was created by tracing backward an implicative rule in BK.

Here is an example of abductive derivation:
Input $q(a)$ (a has property q.)
BK $\forall x, x \in X \Rightarrow q(x)$ (If x is an element of X then x has property q.)
Output $a \in X$ (*Maybe* a is an element of X.)

The *Input* states that the reference set a has the property q. The abductive derivation hypothesizes a statement " a belongs to 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 previous two examples, Output was obtained by tracing backward an implicative rule in BK. Notice, however, an important difference from the previous examples, namely, that the implicative rule traced backward represents here some *domain knowledge* (that may or may not be true), rather than a universally true relationship (a tautological implication), which was used in the BK in the examples of inductive generalization and specialization.

To describe inductive concretion, suppose that q and q' are two attributes characterizing some entity a, and that q' is more specific than q. For example, a may be a personal computer, q its brand, MAC II, and q' its model: MAC II/fx. Here is an example of inductive concretion:

Input $q(a)$ (a has property q.)
BK $\forall x \in X, q'(x) \Rightarrow q(x)$ (If x is from X and has property q' then it has property q.)
Output $q'(a)$ (*Maybe* a has property q'.)

where X stands for a set of personal computers. The background knowledge states that if x is a MAC II/fx, then it is also a MAC II. Given that the computer is MAC II, a concretion transmutation hypothesizes that perhaps it is a MAC II/fx. Without more background knowledge, such a hypothesis would just be a pure guess. Having more BK, e.g., that the computer belongs to someone for whom the speed of the computer is important, and that MAC II/fx is presently the fastest desktop model of MAC II, then such a hypothesis would be plausible. Because q' is a more specific property than q, thus q' conclusively implies q, and the presented example of concretion is a form of conclusive induction.

The above examples illustrated several important types of inductive transmutations: inductive generalization, inductive specialization, abductive derivation, and concretion (other inductive transmutations are mentioned in Section 6). By reversing the direction of inference in these examples, that is, by replacing Output by Input, and conversely, one obtains the opposite transmutations, specifically, *deductive specialization*, *deductive generalization*, *prediction*, and abstraction, respectively. Prediction is viewed as opposite of abductive derivation, because it generates effects of the given premises ("causes"). While abductive derivation is tracing backward given domain rules, prediction traces them forward. Abstraction is viewed as opposite of concretion, because it transfers a more detailed description into a less detailed description of the given reference set.

The presented characterization of the above transmutations differs from the traditional views of these inference types, and it needs more justification. The next two sections give a more systematic analysis of the proposed ideas. We start with abduction, and its relation to contingent deduction.

4. ABDUCTION VS. CONTINGENT DEDUCTION

In the literature on abduction, many authors view it as a process of creating the "best" explanation of a given fact. A difficulty with this view is that it is not always easy to determine which explanation among alternative ones is the best. If producing an alternative explanation, but not the "best" one, is not classified as abduction, then what is and what is not abduction depends on the measure of "goodness" of an explanation, rather than on logical properties of inference. Another difficulty with this definition is that there are also deductive explanations. For example, if a child states "This orange is sweet," then one can explain this by saying "because oranges have a lot of fructose" (assuming that BK contains knowledge that "fructose is sweet.").

Some authors restrict abduction to processes of creating causal explanations of given facts, i.e., they limit it to inferences involving tracing backwards "causal implications." The example of abduction given in the previous section was based on the rule "If an entity belongs to X, then it has property q." This rule is not a causal implication, but a logical dependency. Consequently, according to such a view, the above example would not qualify as abduction. One may point out that Peirce, who originally introduced the concept of abduction, did not have any measure of "goodness of explanation" and did restrict abduction to a reasoning that produces only "causal" or "best" explanations (Peirce, 1965).

For a discussion on the relationship between abduction and deduction see (Console, Theseider and Torasso, 1991). Some theoretical views on abduction are in (Zadrozny, 1991). For an analysis and development of casual reasoning in humans see (Schultz and Kestenbaum,1985).

The proposed view of abduction extends usual characterizations of it. Abduction is viewed here as a form of knowledge-intensive induction that hypothesizes explanatory knowledge about a given reference set. This process involves tracing backward *domain-dependent* implications. Depending on the type of implications involved, the hypothesized knowledge may be a logical explanation, or a causal explanation. If there is more than one implication with the same consequent, tracing backward any of them is an abduction. The results of these abductions may have different credibility, depending on the "backward strength" of the implications involved (see below).

This view of abduction extends its conventional meaning in yet another sense. It is sometimes assumed that abduction produces only ground facts, meaning that the reference set is a specific object. As stated earlier, our view is that abduction generates explanatory knowledge that characterize a given reference set. If the reference set is a specific object, then abduction produces a ground fact; otherwise, it generates additional properties of the reference set. Below is an example of the latter form of abduction (variables are written with small letters):

Input	$\forall$x, In(x,S) & Banana(x) $\Rightarrow$ NotSweet(x)	(All bananas in shop S are not sweet.)
BK	$\forall$x, Banana(x) & FromB(x) $\Rightarrow$ NotSweet(x))	(Bananas from Barbados are not sweet.)
Output:	$\forall$x, In(x,S) & Banana(x) $\Rightarrow$ FromB(x)	(*Maybe* all bananas in S are from Barbados.)

In this example, the hypothesized output is not a ground statement, but a quantified expression. The output was generated by tracing backward an implicative rule in BK, and making a replacement in the right-hand-side of the input expression.

Let us now analyze more closely the view of abduction as an inference that traces backward implicative rules. It is easy to see that this view makes some tacit assumptions which, if violated, would allow 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.)

The inference that my pencil may be grass because it is green, clearly strikes us as faulty. The reason for this is that reversing implication in BK produces the implication:

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

which holds only with an infinitesimal likelihood.

This example demonstrates that abduction, if defined as tracing backward any implication, may produce a completely i mplausible hypothesis. This will happen if the “reverse implication” has insufficient “strength.” This simply means that standard abductive inference makes a tacit assumption that there is a sufficient “reverse strength” of the implications used to perform abduction. To make this issue explicit, we employ the concept of “mutual implication” as a basis for abductive reasoning.

Definition. A *mutual implication* or, for short, an *m-implication,* describes a logical dependency between statements in both directions:

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

where α and β are called *merit parameters*, and 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, which has merit parameters (or only one) sufficiently high to merit their consideration, without stating their precise values. For this purpose, we use symbols <---> (or --->), without listing α and β. Thus, an implication $A \Leftrightarrow B$: α, β, in which α and β are unspecified, but above some "threshold of acceptability," is alternatively written A <--> B, or A ---> B, if only α is above the threshold. The concept of mutual implication has been originally postulated in the theory of plausible reasoning (Collins and 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.

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 $\alpha = p(A \mid B)$ and $\beta = p(B \mid A)$. However, to make the concept of m-implication applicable for expressing many kinds of dependencies (including those occurring in human plausible reasoning), it is assumed that merit parameters do 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 and 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 can be characterized as those that involve mutual implications traced in the direction which has the strength parameter equal to 1 (it is also assumed that a match between an input statement and conditions in the implications are perfect, and the input information is perfect). Contingent inferences use m-implications in the direction which has the strength parameter less than 1, or involve imperfect input information (e.g., incomplete, imprecise, or incorrect information). Consider, for example, a statement: "Fire usually causes smoke." This statement can be represented as a mutual implication. If one sees fire somewhere, then one may derive a conclusion that there may be smoke there too. Conversely, observing smoke, one may hypothesize that there may be fire there. Assuming that this m-implication has both merit parameters smaller than 1, the above conclusions are uncertain. The first inference can be viewed as a contingent deduction, and the second inference as a contingent induction. Since the latter does not change the reference set (in this case, the area where there is fire or smoke), but derives an explanation of the reference set, it would be a contingent abduction.

Since both conclusions are uncertain, this might suggest that there is no real difference between contingent deduction and contingent abduction. A way to characterize the difference between the two types of inference is to check if the entailment |= in (1) could be interpreted as a causal dependency, i.e., if P could be viewed as a cause, and C as an effect. Contingent deduction derives a plausible consequent, C, of the causes represented by P. Abduction derives plausible causes, P, of the consequent C. Since we say that "fire causes smoke," and not conversely, then the above rule allows us to make a qualitative distinction between inferences that trace this implication in different directions. Contingent deduction can thus be viewed as tracing forward, and contingent induction (abduction, inductive generalization or specialization) as tracing backward contingent, causally-ordered m-implications.

The above distinction, however, is generally insufficient. The problem is that there are mutual implications that do not represent causal dependencies. For example, consider the statement "Prices at Tiffany tend to be high." This statement can be expressed as a non-causal m-implication :

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

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

We propose to resolve this problem by observing that in a standard (conclusive) deduction an m-implication is traced in the "strong" direction (with the degree of strength 1), and in an abductive derivation it is traced in the "weaker" direction. Extending this procedure to reasoning with mutually-contingent implications that are not causal dependencies, leads to the following rule:

> *If an m-implication is a non-casual dependency, then reasoning in the direction of the greater strength of the implication is a contingent deduction (e.g., contingent prediction), and reasoning in the direction of the weaker strength is a contingent induction (e.g., contingent abduction or contingent generalization).*

If a non-causal m-implication has equal strength in both directions, there is no distinction between contingent deduction and contingent induction. Considering the example with Tiffany, one may observe that α is usually significantly higher than β, unless Tiffany is the only expensive store in the area under consideration. Thus, the forward reasoning based on (3) can be viewed as a contingent deduction, and the backward reasoning as a contingent abduction. The distinction between contingent deduction and contingent abduction in the case of non-causal implications is thus a matter of degree.

Summarizing, contingent deduction and contingent abduction can be distinguished by the direction of causality in the involved m-implications. In case of non-causal implications, they can be distinguished on the basis of the strength of the merit parameters. Both forms of inference are truth- and falsity-preserving to the degree specified by the forward and backward merit parameters of the involved m-implications.

5. ADMISSIBLE INDUCTION AND INDUCTIVE TRANSMUTATIONS

Section 3 described induction as a general type of inference opposite to deduction, which includes several different forms. Inductive inference can produce hypotheses that can be generalizations, specializations, concretions or derivations from the given input. Another aspect of the general formulation of induction is that induction is not limited to inferences that use small amounts of background knowledge, i.e., to *knowledge-limited* or *empirical* induction, but may use a considerable amount of background knowledge, as in *knowledge-intensive* or *constructive* induction. A simple inductive generalization is an example of empirical induction, because it requires little background knowledge. Abduction can be viewed as a knowledge-intensive induction, because it requires domain knowledge in the form of implicative relationships.

An important aspect of inductive inference is that given some input information (a consequent C), and some BK (which by itself does not entail the input), the fundamental equation (1) can be satisfied by a potentially infinite number of hypotheses. Among these, only a few may be of any interest. One is usually interested only in "simple" and most "plausible" hypotheses. If a learner has sufficient BK, then this knowledge both guides the induction process, and provides constraints on the hypotheses considered. Due to BK, people are able to overcome limitations of background empirical (i.e., knowledge-limited) induction (Dietterich, 1989). The problem of selecting the "best" hypothesis among candidates appears in any type of induction. A standard method to limit a potentially unlimited set of hypotheses is to impose some additional extra-logical criteria. This idea is captured by the concept of an *admissible induction*.

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

$$P \cup BK \models C \tag{4}$$

and P satisfies the *hypothesis selection criterion.*

The selection criterion specifies how to choose a hypothesis among all candidates satisfying (4), 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 and Russell, 1989) *or* a *comparator* (Poole, 1989).

Ideally, the selection criterion should not be problem-independent, or dictated by a specific learning mechanism, but should specify properties of a hypothesis that reflect the *learner's goals*. This condition is not always satisfied by machine learning programs.

In some machine learning programs, the selection criterion is hidden in the description language employed (a "description language bias"). For example, a description language may be incomplete, in the sense that it may allow only to construct hypotheses in the form of conjunctive descriptions. If the "true" hypothesis is not expressible this way, the program cannot learn the concept. In human learning and in advanced machine learning programs the representation languages are complete, and the linguistic constraints apply only in the sense that some relationships are easy to express, and some are more difficult (Michalski, 1983; Muggleton, 1988).

Sometimes the selection criterion is dictated by the form of the representational system. For example, in decision tree learning, the selection criterion may seek a tree with the minimum number of nodes. This requirement does not, however, necessarily produce the most desirable concept descriptions. Because concept descriptions have to be expressed as a single decision tree, some unnecessary conditions may be introduced in the concept representation (Michalski, 1990).

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 (4), 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).

In general, the plausibility of a hypothesis depends on the background knowledge of the learner. The core theory of plausible inference (Collins and Michalski, 1989) postulates that the plausibility depends on the structural aspects of the organization of human knowledge (Hieb and Michalski, 1992), and on various merit parameters. 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. A “good” hypothesis selection criterion should take into consideration all the above characteristics.

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 and Gorman, 1987; Aristotle). 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.

As mentioned earlier, induction underlies a number of different knowledge transmutations, such as inductive generalization, inductive specialization, abductive explanation, and concretion. The most common form is inductive generalization, which is central to many learning processes. From properties of some entities in a given class, it hypothesizes properties of the entire class.

Inductive specialization creates hypotheses that apply to a smaller reference set as the one described in the input. Typically, a generalization is inductive and specialization is deductive. However, depending on the meaning of the input and BK, a generalization may also be deductive, and a specialization transmutation be inductive (see figure below). An abductive derivation generates hypotheses that explain the observed properties of a reference set, and is opposite to deductive prediction. Concretion generates more specific information about a given reference set, and is opposite to abstraction (see next section).

Examples of the above transmutations are presented in Figure 3. In the examples, to indicate that some m-implications are not conclusive (not logical implications), but sufficiently strong to warrant consideration, the symbol < --> is used. 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.”

To indicate that Outputs of the transmutations in Figure 3 are hypothetical, their symbolic expressions are annotated by certainty parameter α, which stands for “maybe.”

• Empirical inductive generalization

(Background knowledge-limited)

Input: Pntng(GF, Dwski) $\Rightarrow$ 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(a) (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 3: Examples of inductive transmutations.

The first, third, and fourth example in Figure 3 represent conclus ive 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 was:

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

("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 "$\subset$" is used under the assumption that cities, states, apartments and buildings can be viewed as sets of space parcels.

6. SUMMARY OF TRANSMUTATIONS

As stated earlier, transmutations are patterns of knowledge change, and they can be viewed as generic operators in knowledge spaces. A transmutation may change one or more aspects of knowledge, i.e., its content, organization, and/or its certainty. Thus, transmutations can generate intrinsically new knowledge, produce (deductively) derived knowledge, modify the degree of belief in some components of knowledge, or change knowledge organization. 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'):

$$T\text{: } S, E, BK \dashrightarrow S', E', BK' \tag{5}$$

Transmutations can be classified into two categories. In the first category are *knowledge generation* transmutations that change the content of knowledge and/or its certainty. 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. These transmutations change only knowledge organization. They can be performed on statements (well-formed logical expressions) or on terms (sets). They include inserting (deleting) knowledge components into (from) given knowledge structures, physically transmitting or copying knowledge to/from other knowledge bases, or ordering knowledge components according to some syntactic criteria. Since they do not change the content of knowledge (are truth-preserving), they can be viewed as based on deductive inference.

Transmutations are typically bi-directional 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 occurring 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(s). 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* that 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. The 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* that 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 of form of generation transmutation.

6. Agglomeration/decomposition

The agglomeration transmutation groups entities into larger units according to some goal criterion. If it also hypothesizes 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 and Michalski, l983). 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* that 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.

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* that 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 4 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 that way 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. This is illustrated by linking these transmutations with all three forms of inference. Exceptions from this rule are similization and dissimilization transmutations, which are based on analogy (analogy can be viewed as a special combination of deduction and induction). 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. In actual use, different transmutations are typically performed using only one type of inference.

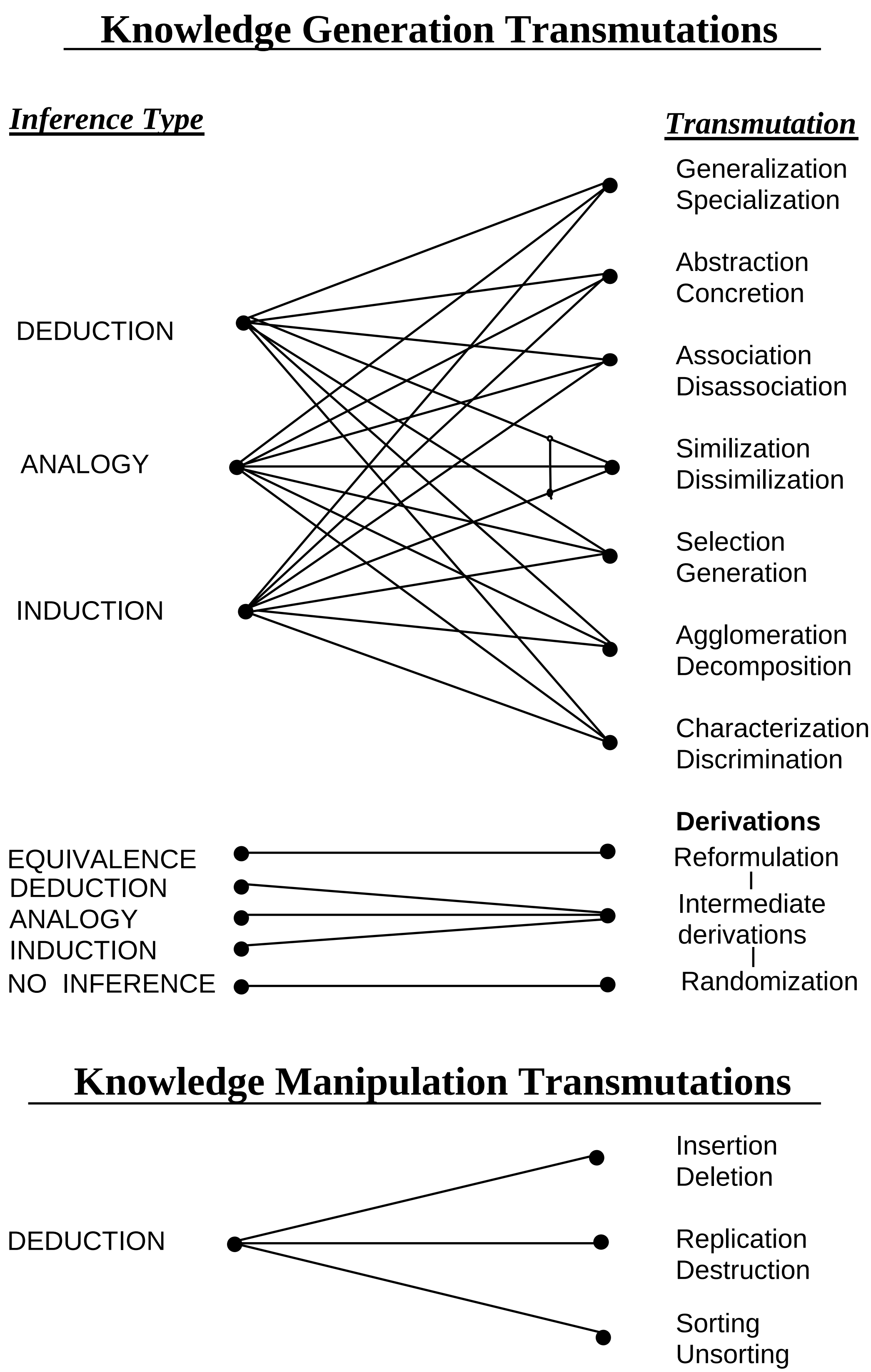

Figure 4: A summary of transmutations and the underlying types of inference.

For example, generalization and agglomeration are typically done through induction; and specialization and abstraction through deduction. Generalization, however, can 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.

Transmutations that employ induction, analogy or contingent deduction increase the amount of intrinsically new knowledge in the system (knowledge that cannot be conclusively deduced from other knowledge in the system). Learning that produces intrinsically new knowledge is called *synthetic* [some authors call it also "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 systems is called *analytic*. (Michalski and Kodratoff, 1990). Transmutations are not independent processes. An implementation of one complex transmutation may involve performing other transmutations.

Thus, the theory views transmutations as *types* of knowledge change, 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 relate this explanation to formal types of inference in a consistent way. Experiments performed with human subjects have 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.

Learning is viewed 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 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 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, e.g., may communicate the description to another learner. The next sections analyze some knowledge generation transmutations in greater detail, specifically, generalization, specialization, abstraction, concretion, similization and dissimilization.

7. GENERALIZATION VS. ABSTRACTION

This section analyzes two fundamental knowledge generation transmutations : 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.

7.1. 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 presented view more adequately captures the common intuitions and the natural language usage of the term "generalization." To express the proposed view more 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 which 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

$$R2 \supset R1 \quad \text{and}$$

$$D2[R2] \cup BK \Rightarrow D1[R1] \tag{5'}$$

$$\text{or}$$

$$BK \Rightarrow D2[R2] \tag{5''}$$

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.

<u>*Example 1.*</u> *(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: $\forall x$, Pntng(x, Dwski) $\Rightarrow$ 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 $\subset$ All_DPs

The interpretation of the predicate Btfl(R) is that the property Btfl applies to every element of the set R. Since R2 $\supset$ R1, and D2[R2] $\Rightarrow$ 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 $\subset$ Virginia (Fairfax is a part of Virginia)
S2 is more general than S1 because R2 $\supset$ R1, and D1[R1] $\cup$ BK $\Rightarrow$ 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 $\subset$ GKBld (John's apartment is a part of the Golden Key building)

S2: In(GKBld, Fire)
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 output 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 where 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 3 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 a 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 all its subsets, 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 transmutation, 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.

7.2. 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, for example, 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 (that latter 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

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

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 abstraction is when a picture of a table seen from one side (without seeing all legs), and 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.

7.3. 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 $d(\{s_i\},v)$, saying that descriptor d takes value v for entities from the set $\{s_i\}$. Thus, the reference set of this statement is $R = \{r_i\}$, $i = 1,2,...$, and a descriptive schema is $D[R] = d(R,v)$. Let us write the above statement in the form:

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

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* that increases the amount of information that is conveyed by a statement(s) about the given set of entities (reference set).

The two pairs of mutually opposite transmutations: {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).

8. SIMILIZATION VS. DISSIMILIZATION

The similization transmutation uses analogical inference to derive new knowledge. A dissimilization transmutation uses 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 are dependencies among various entities in the world. These dependencies can be of different strength or type, for example, 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 causal dependency), or that improving education of citizens is good for the country (this is an unquantified belief).

Such dependencies are often bi-directional, 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 a 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, is not equal. Betty may have lung cancer for other reasons, or 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 4 introduced the notion of mutual implication (eq.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 the 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 the similization and dissimilization transmutations. These transmutations involve determining a *similarity* or *dissimilarity* between entities, and then hypothesizing some new knowledge from this. The concept of similarity has been sometimes misunderstood in the past, and viewed as an objective, context-independent property of objects. In fact, the similarity between any two entities is highly context-dependent. Any two entities (objects or sets of objects) 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 express this, we use the concept of the *similarity in the context* of a given set of descriptors (introduced by Collins and Michalski, 1989). To say that entities *E1* and *E2* are similar in context (CTX) of the descriptors in the set D, we write

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

This statement says that values of the descriptors from D for the entity E1 and for the entity E2 differ no more than by some assumed tolerance threshold. For numerical descriptors, the threshold "Th" is expressed as a percentage, relative to the larger value. For example, if Th=10%, the values of the descriptor cannot differ more than 10%, relative 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 E1 has property A, and knowing that there is a similarity between E1 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, if there is a dependency between the descriptors defined by D and the property A. The reason for the latter condition can be illustrated by the following example. Suppose we know that some person who is handsome got their Ph.D. from MIT. It would not be reasonable to hypothesize that perhaps another person who we find handsome also got her/his 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 got the 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 $\Rightarrow$ A	
BK:	E1 DIS E2 in CTX(D)	
	D $\Rightarrow$ A: α > RT	
Output:	E2 $\Rightarrow$ ~A	(11)

where DIS denotes a relation of dissimilarity, and other parameters are like 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 a dissimilarity transmutation. Suppose we ar e told that apples grow in Poland. Knowing that apples are different from oranges in a number of ways, including the climate in which they normally grow, and that a climate of the area is m -dependent on the type of fruit grown there, one may hypothesize th at perhaps oranges do not grow in Poland. We will now illustrate the similization transmutation by a real -world example, and then show that it involves 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 US, 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 a good decision.

Using the schema (10), we have:

Input: Decision(Bld, NH) SIM Decision(Bld, NN) in CTX (FutCom)
BK: Decision(Bld, NH) $\Rightarrow$ Effect-on(U.S., good)
FutCom (US, x) $\Rightarrow$ Effect-on(US, x): $\alpha > RT$
Output: Decision(Bld, NN) $\Rightarrow$ Effect-on(US., good) (12)

where NH stands for National Highways and NN stands for National Network
Decision(Bld, x) is a statement expressing the decision to build x
FutCom(area, state) is 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) is a descriptor stating that "the effect on the US is x."

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

An inductive step:

Input: E1 SIM E2 in CTX(D)
BK: D $\Leftrightarrow$ A: $\alpha > T$
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 a nd 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 $\Rightarrow$ A(a)
Output: E2 $\Rightarrow$ A(a') (14)

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

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: Dec(Bld, NH) SIM Dec(Bld, NN) in CTX (FutCom(US, x))
BK: FutCom(US, x) $\Rightarrow$ Effect-on(US, y): $\alpha > T$
Output: Dec(Bld, NH) SIM Dec(Bld,NN) in CTX (FutCom, Effect-on) (15)

A deductive step:

Input: Dec(Bld, NH) SIM Dec(Bld, NN) in CTX (FutCom, Effect-on)
BK: Dec(Bld, NH) $\Rightarrow$ Effect-on(US, good)
Output: Dec(Bld, NN) $\Rightarrow$ 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 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 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.

As mentioned earlier, an opposite to a similization is a dissimilization transmutation. For example, knowing that two plants are very different from the viewpoint of the climate in which they grow, and that one lives in a particular area, one may hypothesize that the second plant may not be growing in that area. More details on dissimilization transmutation are in (Collins and Michalski, 1989).

Summarizing, a similization transmutation, given some piece of knowledge, hypothesizes another piece of knowledge based on the assumption that if two entities are similar in terms of some properties (or transformations characterizing their relationship), then they may be similar in terms of other properties (or transformations). This holds, however, only if the other properties are sufficiently related, by an m-dependency, to the properties used for defining the similarity.

9. 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 and 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 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 Multitype Knowledge Base to emphasize the fact that it may contain, in the general case, different types of knowledge (various forms of symbolic, numeric and 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 among 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. 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 & Michalski (1983), and originally used for conceptual clustering. In a general GDN for learning processes, the most general and domain-independent goal (represented by a node with no input links) is to store any given input and any plausible information that can be derived from it. More specific goals, though also domain-independent, 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, which call for determining some specific piece of knowledge, e.g., “learn basic facts about the Washington’s monument.”

Such a goal in turn subsumes a more specific goal “learn the height of the Washington’s monument.”

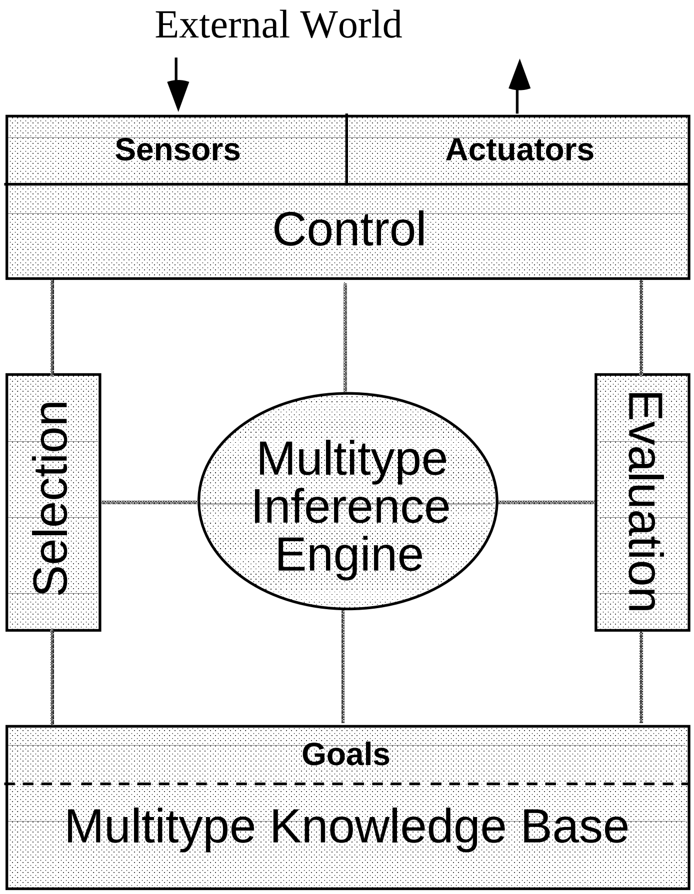

Figure 5: A general schema of a multistrategy task-adaptive learning (MTL) system.

Any learning step starts with the goal defined either directly 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 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 knowledge base. 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 chapter by Tecuci—chapter 11). 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 ("**D**ynamically **I**nterlaced **H**ierarchies"), 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 in (Collins and Michalski, 1989). Details are described in (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 *Action*s, 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. 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 and corresponding functions is only conceptual. It does not imply that a learning system needs to process each type by a separate module. In fact, 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 pragmatically 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, it is subsumed by, or it 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 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 synthetic learning (empirical learning, constructive induction, analogy), 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 the ideas on 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." If such a rule is sufficiently general so that it is evoked sufficiently often, then storing it is cost-effective. Such a mechanism is related to chunking used in SOAR (Laird, Rosenbloom, and 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 a 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 lever of abstraction; and inductive inference is employed when there is a need to hypothesize a new and/or more general knowledge. The above cases have been distinguished for the sake of theory. By using proper knowledge representation (such as DIH), they all can be performed in a seamless way by one integrated mechanism.

10. AN ILLUSTRATION OF MTL

To illustrate the above-sketched ideas in terms of the inferential theory of learning, let us use a well-known example of learning the concept of a "cup" (Mitchell, Keller and Kedar-Cabelli, 1986). The example is deliberately oversimplified, so that major ideas can be presented in a very simple way.

Figure 6 presents several inferential learning strategies as applicable to different learning tasks (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 produced output knowledge. 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 system. The system specializes to any specific strategy using the same general computational mechanism, based on Dynamic Interlaced Hierarchies (Hieb and Michalski, 1993). In the Figure 6, the name "obj" (in small letters) denotes a variable; the name "CUP1" (in capital letters) denotes a specific object. It defined a cup as an object that is an open vessel, is stable and is liftable.The top part of the figure presents:

- 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 abstract terms, i.e., in terms that are assumed not to be directly observable or measurable. Here, it states that a cup is an open vessel that is stable and liftable. Individual conditions are linked to the concept name by arrows.

Cup(obj)

⟷

Abstract CD:

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

Domain rules:

Open-vessel(obj) ⟷ Up-concave(obj)

Stable(obj) ⟷ Has-flat-bottom(obj)

Liftable(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 an of the above transmutations depending on the learning task, i.e., a given combination of the input, BK and the learning goal.		

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 spec ification 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 |< denote deductive and inductive transmutations, respectively.

Figure 6: An illustration of inferential strategies.

- The *domain rules*.

  These rules (formally, m-implications) relate abstract terms to observable or measurable properties ("operational" properties). These rules permit to derive (deductively) abstract properties from operational properties, or to hypothesize (abductively) operational properties from abstract ones. For example, the abstract property "open vessel" can be derived from the observed (operational) property that the object is "up-concave," or that 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, as 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 or measurable properties of the object. Notice that argument of the predicates here is not some specific cup, but the variable "obj."

The bottom part of the figure specifies several basic learning strategies (corresponding to the primary inferential transmutation involved), and presents learning tasks to which they apply. For each strategy, the input to the process, the background knowledge (BK), and the goal description are specified.

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 a concept is associated with a concept name by a mutual dependency relation <--> (without defining the merit parameters). Using the mutual dependency relation allows us to emphasize the fact that if an unknown entity matches the left-hand-side of the dependency, then this entity can be classified to a given concept.

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. This sign also implies that general concept description is a hypothesis rather than a proven generalization. The mutual dependency can be viewed as a generalization of the *concept assignment operator* " ::> " that is sometimes used in machine learning literature for linking a concept description with the corresponding concept name.

11. SUMMARY

This chapter presented the Inferential Theory of Learning that provides a unifying theoretical framework for characterizing logical capabilities of learning processes, and outlined its application to the development of a methodology for multistrategy task-adaptive learning. The theory 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 described in a novel way, and illustrated by examples: generalization, abstraction, and similization. They 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 their combination, 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 such strategies 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 of 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.

Acknowledgements

The author thanks Thomas Arciszewski, Eric Bloedorn, Mike Hieb, David Hille, Ibrahim Imam, Ken Kaufman, Zenon Kulpa, Marcus Maloof, Elizabeth Marchut-Michalski, Ray Mooney, Lorenza Saitta, David Schum, Anna Stein, Gheorghe Tecuci, Brad Whitehall, Janusz Wnek, his students from Machine Learning and Inference classes, and unknown reviewers for constructive suggestions, discussions, and criticisms that substantially helped in the preparation of this chapter.

This research was done in the Machine Learning and Inference Laboratory at George Mason University. The Laboratory's research is supported in part by the National Science Foundation under the Grant No. IRI-9020266, in part by the Office of Naval Research under the grant No. N00014-91-J-1351, and in part by the Defense Advanced Research Projects Agency under the grant No. N00014-91-J-1854, administrated by the Office of Naval Research and the grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research.

References

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

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

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

Baroglio, C., Botta, M. and Saitta, L., WHY: A System that Learns Using Causal Models and Examples, in *Machine Learning: A Multistrategy Approach, Volume IV*, Michalski, R.S. and Tecuci, G. (Eds.), Morgan Kaufmann Publishers, 1993.

Bergadano, F., Matwin, S., Michalski, R.S. and Zhang, J., "Learning Two-tiered Descriptions of Flexible Concepts: The POSEIDON System," *Machine Learning*, Vol. 8, pp. 5-43, 1992 (originally published in *Machine Learning and Inference Reports, No. MLI-3*, Center for Artificial Intelligence, George Mason University, September 1990).

Birnbaum, L. and Collins, G., *Proceedings of the 8th International Conference on Machine Learning*, Chicago, June 1991.

Bloedorn, E. and Michalski, R.S., Data-Driven Constructive Induction, *Proceedings of the Tools for Artificial Intelligence Conference*, San Jose, CA, 1991.

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

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

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

Console, L., Theseider, D. and Torasso, P., On the Relationship Between Abduction and

Deduction, *Journal of Logic and Computation*, Vol. 1, No. 5, October 1991.

Danyluk, A.P., “The Use of Explanations for Similarity-Based Learning,” *Proceedings of IJCAI-87*, pp. 274-276, Milan, Italy, 1987.

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

Danyluk, A.P., Gemini: An Integration of Analytical and Empirical Learning, in *Machine Learning: A Multistrategy Approach, Volume IV*, Michalski, R.S. and Tecuci, G. (Eds.), Morgan Kaufmann Publishers, 1993.

De Raedt, L. and Bruynooghe, M., “Interactive Theory Revision,” in *Machine Learning: A Multistrategy Approach, Volume IV*, Michalski, R.S. and Tecuci, G. (Eds.), Morgan Kaufmann Publishers, 1993.

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

Dietterich, T.G., “Limitations on Inductive Learning,” *Proceedings of the 6th International Workshop on Machine Learning*, Ithaca, NY, pp. 124-128, 1989.

Dietterich, T.G., “Learning at the Knowledge Level," *Machine Learning*, Vol. 1, No. 3, pp. 287-316, 1986 (Reprinted in J.W. Shavlik and T.G. Dietterich (Eds.) *Readings in Machine Learning*, San Mateo, CA: Morgan Kaufmann, 1990).

Fulk, M. and Case, J. *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, University of Rochester, N.Y., August 6-8, 1990.

Giordana, A., Saitta, L. and Roverso, D. “Abstracting Concepts with Inverse Resolution,” *Proceedings of the 8th International Workshop on Machine Learning*, pp. 142-146, Evanston, IL, June 1991.

Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

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

Grosof, B.N. and Russell, “Declarative Bias for Structural Domains,” *Proceedings of the Sixth International Workshop on Machine Learning*, Cornell Univerist y, Ithaca, New York, Morgan Kaufmann Puboishers, Inc. 1989.

Hieb, M. and Michalski, R.S. “A Knowledge Representation System Based on Dynamically Interlaced Hierarchies: Basic Ideas and Examples,” *Reports of Machine Learning and Inference Laboratory*, Center for Artificial Intelligence, George Mason University, 1993.

Hunter, L., “Planning to Learn,” *Proceedings of the Twelveth Annual Conference of the Cognitive Science Society*, pp.26-34, Hillsdale, NJ, Lawrence Erlbaum Associates, 1990.

Kodratoff, Y.and Michalski, R.S. (Eds.) *Machine Learning: An Artificial Intelligence Approach vol. III*, Morgan Kaufmann Publishers, Inc., 1990.

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

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

Laird, J.E., Rosenbloom, P.S., and Newell A., "Chunking in SOAR: the Anatomy of a General Learning Mechanism," *Machine Learning*, Vol. 1, No. 1, pp. 11-46, 1986.

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

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

Michalski, R.S., Learning Flexible Concepts: Fundamental Ideas and a Method Based on Two-tiered Representation, in *Machine Learning: An Artificial Intelligence Approach Volume III*, Y. Kodratoff and R.S. Michalski (Eds.), Morgan Kaufmann Publishers, Inc., 1990.

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

Michalski, R.S., "A Methodological Framework for Multistrategy Task-adaptive Learning," *Proceedings of the Fifth International Symposium on Methodologies for Intelligent Systems*, Knoxville, (Elsevier Pub.), October 1990b.

Michalski, R.S., "Toward a Unified Theory of Learning: An Outline of Basic Ideas," *Proceedings of the First World Conference on the Fundamentals of Artificial Intelligence*, M. De Glas and D. Gabbay (Eds.), Paris, France, July 1-5, 1991.

Michalski, R.S., "Inferential Theory of Learning as a Conceptual Framework for Multistrategy Learning," *Machine Learning Journal* (Special Issue on Multistrategy Learning), 1993.

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

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

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

Mitchell, T.M., Keller,T. and Kedar-Cabelli, S., "Explanation-Based Generalization: A Unifying View," *Machine Learning*, Vol. 1, No. 1, 47-80, 1986.

Mooney, R.J. and Ourston, D., A Multistrategy Approach to Theory Refinement, in *Machine Learning: A Multistrategy Approach Volume IV*, Michalski, R.S. and Tecuci, G. (Eds.), Morgan Kaufmann Publishers, 1993.

Muggleton, S., "A Strategy for Constructing New Predicates in First-Order Logic," *Proceedings of EWSL-88*, Glasgow, Scotland, pp. 123-130, 1988.

Newell, "The Knowledge Level," *AI Magazine*, No.2, 1-20, 1981.

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

Peirce, C.S., "Elements of Logic," in *Collected papers of Charles Sanders Peirce* (1839-1914), Ch. Hartshorne and P. Weiss (Eds.), The Belknap Press Harvard University Press, Cambridge, MA, 1965.

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

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

Plaisted, D., "Theorem Proving with Abstraction," *Artificial Intelligence*, Vol. 16, 47-108, 1981.

Polya, G., *Mathematics and Plausible Reasoning,* Vol. I and II, Princeton University Press, Princeton, NJ, 1968.

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

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

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

Ram, A., "A Theory of Questions and Question Asking," *The Journal of the Learning Sciences*, Vol.1, No. 3 and 4, pp.273-318, 1991.

Ram, A. and Hunter, L., "The Use of Explicit Goals for Knowledge to Guide Inference and Learning," *Applied Intelligence*, No. 2, pp. 47-73, 1992.

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

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

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

Schultz T.R. and Kestenbaum N. R., Causal Reasoning in Children, *Annals of Child Development,* G.J. Whitehurst (Ed.), vol. 2, pp. 195-249, JAI Press Inc., 1985.

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

Sleeman, D. and Edwards, P., *Proceedings of the Ninth International Workshop*, Univeristy of Aberdeen, G.B., July 1-3, Morgan Kaufmann Publishers, 1992.

Stepp, R.S. and Michalski, R.S., "How to Structure Structured Objects," *Proceedings of the International Machine Learning Workshop,* University of Illinois Allerton House, Urbana, pp. 156-160, June 22-24, l983.

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

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

Tecuci, G. "Plausible Justification Trees: A Framework for Deep and Dynamic Integration of Learning Strategies," *Machine Learning Journal* (Special Issue on Multistrategy Learning), 1993.

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

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

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

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

Whitehall, B. L., "Knowledge-based learning: Integration of Deductive and Inductive Learning for Knowledge Base Completion," *Ph.D. Thesis*, Computer Science Department, University of Illinois, 1990.

Whitehall, B.L. and Lu, S. C-Y., Theory Completion using Knowledge-Based Learning, in *Machine Learning: A Multistrategy Approach, Volume IV*, Michalski, R.S. and Tecuci, G. (Eds.), Morgan Kaufmann Publishers, 1993.

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

Wnek, J. and Michalski, R.S., “Hypothesis-Driven Constructive Induction in AQl7: A Method and Experiments,” *Proc. of the IJCAI-9l Workshop on Evaluating and Changing Representation in Machine Learning*, K. Morik, F. Bergadano, W. Buntine (Eds.), pp. 13-22, Sydney, Australia, August 24-30, 1991a.

Wnek, J. and Michalski, R.S.,“An Experimental Comparison of Symbolic and Subsymbolic Learning Paradigms: Phase I - Learning Logic-style Concepts,” *Proceedings of the First International Workshop on Multistrategy Learning,* R.S. Michalski and G. Tecuci (Eds.), GMU Center for Artificial Intelligence, Harpers Ferry, Nov. 7-9, 1991b.

Wnek, J. and Michalski, R.S., “COMPARING SYMBOLIC AND SUBSYMBOLIC LEARNING: Three Studies,” in *Machine Learning: A Multistrategy Approach*, Volume IV, Michalski, R.S. and Tecuci, G. (Eds.), Morgan Kaufmann, Los Altos, CA, 1992.

Zadrozny, W. “The Logic of Abduction (Preliminary Report),” *First International Workshop on Principles of Diagnosis*, Stanford, CA., 1990.

APPENDIX

A Table of Symbols, Abbreviations and Definitions of Fundmental Concepts

$\cup$	Set-theoretical union
$\subset$	Subset relation
$\supset$	Superset relation
|=	Logical entailment relation
~	Logical "NOT"
&	Logical "AND"
$\Rightarrow$	Logical implication or unidirectional mutual implication
$A \Leftrightarrow B$: α, β	Mutual dependency (or m -dependency) between A and B; if A and B are statements (well-formed logical expressions) then an m -dependency becomes mutual implication; it if A and B are terms, then an m -dependency represents a relationship between terms. Parameters α, β, are called merit parameters, and express the forward and backward strenght of the dependency, respectively.
A < --> B	Mutual dependency in which merit parameters are not defined
$\forall$x, P(x)	Universal quantification (for every x, P is true)
|>	Deductive knowledge transmutation
|<	Inductive knowledge transmutation
BK	Background knowledge
CTX	Context in which similarity or dissimilarity is measured
D[R]	Descriptive schema of the reference set R
R	The reference set of a description
ITL	Inferential theory of learning
m-dependency	Mutual dependency (see above)
m-implication	Mutual implication (see above)
MTL	Multistrategy task-adaptive learning
OD	Object description
SIM	Similarity relation
DIS	Dissimilarity relation

Definition of Fundamental Concepts

***Data**:* *A set of symbols*

***Information**:* *Interpreted data*

***Knowledge**:* *Organized, generalized and abstracted information*

Intelligent system: *A system endowed with the capability to:*

C1. perceive (has sensors that generate information about the environment)

C2. learn (create knowledeg from that information), and

C3. reason (use that knowledge for achieving its goals)