LEARNING FLEXIBLE CONCEPTS: FUNDAMENTAL IDEAS AND A METHOD BASED ON TWO-TIERED REPRESENTATION

by

Ryszard S. Michalski

Machine Learning: An AI Approach, Vol. III, Y. Kodratoff and R.S. Michalski (Eds)
San Mateo, CA. Morgan Kauffman Publ., pp. 63-111, June 1990.

LEARNING FLEXIBLE CONCEPTS:

A Method Based on Two-tiered Representation and the AQ15 Program

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

Abstract

Most human concepts elude precise definition, as they have fluid boundaries and context-dependent meaning. We call such concept *flexible*, in contrast to *crisp* concepts that are well-defined and context-independent. Machine learning research has so far concentrated on learning crisp concepts, and learning flexible concepts emerges as new fundamental challenge to the field.

This chapter describes an approach to learning flexible concepts that employes the idea of *two-tiered representation,* in which the meaning of a concept depends on two components: the *base concept representation* (BCR), and the *inferential concept interpretation* (ICI). The BCR (the 1st tier) expresses the general, typical and easy-to-define concept meaning, while the ICI (the 2nd tier) defines allowable modifications of the typical meaning, the matching procedures, context-dependency, and describes special cases. In matching an instance with a concept, the ICI may employ deductive, analogical or inductive inference.

In the method desribed, the BCR is determined in two steps. First, a complete and consistent concept description is induced from concept examples using a domain-independent learning program AQ15. This description is then reduced and optimized according to a *description quality measure*. The ICI is given to the system in the form a procedure for a simple *flexible* matching, which determines the "closeness" between an instance and the BCR of a concept.

Early experiments with this methodology have shown that by shifting a large part of the concept meaning from the BCR to the ICI, the amount of memory needed for concept representation can be greatly reduced, without decreasing its performance accuracy on new examples. This surprising and potentially significant result opens a new avenue for research. In particular, new research is needed to determine the generality of these ideas across different domains and consistency of the results. Further research is also required to explore more advanced two-tiered knowledge representations and methods of applying them in learning systems.

1. Introduction

Machine learning research has so far focused primarily on learning concepts that have precise and context-independent meaning. Such *crisp* concepts are represented by well-defined symbolic structures, which specify precisely the concept boundaries, and thus hold either true or false for any given object or observation. It is also often assumed that all instances of a concept are equally representative, and that the meaning of a concept is independent of the context of discourse. Examples of such representations include logical expressions, semantic networks and frames. Relevant references, can be found, e.g., in (Michalski, Carbonell and Mitchell, 1983 and 1986).

The above-mentioned assumption that concepts are crisply defined is not limited to machine learning research. In fact, the tendency to use crisp concepts has been characteristic of every

scientific activity. The precision of concepts and the clarity of their interrelationships has traditionally been and remains a clear mark of quality of scientific theories.

Yet, most human concepts characterizing objects in the real world and our activities in it are *flexible*, as they have fluid, imprecisely definedboundaries and context-dependent meaning. Note how difficult is to define precisely concepts such as chair, key, space, game, freedom, mechanism or the like, that we regularily use in our communication. Even concepts that have been given a formal scientific definition are often used flexibly and with context-dependent meaning.

Consider, for example, the well-defined concept of a triangle. It is typically used to describe a geometrical configuration, but it can also be used to characterize a configuration of streets, a relationship among people, or a shape of a musical instrument (that has the name based on its shape). In all these usages some common core properties are preserved, but the specific meaning depends on the context in which the concept is used.

Moreover, even the mathematical meaning of a triangle allows one to distinguish between more or less *typical* triangles. There is, generally, a perceived distinction in the representativeness of different instances for any given concept. As an example, consider the concept of a bird. A sparrow is viewed as a more typical representative of a bird in our country than, say, an ostrich or kiwi. In most machine learning programs, however, the distinction between the degrees of typicality of different concept examples has not been taken into consideration. Except for a few efforts, for example, the idea of *near miss* (Wiston, 1970), or the method of *outstanding representatives* for selecting "best" learning examples (Michalski and Larson, 1978), the issue of typicality of examples recieved little attention in machine learning research.

A related and also relatively little explored issue is how to take into consideration the accuracy or precision with which individual examples are described or represented. For example, a triangle may be drawn in many different ways, for example, with dotted lines or lines made of other shapes, or can be projected as a shadow on a wall, and still recognizable as a triangle. Thus, examples may vary greatly in the forms they are presented, be strongly distorted or modified but nevertheless may represent the same concept. Determining a definition of a flexible concept that would capture all its possible forms and permissible modifications is a very difficult and unresolved problem.

Finally, the complete meaning of a concept to a person depends on the amount of knowledge this person possesses about it. Clearly, the conceptualization of a triangle by a layperson is different from that of a mathematician specializing in geometry. The difference is in the amount facts they know, and the depth of understanding they have of the concept and its properties. Such background knowledge-dependency in understanding a concept indicates that human concepts are personalized, living and growing constructs, rather than fixed and stable entities that mean exactly the same thing to all people using them. The meaning of a concept changes not only from an individual to an individual, but also evolves in time. Note how even scientific concepts have changed and evolved, for example, the concept of energy, force, light, gravitation, atom, or electricity. Non-scientific concepts are almost always flexible, rather than crisp. Thus, human concepts are, in general, very different from concepts used in today's computer systems, which are well-defined and context-independent.

Some authors have relaxed crisp concept representations by associating with them a probability or set membership distribution (e.g., Zadeh, 1976). These methods can handle some aspects of flexible concepts and have found a number of applications. Their ability to adequately capture the immense fluidity, context-modifiability and background knowledge-dependency of human concepts is, however, limited. There is a growing realization that the imprecision of human concepts stems not so much from their statistical nature or from some undesirable vagueness of our definitions, but rather from the need for cognitive economy. By allowing concepts to be flexible

and context-dependent, we can extend their expressive power, and by this improve cognitive economy of our descriptions of the world (Michalski, 1986).

It should be noted, however, that although concepts may be individually flexible, sentences consisting of such concepts may convey precise meaning in a given context. For example, the sentence "this tall man in the group of people in the room" conveys precise meaning (i.e., precisely identifies the man), if there is only one man visibly taller than others, although the concept "tall" by itself is flexible.

If machine learning is to provide useful mechanisms for real-world concept acquisition, and to resolve the knowledge acquisition bottleck that plagues artificial intelligence systems, it has to solve the problem of representing and learning flexible concepts. Further development of practical inductive learning programs capable of acquiring concepts from their instances is also predicated upon a solution of this problem. Learning flexible concepts is thus a fundamental, yet unsolved problem in machine learning.

This chapter describes an approach to learning flexible concepts based on the recently proposed idea of *two-tiered concept representation* (Michalski, 1986). It presents basic ideas, a computational method and briefly describes the inductive learning program AQ15, which is the major component of the method. We also present some results from early experiments with the method. A detailed desciption of the program AQ15 and the experimental results are in [Michalski, Mozetic & Hong 86; and Hong, Mozetic & Michalski 86]. New improvements to the method and further experimental results with two tiered representation are in [Bergadano at al, 1988a,b].

2. Two-Tiered Concept Representation

Concepts can be viewed as symbolic structures denoting classes of entities united by some principle. The principle may be specified in a wide range of forms: from an explicit definition to a weak implication by the context and the useage of the concept. Typically, for concepts relating to the events in the real world, the boundaries of these classes are flexible. The flexibility of the meaning depends on the context of discourse and on the intepreter's background knowledge related to the concept. A simple introspection tells us that there can be a different degree of match between our internal representation of a concept and an observed concept instance, and performing such a match may involve inference (see, e.g., examples below). There can be great diversity in the meaning of some concepts depending on the situation and the context in which they are used.

How, then, concepts with an imprecise, variable and context-dependent meaning can be learned and represented? This chapter attempts to shed light on this problem by employing the idea of *two-tiered concept representation*, introduced by the author (Michalski, 1986). This idea is illustrated by examples, both hypothetical and practical, mostly taken from the medical domain.We also present a computational method employing this idea.

In two-tiered representation, the meaning of a concept is defined by two components: the *base concept representation* (BCR), and the *inferential concept interpretation* (ICI). The base concept representation (1st tier) includes assertions describing the general and easy-to-define meaning of the concept, and its typical purpose or use. It may also include representative examples and counter-examples of the concept, as well as observed exceptions from its typical meaning. The BCR is thus an explicit memorized set of basic facts about a concept.

The inferential concept interpretation (2nd tier) specifies how to match the concept with instances, what are the allowable modifications of the typical meaning, how the concept depends on the context, and how to handle special cases. The ICI may employ deductive, analogical or inductive inference.The interpretation method may use the concept metaknowledge, for example, the

importance of concept attributes, observed frequencies of concept occurence, the context of the discourse, and the interpreter's background knowledge. This way the actual meaning of any concept can be flexibly modified or extended by varying the context and/or the method of interpretation, without changing the base concept representation. By evaluating the amount of inference involved in matching a concept with an instance one may also produce a qualitative or numerical estimation of the typicality or certainty that an instance is a member of the concept in a given context (a "concept membership function").

The above remark points to the principal difference between this approach and, for example, the fuzzy logic approach (Zadeh, 1976) to representing imprecise concepts. In fuzzy logic, a concept is associated with a numeric membership function, defined by a human. In the proposed approach, we have a set of inference rules and an interpretation method (2nd tier) associated with a concept. These rules and the method could be used to compute the membership function in different contexts. The rules are modifiable and can grow with the usage of a concept. Figure 1 illustrates the relationship between BCR and ICI in a two-tiered concept representation.

Figure 1. An illustration of the relationship between the base concept representation (BCR) and inferential concept interpretation (ICI).

Learning a concept using a two-tiered representation consists of two parts: one - to acquire the base representation, and the second - to acquire the interpretation method and inference rules in various contexts. The interpretation method can be shared by concepts in the same class, or inherited from a superclass. By sharing complete or partial ICI, a significant economy of the concept representation can be achieved. Let us now illustrate the idea of two-tiered representation by a few examples.

Example 1: Concept of fish

Typical characteristics of fish are that they live in water and they swim. This information, as well as information about their typical shape, physical properties and other relevant facts would be stored in the base concept representation.

Suppose someone found an animal that matches many physical characteristics of fish, but which does not swim. Suppose that this animal appears to be sick. The inferential concept interpretation would involve background knowledge that sick animals may not be able to move, and that swimming is a form of moving.

By deductive reasoning from these facts one concludes that apparent lack of ability to swim should not be taken as negative evidence for the animal being a fish. To the contrary, the fact that the animal does not swim might even add to the confidence once the animal was recognized as being sick (a sick-looking toy-fish may still "swim" if its motor is ok). The rules for such reasoning would not be stored with the concept of fish, but would be a part of the concept interpretation method. The method could be associated with the concept of animal, and by inheritance applied to the concept of fish.

Example 2: Concept of sugar maple

Our prototypical image of a sugar maple is that it is a tree with three- to five-lobed leaves that have V-shaped clefts. Some of us may also remember that the teeth on the leaves are coarser than those of red maple, that slender twigs turn brown, and the buds are brown and sharp-pointed. As a tree, a maple has, of course, a trunk, roots and branches.

Suppose that while strolling on a nice winter day someone tells us that a particular tree is a sugar maple. A simple introspection tells us that the fact that the tree does not have leaves would not strike us as a contradiction of what we know about sugar maples. Yet, clearly, the presence of leaves of a particular type is deeply embedded in our typical image of a maple tree. The two-tiered theory of concept representation explains this phenomenon simply: the inferential concept interpretation associated with the general concept of deciduous trees evokes a rule "in winter trees lose leaves." By deduction based on the subset relationship between maple trees and deciduous trees, the rule would be applied to the former. The result of this inference would override the stored standard information about maple trees, and the inconsistency would be resolved.

Suppose further that when reading our first book on computer data structures we encounter a drawing of an acyclic graph structure of points connected by straight lines, which the author calls a tree. Again, calling such a structure a tree does not evoke in us a strong objection, because we can see in it some abstracted features of a biological tree. Here the matching process involves inductive generalization.

Example 3. Concept of a triangle

Let us go back to the concept of a triangle. Formally, it can be described as a geometrical figure consisting of three non-colinear points connected by straight lines. Using notation of annotated predicate calculus (APC), which is equivalent to predicate calculus, but permits one to write logical statements in a more compact form and facilitates processes of generalizing expressions (Michalski, 1983), one can write:

```
Triangle(T, P1, P2, P3) <=   Consists(T, P1 & P2 & P3) &
                             ConnectedBy(P1,P2 & P1,P3 &  P2,P3) = StraightLine &
                             RelationType(P1, P2, P3) = non-colinear
```

In the above expression, the symbol "&" denotes the *internal conjunction, i.e.,* conjunction of terms (rather than predicates). For example, the predicate "consists(T, P1 & P2 & P3)" states that the triangle T consists of the point P1and P2 and P3.

Suppose someone tells us that three trees in his yard form a triangle. Obviously, the meaning of the triangle in this statement differs from that in the formal descritpion. To match the two, one needs to make the following assumptions:

a. In the context of describing a configuration of real objects, such as trees, the individual objects play the role of the nodes. Thus, the matching operation involves drawing an analogy between the abstract nodes and the trees. This analogy can be viewed as consisting of one step of generalization (GEN):

Point ----GEN----> Object

and one step of specialization (SPEC):

Object ----SPEC----> Tree

b. In the context of trees, the presence of "straight line" is imaginary, i.e., there is not physical connection, but one could imagine a line between the objects (trees). The condition "ConnectedBy" is satisfied in such an abstract sense. This is an operation of generalization. Thus, matching the statement about a triangle arrangement of three trees with formal definition of a triangle requires various operations of generalization and specialization.

The examples presented above illustrate the ideas behind the two-tiered presentation, and show that relating concept instances to concept representations is more than matching features and determining a numerical score, as done in various mechanized decision processes, e.g., expert systems. It shows that it may involve performing various forms of inference.

3. The Interrelationship between BCR and ICI

To illustrate the idea of two-tiered representation, and a trade-off between the BCR, let us consider an imaginary concept, which we call an "R-ball." Suppose that the meaning of this concept is defined by the diagram in figure 2.

Figure 2. A diagram illustrating the concept of R-ball.

Each "1" in the diagram describes an instance of the R-ball, by defining specific values of attributes for this instance. The set of all instances of the R-ball depicted in the diagram constitutes the concept of R-ball. A complete and consistent description of the concept (i.e., a one that covers all "1"s, and only them) is:

shape=round & bounces or
shape=round & size=medium or large or
bounces=yes & size = medium or large (1)

Any instance that matches the above description is recognized as an R-ball (assuming that "&" is interpreted as logical conjuntion, and "or" as logical disjunction).

Let us now consider the diagram in figure 3, in which there are only four examples (the four "1"s).

Figure 3. A subset o examples of the R-ball.

A complete and consistent description of the examples is:

shape=round & bounces=yes & size=medium or large (2)

If interpreted the same way as above, this description covers only a subset of R-balls. Suppose, however, that we interprete the "&" not as conjunction, but as the *average* function, and the individual conditions (e.g., shape is round), as taking value 1 when satisfied (matched), and value 0 when not satisfied by an instance of the ball. Let us assume that a ball is classified as an R-ball, if the average of the degrees of match (DM) of the three conditions is equal or greater than 2/3. It is easy to notice that with such an interpretation of the description, the ultimate classifitaion of different instances into R-balls and not-R-balls will be exactly the same as in the first case (eq. 1).

Thus, we have two logically equivalent representations of R-balls, as summarized in Table 1. The first representation, CR1, is significantly more complex than the second, CR2, as it consists of 3 conjunctions (in BCR), while the second consists of only 1 conjunction. A price for the memory saving in CR2 is a slightly more complex interpretation method (ICI) of the second description. In general, there can be a range of logically equivalent descriptions that differ in the relative proportion of BCR and ICI.

The method of learning two-tiered concept representations to be described later is based on a general inductive learning program, AQ15. Before introducing the method, we first briefly describe this program.

4. An Overview of AQ15

The program AQ15 is a descendant of the the AQ1-AQ11 series of inductive learning programs (e.g., Michalski,1972; Michalski & Larson 75), and more recently of the GEM program (Mozetic & Hong, 1984). Various versions from the AQ family were experimentally applied to many practical problems, such as learning criteria for discriminating between cancer of the pancreas and cancer of the liver (Michalski, 1973), determining rules for plant disease diagnosis (Michalski &

Chilausky 80), describing "win" and "draw" positions in chess end-games (Negri and Michalski, 1977), and more recently for automatically creating a knowledge base for diagnosing cardiac arrhythmias from electrocardiograms (Mozetic, 1986).

The above-mentioned series of programs are based on the AQ algorithm, which generates the minimum or near minium number of general decision rules characterizing a set of instances, as originally described in (Michalski 69; Michalski & McCormick 71). While a complete description of the algorithm is somewhat complicated, its basic structure is simple.

1. A single positive example, called a *seed*, is selected, and a set of most general conjunctive discriptions of this example is computed (such a set is called a *star* for the seed). Each of these descriptions must exclude all negative examples.

2. Using a *description preference criterion* (described below), a single description is selected from the star, called the "best" description. If this description covers all positive examples, then the algorithm stops, as this description is a complete and consistent characterization of all training examples.

3. Otherwise, a new seed is selected among the unexplained (uncovered) examples, and steps 1 and 2 are are repeated until all examples are covered.

The disjunction of the descriptions selected in each step constitues a complete, consistent and general description of all examples. The preference criterion used in selecting a description from a star is expressed as a list of elementary criteria that are applied lexicographically and with a certain tolerance. The criteria may include the simpllicity of description (measured, e.g., by the number of variables used), the cost (the sum of measurement costs of individual variables), an estimate of generality (e.g., the ratio of the number of all possible examples covered to the number of training examples covered), or other criteria (Michalski, 1973, 1983).

The conjunctive descriptions are expressed using the *variable-valued logic system 1* (VL_1), which is a multiple-valued logic propositional calculus with typed variables (e.g., Michalski, 1974). In VL_1, the simplest statement is a *selector* , which relates a variable to a value or a disjunction of values, for example, [color = blue or red], or [x > 3]. A conjunction of selectors forms a *complex*, which represents a conjunctive description. A *cover* of a concept is a disjunction of complexes describing all positive examples and none of the negative examples of the concept. Thus, a cover can be viewed as the condition part of a rule that defines a concept.

By modifying a "generality parameter" AQ15 can produce rules of different degrees of generality (e.g., rules may be most general, most specific or intermediate). The basic mode of the program produces the most general rules, i.e., rules (complexes) that cover the maximum number of instances (observed or hypothetical), without covering any negative examples. By specializing such rules one can reach the other extreme, i.e., rules that cannot be more specialized without increasing the number of rules or "uncovering" some examples.

The program is also capable of *incremental learning with perfect memory*. In this type of learning, the process of modifying the current hypothesis to accommodate new facts takes into consideration all past examples (Reinke & Michalski 86). Thus, past examples must be stored, but the advantage is that the modified hypothesis is guaranteed to be always complete and consistent with regard to all the examples, and therefore such a method is potentially capable of producing higher quality rules than methods that remember only the current hypothesis. An incremental AQ-based learning program that does not store past examples is described in (Michalski and Larson, 1976).

Because the program can learn incrementally, it allows a user to supply some initial decision rules, which are then improved by an exposure to new examples. The program can also perform

constructive induction, in which background knowledge is used to generate new attributes, not present in the original set of examples. The background knowledge is expressed in the form of rules, which can be one of two types: L-rules that define values of new variables in the form of *logical assertions*, and A-rules that introduce new variables as *arithmetic functions* of original variables. A detailed description of the program is in (Michalski, Mozetic and Hong, 1989).

5. The TRUNC Method and Flexible Matching

The basic idea behind the TRUNC method is determine the "best" distribution of the concept description between the explicit base concept representation (BCR) and the inferential concept interpretation (ICI) [Michalski, 1986a, Michalski, 1986b]. This idea can be simply realized as described below.

As mentioned before, in AQ15 a concept is represented in the form of a disjunction of conjunctive statements (complexes). Each complex corresponds to a single rule stating that if conditions in the complex are satisfied by an event then this event belongs to the concept. The program associates with every complex a pair of weights: t and u, representing the *total* number of instances (events) covered by the complex, and the number of events covered *uniquely* by that complex, respectively. Thus, the t-weight may be interpreted as a measure of the representativeness of a complex as a concept description. The u-weight may be interpreted as a measure of importance of the complex. The complex with the highest t-weight may be viewed as describing the most typical concept examples, and thus serve as its prototypical description. The complexes with lowest u-weights describe rare, exceptional cases. If the learning events from which rules are derived are noisy, such "light" complexes may also be indicative of errors in the data.

We distinguish between two methods for recognizing the concept membership of an instance: the *strict* match and the *flexible* match. In the strict match, one tests whether an instance satisfies strictly the condition part of a rule (a complex). In the flexible match, one determines a degree of closeness between the instance and the condition part.

Using the strict match, one can recognize a concept without checking other candidate concepts. In the flexible match, one needs to determine the most closely related concept. Such a match can be accomplished in a variety of ways, ranging from approximate matching of features to *conceptual cohesiveness* [Michalski & Stepp 83].

The above described concept of t- and u-weights associated with complexes suggests an interesting possibility. Suppose that we order complexes in a description from those with the highest t-weight to those with the lowest t-weight. Figure 4 illustrates an ordering on the basis of the t-weight.

In each pair (x, y), x represents the t-weight, and y represents the u-weight.

Figure 4. An illustration of a t-ordered cover.

Suppose that we remove from such a cover the lightest complex (in figure 4, the complex Cpx_4). The so truncated description will not strictly match the events that uniquely satisfy the removed complex. However, by applying a flexible match, these events may still come out to be the most similar to the correct concept, and thus be correctly recognized. Thus, the removal of a complex does not necessary mean a decrease of the performance of the description on learning examples. The price for this simplification is a somewhat more complex evaluation of the description.The so truncated descritpion is applied to a set of *testing events* , and the performance score is determined.

Next, the "lightest" complex in the truncated cover is removed, and the performance of the obtained description on the same testing examples as before is determined. This process is continued until only one complex remains (the heaviest). In figure 4, cuts a, b and c mark covers obtained after the 1st, 2nd and 3rd step of truncation (with the number of complexes equal 3, 2 and 1, respectively). The "best" description among the truncated covers generated in consecutive steps is the one that scores highest on some criterion that pits the reduction in description complexity against its performance on the training and testing set of examples. The specific form of such a criterion can be defined appropriately to the problem domain. The above described method of knowledge reduction by truncating ordered covers, and employing flexible matching for example recognition, is called TRUNC.

Intuitivelly, one might expect that there is some trade-off between the simplicity of a description and its performance. An interesting problem is then to test the existence of this trade-off, in particular, to test how cover truncation affects the accuracy and the complexity of the decision rules in different practical settings. Section 7 presents rather surprising results of such experiments. Various trade-offs in creating descriptions are studied in *variable precision logic* (Michalski & Winston, 1986).

6. Matching concepts with examples

We now turn to the problem of matching concepts descriptions with examples, and the resolution of conflict when several descriptions are matched by a single example. When strictly matching a new example against a set of (disjunctive) rules, three outcomes are possible: there may be only one match, more than one, or no match. These three types of outcomes are called *single-match, multiple-match* and *no-match*, respectively (figure 5). Each type of match requires a different evaluation procedure, and a different method of determining the accuracy of concept recognition.For single match, the evaluation is easy: the decision is counted as correct, if it is equal to the expert-given classification of the testing example, and as incorrect, otherwise.

If there is a multiple match or no-match, the system activates a *flexible evaluation* scheme that determines the most plausible decision. This decision is compared with an expert decision, and evaluated as correct or incorrect. There many ways one could define a flexible evaluation scheme. Below is an example of a scheme, which defines two simple heuristic classification criteria, one for the multiple-match case, and the other for the no-match case.

The multiple-match case

When an event matches more than one rule, the system selects the one that represents the most plausible decision. Let $C_1,...,C_n$ denote decision classes and, e denote an event to be classified. For each decision class C_i we have a rule (a cover) that consists of a disjunction of complexes (Cpx). Each complex is, in turn a conjunction of selectors (Sel).

We first define the *strenght of support*, SS(Cpx_j, *e*), provided by a complex Cpx_j that is satisfied by the event *e* , as follows:

The SS(Cpx_j, *e*) is the ratio of the t-weight(Cpx_j), i.e., the total number of training examples covered by the complex Cpx_j, by the total number of possible examples covered by the complex (#examples), if the complex is satisfied by the event *e*, and is 0 otherwise:

$$SS(Cpx_j, e) = \begin{cases} \text{t-weight}(Cpx_j)/\#\text{examples, if } e \text{ satisfies } Cpx_j \\ 0, \text{ otherwise.} \end{cases}$$

A justification for this measure is that if a "heavy" complex (i.e., representing more typical events) is satisfied, then it is more likely that the event belongs to the class described by this complex.

The *strength of support* , SS(C_i), for the class C_i, is the *probabilistic sum* of SSs of the complexes in the rule for this class. Thus, if the rule for C_i consists of a disjunction of two complexes, Cpx_1 and Cpx_2, we have:

$$SS(C_i,e) = SS(Cpx_1,e) + SS(Cpx_2,e) - SS(Cpx_1,e) \times SS(Cpx_2,e)$$

The probabilistic sum is used, because individual complexes may intersect, and thus are interdependent. The most plausible class is defined as the one with the largest SS, i.e., the one whose satisfied complexes cover the largest number of learning examples. Obviously, if the description is not satisfied by the given event, its SS equals 0.

The no-match case

If an event does not satisfy any rule in the rule set under consideration, then the system performs a flexible matching, i.e., evaluates a degree of closeness between the event and each rule, and selects the class, whose rule is "closest" to the event. One way to perform such a matching is to define a *measure the fit* between attribute values in the event and the class description, taking into consideration the prior probability of the class. For illustration, let us present here a very simple measure of fit , MF, which was described in (Michalski, Mozetic & Hong, 1986):

First, we define the measure of fit, MF(Sel_k, *e*), of the selector Sel_k to the event *e*. This measure takes value 1, if the selector is satisfied. Otherwise, it is proportional to the amount of the domain space covered by the selector:

$$MF(Sel_k, e) = \begin{cases} 1 & \text{if selector } Sel_k \text{ is satisfied by } e \\ \#\text{values/DomainSize,} & \text{otherwise} \end{cases}$$

where #values is the number of alternative attribute values in the selector, and DomainSize is the total number of the attribute's possible values.

An intuitive justification of this measure is that if an event does not satisfy a selector with just one or few alternative values, then this is a stronger negative evidence then if it does not satisfy a selector with many alternative values.

The measure of fit, MF(Cpx_j, e), of a complex Cpx_j to the event *e*, is defined as the product of MFs of selectors in the complex, weighted by the ratio of t-weight(Cpx_j) (the number of learning

examples covered by the complex) and #examples (the total number of possible examples covered by the complex):

$$MF(Cpx_j, e) = \prod_i (MF(Sel_k, e) \times (\text{t-weight}(Cpx_j)/\#\text{examples})$$

Individual selectors in a complex can be viewed as independent conditions, and therefore multiplying MFs of the selectors in the complex seems to be reasonable.

The measure of fit, $MF(C_i, e)$, of the class C_i to event *e*, is defined as the probabilistic sum of the MFs of complexes in the rule for the class:

$$MF(C_i, e) = MF(Cpx_i, e) + MF(Cpx_2, e) - MF(Cpx_1, e)\, MF(Cpx_2, e)\}$$

As before, the probabilistic sum is justifiable because individual complexes may be intersecting, and thus interdependent.

The above measure of fit between a class and event is thus a combination of "closeness" of the event to the class and an estimate of the prior probability of the complexes in the class. This measure could be further extended by introducing a measure of degree to which a selector is satisfied by an event [Michalski& Chilausky, 80].

EXPERIMENTS AND ANALYSIS OF RESULTS

To illustrate the ideas described in this chapter, we will briefly describe some preliminary experiments performed using data from three medical domains: lymphography, prognosis of breast cancer recurrence and location of primary tumor (Michalski, Mozetic &Hong; 1986). All data were obtained from the Institute of Oncology of the University Medical Center in Ljubljana, Yugoslavia (Kononenko, Bratko& Roskar, 1984).

Lymphography
There were 4 decision classes (possible diagnoses). Each patient was characterized by 18 multivalued attributes. The data about 148 patients diagnosed by experts were used in the experiment. The diagnoses were not verified independently, and physicians were not tested in any way. A specialist's estimation is that internists diagnose this kind of diseases correctly in about 60% of cases, and specialists in about 85% of cases.

Prognosis of Breast Cancer Recurrence
There were 2 decision classes. Each patient was characterized by 9 multivalued attributes. The set of attributes was incomplete, in the sense that these attributes were insufficient to completely discriminate between cases of different disease. Thus, diagnosing on the basis of these attributes has to produce a certain amount of error. Data about 286 patients with known diagnosistic status 5 years after the operation were used in the experiment. Five specialists of the Institue of Oncology in Ljubljana were tested, and they gave a correct prognosis in 64% of cases.

Location of Primary Tumor
Physicians distinguish between 22 possible locations of primary tumor. Each patient was characterized by 17 attributes (as in the prognosis of breast cancer recurrence, this set was also incomplete). Data of 339 patients with known locations of primary tumor were used in the experiment. Four internists that were tested determined a correct location of the primary tumor in 32% of cases, and four oncologists (specialists) in 42% of test cases.

Table 1 summarizes the data used in the experiments. Individual columns represent: the problem domain (a disease type); the number of available examples for this disease type; the number of

different diseases of the given type; the number of attributes used to characterize a case, and the average number of values per attribute for each of the domains, respectivley.

Domain	Examples	Classes	Attrs	Vals/Attr
Lymphography	148	4	18	3.3
Breast cancer	286	2	9	5.8
Primary tumor	339	22	17	2.2

Table 1. A characterization of three poblem domains

In all three domains, 70% of examples were selected for learning diagnostic rules by the program, and the remaining 30% for testing the obtained rules. Each testing experiment was repeated 4 times with randomly chosen examples.

The statistics in Table 2 include average number of complexes per rule, average number of attributes per complex, average number of values per attribute and finally, average number of learning examples covered by one complex in covers generated by AQ15. We can see that in the domain of primary tumor decision rules consist of complexes that in average cover slightly more than 2 examples. In the domain of lymphography complexes in average cover 8 examples, which indicates a presence of relatively strong patterns.

Domain	Cpx/Rule	Attrs/Cpx	Values/Attr	Examples/Cpx
Lymphography	3	3.1	1.8	8
Breast cancer	20	3.9	1.7	5
Primary tumor	5.2	5.3	1.0	2.3

Table 2. Average complexity of AQ15's decision rules in the three medical domains, when no cover truncation was applied.

In addition to results obtained from using complete (untruncated) rules, results of two other experiments were obtained. In the first experiment we eliminated from rules all complexes that cover uniquely only one learning example (unique), and in the second we eliminated all complexes except the most representative one covering the largest number of learning examples (best cpx). Complexity of rules is measured by the number of selectors and complexes. The results describing the average of 4 experiments are presented in *Table 3*.

The column "Rule type" specifies which of the three types of rules was used for testing:

"complete" - the rules were complete and consistent with regard to all training examples, i.e., they were exactly in the form produced by AQ15,

"unique>1" - the rules did not include complexes with u-weight = 1 (i.e., complexes that covered uniquely only one event),

best cpx - the rule for each class consisted of just a single complex - the one with the highest t-weight (i.e., covering the largest number of examples).

The column "Complexity" gives an estimate of the complexity of the rules in each case, as measured by the total number of selectors (#Sel), and the total number of complexes (#Cpx) in the rule(s) for each class.

Domain	Rule type	Complexity #Sel	#Cpx	Accuracy	Experts/ Internists	Random Choice
Lymphography	complete	37	12	81%	85%/60%	25%
	unique >1	34	10	80%		
	best cpx	10	4	82%		
Breast cancer	no	160	41	66%	64%	50%
	unique >1	128	32	68%		
	best cpx	7	2	68%		
Primary tumor	no	551	104	39%	42%	5%
	unique >1	257	42	41%		
	best cpx	112	20	29%		

Table 3. Results of experiments in three medical domains

Results for human experts are the average of testing of five and four domain specialists in the domains of breast cancer recurrence and primary tumor, respectively (Kononenko, Bratko & Roskar, 1986). In the domain of lymphography, the internists' accuracy is only their own estimate, and was not actually measured.

Some results shown in Table 3 seems to be quite surprising. The most striking case concerns the diagnosis of lymphography. When the cover of each disease was truncated to only one complex (best cpx), the complexity of the rule set for went down from the total of 12 complexes (rules) and 37 selectors (conditions) to only 4 complexes (one per class) and 10 selectors. At the same time the performance of the rules went slightly up (from 81% to 82%).

A similar phenomenon occurred in the breast cancer domain, where the number of selectors and complexes went down from 160 and 41 to 7 and 2, respectively; while the performance went again slightly up, from 66% to 68%. This means that by using the TRUNC method one may significantly reduce the knowledge base without decreasing its performance accuracy.

The domain of lymphography seems to have some strong patterns and the set of attributes is known to be complete.There are four possible diagnoses, but only two of them are prevailing. The domain of breast cancer has only two decision classes, but does not have many strong patterns.

The the domain of the location of primary tumor has many decision classes and mostly binary attributes. There were only few examples per class available, and the domain seems to be without any strong patterns. The set of available attributes was incomplete, i.e., was not sufficient to discriminate between different classes. Such a low quality data explain the low performance of the system in this domain.

In the first two domains, the strongly simplified, incomplete rule base gave a slightly better performance on testing examples than the original complete rule base. How was this possible? To explain this, let us first observe that removing complexes from a cover is equivalent to removing disjunctively linked conditions from a concept description. Such a process *overspecializes* the knowledge representation, i.e., produces a concept description that does not cover some positive training examples (hence we have an incomplete concept description).

The fact that such a description produced good results may be attributed to flexible matching. Events that were not strictly covered can still be correctly classified, if they are evaluated to be

"closer" to the truncated descriptions of the correct class than to descriptions of other classes (recall the 'no-match" case). Since the "best complex" may be interpreted as representing the most typical cases, it is likely that examples that are not covered by it are still "closer" to it than to the "best complexes" of other classes.

The above method of knowledge reduction by specialization may be contrasted with knowledge reduction by generalization used in the ASSISTANT learning program, a descendant of ID3 (Quinlan, 1983). This program represents knowledge in the form of decision trees, and has been applied to exactly the same medical problems as here (Kononenko, Bratko & Roskar, 1986). The program applies a tree pruning technique based on the principle of maximizing classification accuracy. The technique removes certain nodes from a tree, and this is equivalent to removing conjunctively linked conditions from a concept description. Thus, such a knowledge reduction technique overgeneralizes knowledge representation, producing an *inconsistent* concept descriptions (i.e., the ones that cover some negative examples).

It should be noted, however, that this method does not use any flexible matching technique, and therefore it, in principle, must produce errors on the training examples (unlike the TRUNC method that may not produce such errors). Nevertheless, this technique may also lead to an improvement of accuracy in classifying testing examples. The reason for this may be the known statistical phenomenon of "overfitting." More general descriptions (due to pruning) may avoid missclassifications produced by overly specific descriptions. Table 4 presents the complexity and diagnostic accuracy of the trees generated by the ASSISTANT's program, with and without the tree pruning mechanism, for the same data as used in the above described experiments. (Kononenko, Bratko & Roskar, 1986).

Domain	Tree pruning	Complexity Nodes	Leaves	Accuracy
Lymphography	no	38	22	76%
	yes	25	14	77%
Breast cancer	no	120	63	67%
	yes	16	9	72%
Primary tumor	no	188	90	41%
	yes	35	18	46%

Table 4: Results from the experiments using the ASSISTANT program.

Comparing these results with those in Table 3 one can notice that in the first two, more structured domains, the "best complex" representation is substantially simpler than the pruned tree representation (a complex corresponds to a leaf, and a selector in a complex roughly corresponds to a node in the tree). For example, in lymphography, the best complex description involves 4 complexes and 25 selectors, as compared to 14 leaves and 25 nodes in the pruned tree. On the other hand, in the domain of primary tumor, in which data were very low quality and noisy, the pruned tree performed better than the best complex, though the results in both cases were rather poor (46% and 29%, respectively).

The technique of tree pruning corresponds to the removal of selectors from complexes. As such, it can also be applied to covers. Thus, in general, the reduction of the base concept representation (BCR) in the form of covers may involve not only removal of complexes from the cover (a specialization process), but also a removal of selectors from complexes (a generalization process).

specialization process), but also a removal of selectors from complexes (a generalization process). This means that the resulting BCR would be both, inconsistent and incomplete. Such a technique has been implemented, and is described in (Bergadano at al, 1988a,b,c). In this technique, complexes and selectors are removed from the original cover so that a certain *criterion of description of quality* is maximized.

7. Summary

We have presented here ideas and a method for representing concepts whose meaning is flexible and dependent on the context of discourse. The two-tiered representation splits the concept description into a part that explicitly defines the general, typical and easy-to-define properites (the base concept representation - BCR), and the part that describes allowable modifications, context-dependency and special cases (inferential concept interpretation - ICI)..

The method presented employs the AQ15 learning program to generate complete and consistent concept descriptions, which are then optimized to create the BCR of concepts. The so obtained concept descriptions (concept recognition rules) are easy to interpret and comprehend. The program has proven itself to be a powerful and versatile tool for experimenting with problems of inductive knowledge acquisition. The knowledge representation in the program is, however, limited to only attribute-based descriptions. For problems that require structural descriptions a related program, INDUCE3 (Hoff, Michalski & Stepp, 1983), or its incremental learning version INDUCE4 [Mehler, Bentrup & Riedsel,86] can be used.

Experiments with the implemented system have shown that it is able to produce relatively high quality decision rules in domains with noisy and overlapping learning examples. The potentially most significant outcome of the experiments is the demonstration that the two-tiered representation obtained by truncating the covers and applying flexible matching may lead to a very substantial reduction of the size of the rule base without decreasing its performance accuracy.

Further research is required to find out the generality and usefulness of the presented ideas across different domains, and to develop and experiment with more advanced two-tiered representations. Other interesting problems are to develop criteria for determining the best trade-off between the base concept representation and inrefential concept interpretation in different domains, and the organization of two-tiered representations of systems of interrelated flexible concepts.

Acknowledgment

The author thanks members of the Advanced Machine Learning Seminar and the Center for Aritical Intelligence at George Mason University for the discussion and critisism of the ideas presented here. The work described was done partially in the Laboratory for Artificial Intelligence at the University of Illinois at Urbana-Champaign, and partially in the Center for Artificial Intelligence at George Mason University.

This research was supported in part by the National Science Foundation under grant No. DCR 084-06801, the Office of Naval Research under grants No. N00014-82-K-0186, N00014-88-K-0226 and N00014-88-K-0397, and the Defense Advanced Research Project Agency under grant administered by the Office of Naval Research No. N00014-87-K-0874.

Rreferences

F. Bergadano, S. Matwin, R.S. Michalski, Z. Zhang, A Measure of Quality of Descriptions, Reports of the Machine Learning and Inference Laboratory, No. 88-3, Artificial Intelligence Center, George Mason University, 1988.

F. Bergadano, S. Matwin, R.S. Michalski, Z. Zhang, Learning Flexible Concept Descriptions Using a Two-tiered Knowledge Representation: Ideas and a Method, Reports of the Machine Learning and Inference Laboratory, No. 88-4, Artificial Intelligence Center, George Mason University, 1988.

F. Bergadano, S. Matwin, R.S. Michalski, Z. Zhang, Learning Flexible Concept Descriptions Using a Two-tiered Knowledge Representation: Implementation and Experiments, Reports of the Machine Learning and Inference Laboratory, No. 88-35, Artificial Intelligence Center, George Mason University, 1988.

Hong, J., Mozetic, I., Michalski, R.S. (1986). "AQ15: Incremental Learning of Attribute-Based Descriptions from Examples, The Method and User's Guide." Report ISG 86-5, UIUCDCS-F-86-949, Dept. of Computer Science, University of Illinois, Urbana.

Hoff, W., Michalski, R.S., Stepp, R.E. (1983). "INDUCE.2: A Program for Learning Structural Descriptions from Examples. Report ISG 83-4, UIUCDCS-F-83-904, Dept. of Computer Science, University of Illinois, Urbana.

Kononenko, I., Bratko, I., Roskar, E. (1986). "ASSISTANT: A System for Inductive Learning." Informatica Journal, Vol. 10, No. 1 (in Slovenian).

Mehler, G., Bentrup, J., Riedesel J. (1986). "INDUCE.4: A Program for Incrementally Learning Structural Descriptions from Examples." Reports of ISG , Dept. of Computer Science, University of Illinois, Urbana.

Michalski, R.S. (1969). "On the Quasi-Minimal Solution of the General Covering Problem." Proceedings of the V International Symposium on Information Processing (FCIP 69)\fR, Vol. A3 (Switching Circuits), Bled, Yugoslavia, pp. 125-128.

R. S. Michalski, AQVAL/1--Computer Implementation of a Variable-Valued Logic System VL_1 and Examples of its Application to Pattern Recognition, *Proceedings of the First International Joint Conference on Pattern Recognition, pp. 3-17*, Washington, DC, October 30 - November 1, 1973.

R. S. Michalski, Variable-Valued Logic: System VL_1, *Proceedings of the 1974 International Symposium on Multiple-Valued Logic*, pp. 323-346, West Virginia University, Morgantown, May 29-31, 1974.

Michalski, R.S. (1983). "Theory and Methodology of Machine Learning." In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach,* Morgan Kaufmann, Palo Alto, 1983.

Michalski,R.S. "Concept Learning." *Encyclopedia of Artificial Intelligence*, John Wiley & Sons, 1986b.

Michalski,R.S."Two-tieredConcept Representation, Analogical Matching and Conceptual Cohesiveness." Invited paper for the Workshop on Similarity and Analogy, Allerton House, University of Illinois, June 12-14, 1986b (an extended and improved version to appear in *Analogy and Similarity*, S. Vosniadou and A. Ortony (eds.), Cambridge University Press, 1989)

Michalski, R.S., Chilausky, R.L. "Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis." *International Journal of Policy Analysis and Information Systems*, Vol. 4, No. 2, pp. 125-161, 1980.

Michalski, R.S., Larson, J. "AQVAL/1 (AQ7) User's Guide and Program Description." Report No. 731, Dept. of Computer Science, University of Illinois, Urbana, 1975.

Michalski, R.S., McCormick, B.H. (1971). "Interval Generalization of Switching Theory."Report No. 442, Dept. of Computer Science, University of Illinois, Urbana.

R. S. Michalski and J. B. Larson, "Selection of Most Representative Training Examples and Incremental Generation of VL_1 Hypotheses: the underlying methodology and the description of programs ESEL and AQ11," Report No. 867, Department of Computer Science, University of Illinois, Urbana, May 1978.

Michalski, R.S., Mozetic, I., Hong, J. (1986)."The AQ15 Inductive Learning System: An Overview and Experiments." Report ISG 86-20, UIUCDCS-R-86-1260, Dept. of Computer Science, University of Illinois, Urbana.

R. S. Michalski and P. Negri, An Experiment on Inductive Learning in Chess End Games, *Machine Representation of Knowledge, Machine Intelligence 8*, E. W. Elcock and D. Michie (Eds.), Ellis Horwood Ltd., New York, pp. 175-192, 1977.

Michalski, R.S., Stepp, R.E. (1983). "Learning from Observations: Conceptual Clustering." In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning - An Artificial Intelligence Approach\fR, Palo Alto: Tioga.

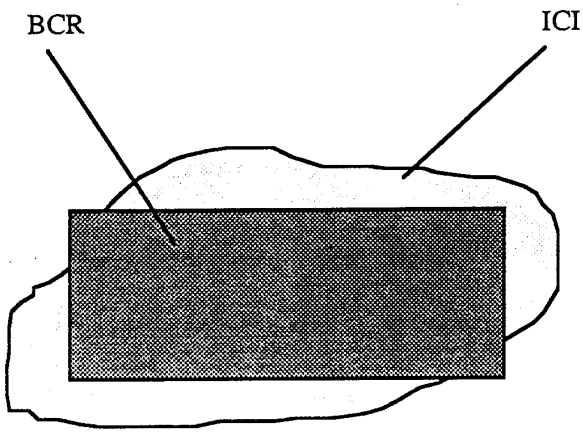
Michalski, R.S., Winston, P.H. (1986). "Variable Precision Logic." AI memo No. 857, MIT, Cambridge. An extended version to appear in \fIAI Journal\fR.

Michalski, R. S. Two-tiered concept meaning, flexible matching and conceptual cohesivenes, an invited paper for the Workshop on Similarity and Analogy, Allerton House, June 1986. ???

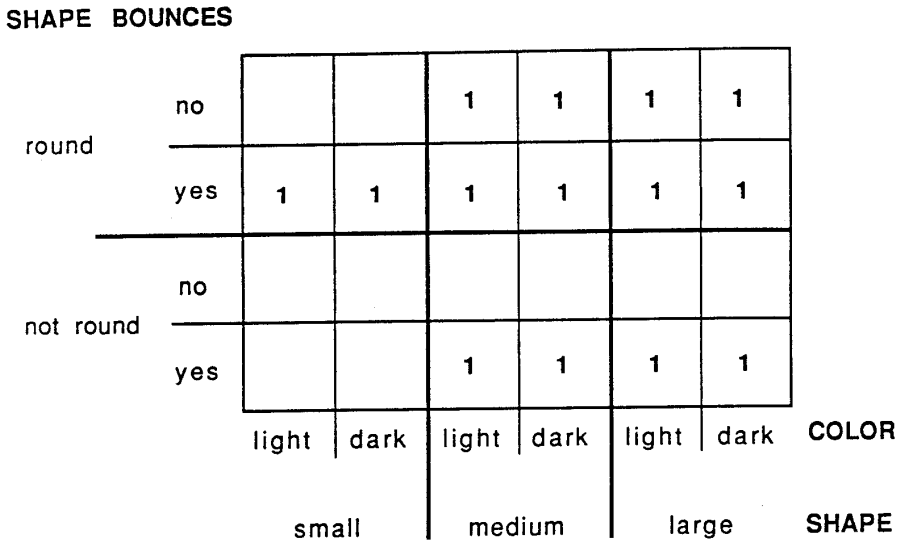
Mozetic, I. (1986). "Knowledge Extraction through Learning from Examples." In T.M. Mitchell, J.G. Carbonell, R.S. Michalski (Eds.), \fIMachine Learning: A Guide to Current Research\fR, Kluwer Academic Publishers. [16] Quinlan, J.R. (1983).

Quinlan, R.J., "Learning Efficient Classification Procedures and their Application to Chess End Games." In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning - An Artificial Intelligence Approach, Palo Alto: Tioga, 1983.

Reinke, R.E., Michalski, R.S. (1986). "Incremental Learning of Decision Rules: A Method and Experimental Results." To appear in J.E. Hayes, D. Michie, J. Richards (Eds.), Machine Intelligence, Oxford University Press.


BCR
ICI

Learning the Concept of R-BALL



Representation:

SHAPE = round & **BOUNCES** = yes
or
SHAPE = round & **SIZE** = medium or large
or
BOUNCES = yes & **SIZE** = medium or large

Interpretation:

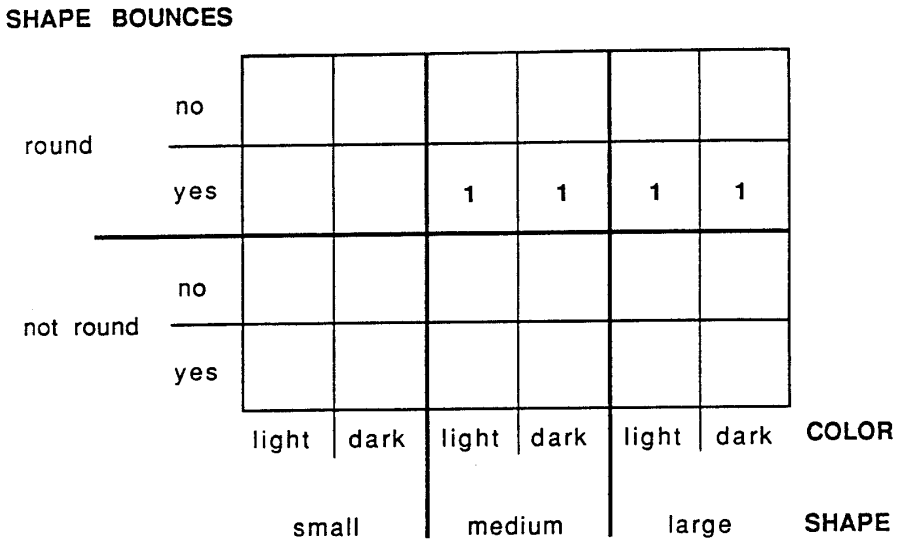
& = MIN (conjunction)
or = MAX (disjunction)

Concept membership:

yes, if degree of match = 1
no, otherwise

LearTwoTier: R-Ball

A Two-tiered Representation



Representation (BCR):

SHAPE = round & **BOUNCES** = yes & **SIZE** = medium or large

Interpretation (ICR):

& = AVG

Concept membership:

yes, if degree of match ≥ 2/3
no, otherwise

LearTwoTier:2tierRep

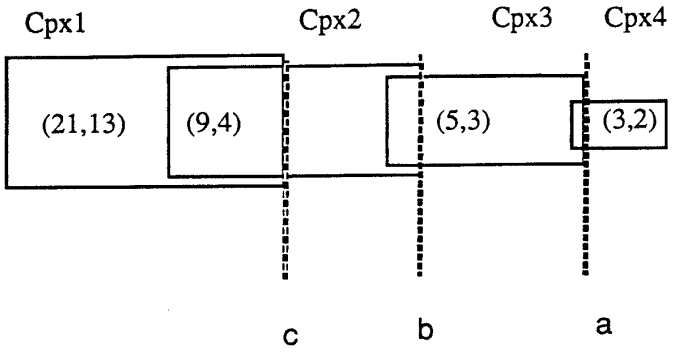
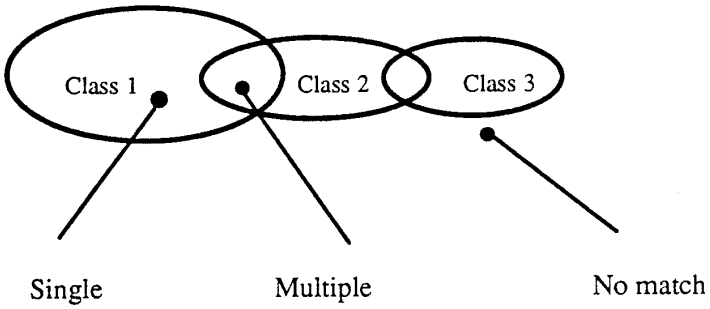

Cpx1
Cpx2
Cpx3
Cpx4
(21,13)
(9,4)
(5,3)
(3,2)
c
b
a

Fig. 6


Class 1
Class 2
Class 3
Single
Multiple
No match

A Comparison Between Two Concept Representations

CR1		CR2	
BCR	ICI	BCR	ICI
3 conjunctions 6 conditions	& = MIN V = MAX DM = 1	1 conjunction 3 conditions	& = AVG V = MAX DM ≥ 2/3

LeaTwoT:REPvsIN